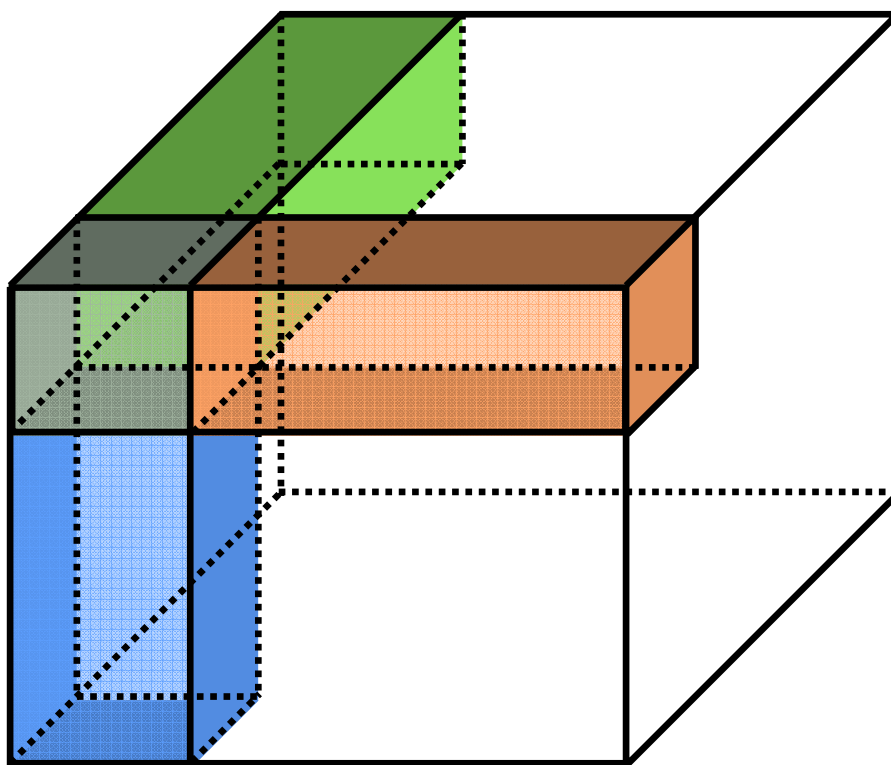


Practical and computational aspects in chemometric data analysis

Ph.D. Dissertation by Giorgio Tomasi



Department of Food Science
The Royal Veterinary and Agricultural University
Frederiksberg, Denmark

Practical and computational aspects in chemometric data analysis

Ph.D. Dissertation by

Giorgio Tomasi

May 30th, 2006

Supervisor: Prof. Rasmus Bro
Department of Food Science
The Royal Veterinary and Agricultural
University, Frederiksberg, Denmark

Co-supervisor: Prof. Age K. Smilde
Swammerdam Institute for Life Sciences
Universiteit van Amsterdam, Amsterdam,
The Netherlands

Title: Practical and computational aspects in chemometric data analysis

Author: Giorgio Tomasi

University: The Royal Veterinary and Agricultural University

Department: Food Science, Quality and Technology section

Abstract: The relevant aspects of this work span several levels in chemometrics. The algebraic properties of a basic operation connected to PARAFAC have been studied to develop computationally efficient procedures that were used to improve the existing methods for fitting this model. Thereafter, the usefulness of such methods has been tested in connection with common problems in chemometric data analysis (e.g., high collinearity or presence of missing values) on real as well as simulated data sets. Likewise, two alignment procedures based on dynamic programming have been studied to determine their connection and their suitability as preprocessing methods for chromatographic data. Subsequently, such algorithms have been employed for both exploratory studies and calibration problems.

Keywords: Exploratory data analysis, multivariate data analysis, chemometrics, multi-way analysis, multilinear models, parallel factor analysis, CANDECOMP, PARAFAC, PARAFAC2, principal components analysis, PCA, dynamic programming, alignment algorithms, registration algorithms, warping algorithms, Correlation Optimised Warping, COW, Dynamic Time Warping, DTW, data preprocessing.

Cite as: Tomasi G., Practical and computational aspects in chemometric data analysis, Ph.D. dissertation, The Royal Veterinary and Agricultural University, Frederiksberg, Denmark, 2006. Available at: www.models.kvl.dk.

Number of pages: xx + 263

ISBN-10: 87-7611-131-8

ISBN-13: 987-87-7611-131-8

Printed by: Samfundslitteratur Grafik, Frederiksberg, Denmark

Editing completed: Saturday, 06 May 2006

Preface

This Ph.D. dissertation is based on the research carried out at The Royal Veterinary and Agricultural University in the Quality and Technology section of the Food and Dairy Science department under the supervision of Professor Rasmus Bro, whom I wish to thank for his help, for the numerous collaborations and for his forbearance.

Four months were spent at the University of Amsterdam under the supervision of Professor Age K. Smilde. I wish to thank him and all the members of the group for making me feel welcome and for all the things I have learnt during my stay.

I gratefully acknowledge the financial support of the NWAYQUAL European project and the contribution of all the people that participated in it at the Pharmaceutical Institute of the Vrije Universiteit Brussel (Belgium), at the Centre de Recherche de Solaize (France) and at the Université d'Orsay (France).

I am also very grateful to all the people at the Quality and Technology group, past and present, for their help and for the excellent working environment. Amongst them, special thanks go to those that had the misfortune of having to share their office with me over the years (Charlotte, Vibeke and Åsmund) and to Frans and Marc, for the countless inspiring conversations and for the several collaborations.

I am indebted to all the people that have contributed to the publications that appear in this thesis and especially to Jan, whose knowledge have proved essential for a considerable fraction of this work, and Gilda, for having proofread almost everything remotely scientific I have written thus far.

The list of men and women whose contribution I feel I should acknowledge is far too long to be included here. Thus, the warmest thanks go to all of these people (and especially to my family and to my girlfriend) for the useful lessons they have taught me and for the support and affection they showed over the years.

Resumé

Denne ph.d. afhandling har til formål at undersøge de praktiske implikationer og beregningsmæssige aspekter af anvendelsen af multilineære modeller til analyse af kemiske data. Multilineære modeller som principal komponent analyse (PCA) og parallel faktor analyse (PARAFAC) er blevet mere og mere populære i kemometri, såvel i eksplorative studier og til kalibreringsformål. Men datasæt, der stammer fra kemiske eksperimenter og målinger, lever ikke altid op til de krav, modellerne stiller for at give meningsfulde resultater. For eksempel, kan dataanalyse umuliggøres af for stor en fraktion af manglende værdier eller underliggende fænomener, der ændrer form fra prøve til prøve. Dette kan føre til misvisende resultater ved efterfølgende anvendelse af multilineære modeller. Visse modeller, som f.eks. PARAFAC, kan være vanskelige at beregne, og der er konstant behov for forbedringer for at udvide deres anvendelighed. Der er derfor undersøgt to hovedemner i dette arbejde:

- Beregning af PARAFAC-modeller (Publikationer I, II og III)
- Anvendelsen af alignment algoritmer baseret på dynamisk programmering som et forbehandlingstrin forud for anvendelsen af multilineære modeller (Publikationer IV, V og VI).

Indledningen skitserer begge emner og nogle relaterede problemer med omfattende reference til eksisterende litteratur. Publikationerne I, II og III går i detaljer omkring beregning af PARAFAC-modeller og viser, hvordan alternating least squares algoritmen (ALS) ikke nødvendigvis er den bedste metode i en række tilfælde almindeligt forekommende i kemi. På grundlag af resultater fra reelle og simulerede datasæt er det vist, hvorledes andre ikke-lineære mindste kvadraters metoder (f.eks. Levenberg-Marquardt metoden) kan være mere effektive end ALS, for højere ordens data eller når store fraktioner og specifikke mønstre af manglende værdier forekommer. Derudover opnår man bedre performance med andre algoritmer i tilfælde af høj kollinearitet mellem faktorer eller forkert specifikation af model rang.

De samme resultater tyder på, at ikke-mindste kvadrater kriterier, der straffer løsninger med lav core consistency, kan give hurtigere algoritmer og bedre forudsigelser i kalibreringsmodeller (f.eks. Self-Weighted Alternating Trilinear Decomposition – SWATLD). Det konkluderes imidlertid, at yderligere undersøgelser er nødvendige for at fastslå grunden til stabiliteten forbundet med disse algoritmers tabsfunktioner og at SWATLD skal anvendes med yderste forsigtighed og hovedsagelig for at give startparametre for ALS eller andre algoritmer.

Publikation II fokuserer på problemet med at beregne PARAFAC modeller, når der er manglende værdier, hvilket er almindeligt inden for kemometrisk dataanalyse. Især effekten af mønstret og fraktionen af manglende værdier er undersøgt. Det er vist, hvorledes PARAFAC er betydeligt mere resistent end bilineære modeller som PCA over for manglende værdier og hvordan relativt gode estimater af modelparametrene kan opnås med op til 70 % manglende værdier. Endvidere er det vist, at mønstret af manglende værdier er langt vigtigere end deres fraktion for omfanget af artefakter i løsningerne og at tilstedeværelse af sådanne artefakter ikke nødvendigvis medfører dårlige forudsigelser, når PARAFAC-modeller er anvendt til kalibrering.

I Publikation III er undersøgt egenskaberne af det kolonne-vise Khatri-Rao produkt som et middel til forbedring af den beregningsmæssige effektivitet af en række ikke-lineære mindste kvadraters metoder. Hurtige algoritmer er blevet udviklet til at beregne Jacobian matricen og dens produkt med en vektor, samt Hessian og gradient matricer for

PARAFAC-modellen. Derudover er udviklet optimerede metoder til ALS og til eksakt line-search for PARAFAC algoritmer.

Den anden del af afhandlingen handler om effekten af såkaldte shifts i kemiske data (f.eks. retentionstids-variationer i kromatografi). Der er fokuseret på to algoritmer (Correlation Optimised Warping – COW og Dynamic Time Warping – DTW). I introduktionen såvel som Publikation IV beskrives de to algoritmer såvel som deres forbindelse til andre alternativer. I Publikation IV undersøges betydningen af forskellige mulige restriktioner i forbindelse med warping som forbehandling før PCA af gaskromatografiske data (GC-FID). Studiet viste, at rigide restriktioner er nødvendige i DTW såvel som i COW for denne type data.

I Publikation V anvendes COW som forbehandling før PCA-modellering af GC-MS data af olieprøver. En simpel metode er udviklet til at optimere warpingen ved hjælp af gentagne målinger. I samme artikel og Publikation VI undersøges også metoder til at eliminere basislinie og koncentrations-problemer.

Endelig viser Publikation VI, hvordan PARAFAC modeller og warping tilsammen kan anvendes til kalibrering. HPLC data med UV-Vis detektion er anvendt til at bestemme koncentration af antibiotika i biologiske prøver. I publikationen vises, at DTW ikke kan håndtere koelutionsproblemer samt at PARAFAC2 modellen kan.

I korte træk dækker afhandlingen algebraiske egenskaber forbundet med beregning af PARAFAC-modeller og numeriske aspekter i dynamisk programmering specielt i forbindelse med kromatografisk dataanalyse. Men en mere overordnet konklusion på afhandlingen er, at kemometrien kan have stor gavn af et mere udtalt samarbejde med eksperter fra andre områder, specielt inden for numerisk analyse og scientific computing. Af denne grund er de fleste implementerede algoritmer lagt på www.models.kvl.dk i form af MATLAB[®] rutiner. Derved er de lettilgængelige for det videnskabelige samfund.

Summary

The objective of this Ph.D. dissertation is to study the practical implications and some computational aspects of the use of multilinear models for the analysis of chemical data. Multilinear models like PCA and PARAFAC have grown increasingly popular in chemometrics both in the context of exploratory studies and for calibration purposes. However, data sets arising from chemical experiments do not always meet the conditions that are required to yield useful results in a reasonable time. For example, they can contain relatively large fractions of missing values or features like shifted factors between measurements that can confound the interpretation of low rank multilinear models. Moreover, the fitting of some of these models, and particularly of PARAFAC, is inherently difficult and there is a constant need for improvements to further extend their applicability. Hence, there are two main subjects investigated in this work:

- the fitting of the PARAFAC model (Papers I, II and III)
- the use of alignment algorithms based on dynamic programming as a preprocessing step prior to the fitting of multilinear models (Papers IV, V and VI).

The introductory part of this thesis outlines both subjects and some of the related problems and draws extensively from the existing literature from different fields.

Papers I, II and III further elaborate on the subject of fitting PARAFAC models and show how the alternating least squares algorithm (ALS) is not necessarily the best method in a number of cases that are common in chemistry. Based on the results on real and simulated data sets, it is shown how other nonlinear least squares solvers (e.g., the Levenberg-Marquardt method) can be more efficient than ALS when PARAFAC is fitted to higher-order arrays or when large fractions and specific patterns of missing values occur. Moreover, better performances are obtained with other algorithms in case of high collinearity between factors or wrong specification of the model rank with respect to the pseudo-rank of the data array.

The same results suggest that non-least squares fitting criteria that penalise solutions with low core consistency may yield faster algorithms and better predictions in calibration models (as in, for example, the Self-Weighted Alternating Trilinear Decomposition algorithm – SWATLD). However, it is concluded that further investigations are necessary in order to determine the reason for the stability associated with these loss functions and that SWATLD should be used with great care and mostly to provide initial values for ALS or other algorithms employing standard fitting criteria.

Paper II focuses on the problem of fitting PARAFAC in the presence of missing values, which are a common occurrence in chemometric data analysis. In particular, the effect of the pattern and fraction of missing values on two different approaches for fitting PARAFAC is studied. Based again on tests on real and simulated data sets, it is shown how PARAFAC is considerably more robust than bilinear models such as PCA to missing values and how relatively good estimates of the model parameters can be obtained with up to 70% missing values. Moreover, it is shown that the pattern of the missing values is far more important than their fraction in determining the magnitude of the associated artefacts and that the presence of such artefacts does not necessarily imply bad predictions when PARAFAC models are used for calibration purposes.

In Paper III, the properties of the column-wise Khatri-Rao product are investigated as a means to improve the computational efficiency of several nonlinear least squares solvers. In particular, fast routines have been obtained for computing the Jacobian and its products with a vector, the Hessian and the gradient of the PARAFAC model. Furthermore,

a method that can greatly reduce the computational load of a single ALS iteration and a relatively efficient procedure to solve the exact line-search problem for PARAFAC for the general N-way case have been devised.

In the second part of this work, the effect of so-called shift in chemical data (e.g., retention time shift in chromatography) on the fitting of low-rank multilinear models has been studied. In particular, two algorithms based on dynamic programming (namely, Correlation Optimised Warping – COW, and Dynamic Time Warping – DTW) are investigated as a means to remove such an effect. In the introductory part as well as in Paper IV the two algorithms are described also in connection to the numerous methods that have been proposed for the alignment of shifted signals. Moreover, in Paper IV, the effect of different constraints on the basic warping algorithms has been analysed in relation to PCA modelling of chemical data obtained through gas chromatography with flame ionisation detection (GC-FID). As a result of this study, it was concluded that rigid slope constraints are necessary in DTW as well as in COW in order to successfully employ PCA on this type of chromatographic data.

In Paper V, COW is used in combination with PCA for an oil fingerprinting study on data obtained with gas chromatography and mass spectrometry detection (GC-MS). Moreover, a simple method based on standard quality assurance practices (i.e., the frequent measurement of a reference oil) has been used to find the optimal parameters for the warping procedure. In the same paper and in Paper VI, other preprocessing methods including numerical differentiation and normalisation have also been investigated in order to remove baseline problems and the effect of the concentration (as opposed to the composition) of the analysed samples.

Paper VI represents the link between the two subjects of this thesis and shows an application of DTW together with PARAFAC and PARAFAC2 models for a calibration problem. In particular, the data was obtained with high performance liquid chromatography with diode array detection (HPLC-DAD) and the purpose of the analysis was to determine the concentration of antibiotics in a biological matrix (kidney samples). Among others, it is shown that DTW cannot deal with heavily coeluting peaks that correspond to separate components in a PARAFAC model and that PARAFAC2 is preferable in this case.

The relevant aspects of this work span several levels in chemometrics. For example, the algebraic properties of a basic operation connected to PARAFAC have been studied to develop computationally efficient procedures (although still rather primitive from a numerical point of view) that were used to improve the existing methods for fitting this model. Thereafter, the usefulness of such methods has been tested in connection with common problems in chemometric data analysis (e.g., high collinearity or presence of missing values) on real as well as simulated data sets. Likewise, two alignment procedures based on dynamic programming have been studied to determine their connection and their suitability as preprocessing methods for chromatographic data. Subsequently, such algorithms have been employed for both exploratory studies and calibration problems.

Perhaps, the main conclusion of this thesis is that chemometrics would greatly benefit from a more strict collaboration with experts from other fields of investigation, and particularly with numerical analysts and computer scientists. For this reason, most of the MATLAB[®] routines used in the course of this thesis were made available to the scientific community on www.models.kvl.dk.

Table of contents

PREFACE.....	I
RESUMÉ	III
SUMMARY	VII
TABLE OF CONTENTS	XI
LIST OF ABBREVIATIONS AND SYMBOLS	XV
LIST OF PUBLICATIONS	XIX
1. INTRODUCTION	1
1.1 CONTENT OUTLINE	4
2. PARAFAC MODEL.....	5
2.1 MODEL STRUCTURE	5
2.2 UNIQUENESS	7
<i>2.2.1 Uniqueness of PARAFAC</i>	<i>9</i>
2.3 APPROPRIATENESS OF THE PARAFAC MODEL.....	13

2.4 DEVIATIONS FROM IDEALITY	17
2.4.1 Presence of non low-rank background components.....	17
2.4.2 Rank overlap	18
2.4.3 Zero signal, nonlinear responses and deviations from multilinearity	20
2.4.4 Presence of interactions	22
2.4.5 Shift	23
2.4.6 Missing values	24
2.5 FITTING CRITERIA	25
2.6 DIFFICULTY OF THE FITTING PROBLEM	29
2.7 ALGORITHMS FOR FITTING PARAFAC MODELS.....	32
2.7.1 Alternating Least Squares.....	34
2.7.2 Direct methods	38
2.7.3 Iterative Non-least squares methods	40
2.7.4 Hessian based methods.....	43
2.7.5 Preconditioned Conjugate Gradient.....	46
2.7.6 Compression	47
3. ALIGNMENT AND DYNAMIC PROGRAMMING	51
3.1 SHIFT	53
3.2 ALIGNMENT PROCEDURES	59
3.2.1 Models for the warping path	63
3.2.2 Landmarks.....	70
3.2.3 Distance choice	71
3.2.4 Optimisation algorithms for alignment.....	77
3.3 DYNAMIC PROGRAMMING	80

3.3.1 <i>Principals</i>	83
3.3.2 <i>Dynamic Time Warping</i>	85
3.3.3 <i>Correlation Optimised Warping</i>	102
4. CONCLUSIONS AND PERSPECTIVES	107
4.1 PARAFAC.....	108
4.2 DYNAMIC PROGRAMMING AND SHIFT.....	111
REFERENCE LIST.....	117
PAPERS	135
I. A comparison of algorithms for fitting the PARAFAC model.....	137
II. PARAFAC and missing values.....	175
III. Use of the properties of the Khatri-Rao product for the rapid computation of Jacobian, Hessian and gradient of the PARAFAC model.....	195
IV. Correlation Optimized Warping and Dynamic Time Warping as preprocessing methods for chromatographic data.....	225
V. Chemical fingerprinting of petroleum biomarkers using time warping and PCA	239
VI. Signal preprocessing and modeling in chromatography	247

List of Abbreviations and Symbols

\odot	Column-wise Khatri-Rao product
\otimes	Kronecker product
2FD	two-Factor Degeneracy
ALS	Alternating Least Squares
ALS-SI	Alternating Least Squares with Single Imputation
ASD	Alternating Slice-wise Decomposition
CANDECOMP	CANonical DECOMPosition
CANDELINC	CANonical DEcomposition with LINear Constraints
COW	Correlation Optimised Warping
CPM	Continuous Profile Model
DECRA	Direct Exponential Curve Resolution Algorithm
DP	Dynamic Programming
DTLD	Direct Trilinear Decomposition
DTW	Dynamic Time Warping
EEM	Excitation-Emission Matrix
\mathcal{F}	Warping path
g	Gradient vector
GC-FTIR	Gas Chromatography-Fourier Transform Infrared

GC-GC	tandem Gas Chromatography
GC-MS	Gas Chromatography with Mass Spectrometry detection
GEMANOVA	GEneralized Multiplicative ANalysis Of Variance
GRAM	Generalized Rank Annihilation Method
H	Hessian matrix
HMM	Hidden Markov Models
HPLC-DAD	High Performance Liquid Chromatography with Diode Array Detection
INDAFAC	INcomplete DAta paraFAC
IV	Indicator Variable
J	Jacobian matrix
LC-MS	Liquid Chromatography with Mass Spectrometry detection
LM	Levenberg-Marquardt
LS	Least Squares
MCR	Multivariate Curve Resolution
ML	Maximum Likelihood
MS-MS	tandem Mass Spectrometry
MS-SIM	Mass Spectrometry with Selected Ion Monitoring
NBRA	Non-Bilinear Rank Annihilation
NIPALS	Nonlinear Iterative Partial Least Squares
NMR	Nuclear Magnetic Resonance
PAFFT	Peak Alignment by Fast Fourier Transform
PARAFAC	PARAllel FACtor analysis
PARAFAC-PDE	PARAFAC with Penalty Diagonalization Error
PARALIND	PARAFAC with LINear Dependencies
PARS	Peak Alignment using Reduced Set mapping
PCA	Principal Component Analysis
PCG	Preconditioned Conjugate Gradient
PLF	Partial Linear Fit
PLS(R)	Partial Least Squares (Regression)
PMF3	Positive Matrix Factorisation for 3-way arrays
PTW	Parametric Time Warping
RBL	Residual BiLinearization

RMSE	Root Mean Squared Error
RMV	Randomly Missing Values
SMCR	Self Modelling Curve Resolution
SMS	Systematically Missing Spectra
SVD	Singular Value Decomposition
SWATLD	Self-Weighted Alternating Trilinear Decomposition
UV-VIS	UltraViolet-Visible light
WLS	Weighted Least Squares
Z	Khatri-Rao product of the loading matrices of a PARAFAC model; if \mathbf{Z}_{-n} , the n -th loading matrix is excluded: $\mathbf{Z} \equiv \mathbf{A}_N \odot \dots \odot \mathbf{A}_1$ $\mathbf{Z}_{-n} \equiv \mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \odot \dots \odot \mathbf{A}_1$

List of Publications

Paper I

A comparison of algorithms for fitting the PARAFAC model, G. Tomasi and R. Bro, Computational Statistics and Data analysis, 50, 2006, 1700-1734.

Paper II

PARAFAC and missing values, G. Tomasi and R. Bro, Chemometrics and Intelligent Laboratory Systems, 75, 2005, 163 -180.

Paper III

Use of the properties of the Khatri-Rao product for rapid computation of Jacobian, Hessian and gradient of the PARAFAC model, G. Tomasi, in preparation.

Paper IV

Correlation Optimized Warping and Dynamic Time Warping as preprocessing methods for chromatographic data, G. Tomasi, F. van den Berg and C. A. Andersson, Journal of Chemometrics, 18, 2004, 231-241.

Paper V

Chemical fingerprinting of petroleum biomarkers using time warping and PCA, J. H. Christensen, G. Tomasi and A. B. Hansen, *Environmental Science and Technology*, 39, 2005, 255 – 260.

Paper VI

Signal preprocessing and modelling in chromatography, I. García, R. Bro, G. Tomasi, L. Sarabia and M.C. Ortiz, in preparation.

Additional papers by the author not included in the thesis

Integrated methodology for forensic oil spill identification, J. H. Christensen, A. B. Hansen, G. Tomasi, J. Mortensen, and O. Andersen, *Environmental Science and Technology*, 38, 2004, 2912-2918.

CuBatch, a MATLAB interface for n-mode data analysis, S. Gourvénec, G. Tomasi, C. Durville, E. Di Crescenzo, C.A. Saby, L. D. Massart, R. Bro and G. Oppenheim, *Chemometrics and Intelligent Laboratory Systems*, 77, 2005, 122-130.

Dioxin screening in fish product by pattern recognition of biomarkers, M. Bassompierre, G. Tomasi, L. Munck, R. Bro and S. Balling Engelsen, *Chemosphere*, accepted.

1. Introduction

Chemometrics is a relatively recent branch of science that has received great impetus in the last twenty years from the development of new analytical techniques as well as powerful mathematical and statistical methods. The process was also made possible by the increasing power and storage capacity of personal computers, which are now capable of handling sets of data and of applying algorithms that only a few years ago were limited to computation centres and mainframes. One of the main consequences of such developments has been the constant increase of size and complexity of data sets. On the one hand, automation allows the analysis of a large number of samples while speed and resolution of analytical instruments allow measuring hundreds or thousands of variables per sample in a reasonable time. On the other, new modes (ways) are added to data arrays when distinct analytical procedures are combined in so-called 'hyphenated methods' [e.g., Gas Chromatography with Mass Spectrometry detection (GC-MS), High Performance Liquid Chromatography with Diode Array Detection (HPLC-DAD), etc.] or for certain techniques such as fluorescence, which may yield matrices (namely Emission/Excitation Matrices - EEM) of data for each analysed sample or even three-way arrays (e.g., Emission/Excitation/Lifetime).

One of the most favoured approaches in dealing with multivariate and multi-way data is to approximate the systematic variation by linear combinations of a relatively small number of parameters representing underlying phenomena (factors/components). The foremost example in this sense is Principal Component Analysis (PCA), whereby data are modelled by bilinear (and mutually orthogonal) components. One of the main advantages of such techniques is that the number of variables to look at is reduced to only a few, simplifying interpretation and removing the (often confusing) non-systematic variation. However, even though bilinear models such as PCA and other related methods like Partial Least Squares (PLS) are extremely powerful, they may still be redundant when applied to matricised multi-way arrays, which often present additional structural relations between variables that may help reduce the number of estimated parameters and further facilitate interpretation. This advantage is not limited to exploratory or qualitative studies, but has very direct consequences on quantitative analyses. The more parsimonious multi-way methods have proved to be more robust than their bilinear counterparts, provided that the chosen model structure is adequate for describing the sources of systematic variation.

Among multi-way methods, PARAFAC is extremely appealing for various reasons. Firstly, upon convergence to the least squares solution, the PARAFAC solution is unique under rather mild conditions, i.e., it can be rotated only at the price of yielding a model that does not fit the data equally well. Secondly, it is structurally simpler and more parsimonious than other models (e.g., the Tucker models), which makes interpretation easier and possibly enhances robustness. Finally, for certain types of data (e.g., EEM fluorescence), the underlying physical model can be approximated by a PARAFAC model.

In spite of its conceptual simplicity, the fitting of PARAFAC to real data sets is often troublesome. Over the years, several methods have been proposed for this task, each with advantages and drawbacks. Paper I is a review of some of the most interesting ones. Of the many reasons for the slowness of the fitting procedure, Paper II analyses the effect of fraction and pattern of missing values on the performances of two different algorithms and on the quality of the corresponding solutions.

Size of data sets, model complexity (i.e., the number of components in the model) and consequently number of estimated parameters hinder the application of the most effi-

cient algorithms in optimisation theory to the problem of fitting the PARAFAC model. This remains true even when the ever increasing performances of modern computers are taken into account and, if one considers that certain validation techniques (e.g., jack-knifing) require the fitting of a considerable number of separate models, it is evident how great the need is for computational efficiency for this specific problem. In Paper III, several algebraic expressions are introduced that exploit the inherent structure of the PARAFAC model and speed up certain key steps for optimisation methods such as Gauss-Newton, Newton and Preconditioned Conjugate Gradients.

Not all data that could potentially be described by low-rank multilinear models fulfill the most basic assumption for low-rank multilinearity, i.e., that each variable describes the same phenomenon in all samples. Exemplary in this sense is the retention-time shift problem for chromatographic data. When analysing this type of data, the concentrations of the constituents related to the single peaks need to be determined before any chemometric method is applied to account for the fact that in different runs the position of the same peak may vary. Especially when complex systems are investigated and whenever the system is not well known beforehand (i.e., in exploratory studies), this step is extremely time consuming and implies a loss of significant information because not all peaks can easily be resolved and/or quantified. Per contra, a correct alignment of the chromatograms of different samples would render most of the tools currently available for multivariate data analysis, prominently PCA and the other multilinear models, directly applicable to chromatograms without a quantitative (or semi-quantitative) intermediate step. As no information would be rejected before the data analysis starts, such an approach is particularly attractive in exploratory terms and would bear the non-secondary advantage of a large reduction of time expense and degree of subjectivity for the data analysis.

A number of proposals have appeared in recent years to correct for shift or more general shape-changes. One of the most promising, or at least most popular, is warping. In the second part of this thesis, two warping algorithms (Correlation Optimised Warping and Dynamic Time Warping) are studied with specific reference to chromatographic data (Paper IV). Furthermore, they are applied to complex hydrocarbon mixtures prior to Principal Component Analysis for pattern recognition purposes (Paper V) and in combination

with PARAFAC and PARAFAC2 for the quantitation and confirmation of drug residues in biological samples (Paper VI).

1.1 Content outline

The main part of the thesis is divided into three chapters.

Chapter 2 contains an introduction to PARAFAC modelling of multi-way chemical data. Useful characteristics like uniqueness and adherence to the underlying physical model of certain types of data are outlined in sections 2.2 and 2.3, while section 2.4 is dedicated to possible deviations from the ideality (e.g., the presence of missing values – also discussed in Paper II). Sections 2.5 and 2.6 briefly discuss some issues related to the use of different fitting criteria and the problem of the proper characterisation of data sets used for testing different algorithms. Finally, a review of some of the most successful methods for fitting the PARAFAC model to N -way data sets is presented in section 2.7 based on the results in Paper I, II and III. Throughout the chapter, special emphasis will be given to the advantages in terms of notation and speed obtained for PARAFAC fitting algorithms through the use of the Khatri-Rao and Kronecker products.

Chapter 3 describes the shift problem (section 3.1) with specific reference to chromatographic data and the fitting of PCA and PARAFAC models in combination with some preprocessing method aimed at removing shift from signals. The principals of alignment algorithms are introduced in section 3.2, while section 3.3 is dedicated to Dynamic Programming and to two alignment algorithms based on it (namely *Correlation Optimised Warping* and *Dynamic Time Warping*). In the papers related to this chapter, examples will be given of the application of these two methods for exploratory data analysis of food process data (Paper IV), environmental data (Paper V) and for determination of drug residues in biological samples (Paper VI).

Finally, Chapter 4 presents the conclusions and perspectives.

2. PARAFAC model

2.1 Model structure

The principles of the PARAFAC model date back to Cattell⁵⁵, who introduced the concept of “parallel proportional profiles”, but the current mathematical formulation and main algorithm were proposed independently by Harshman¹¹⁴ and by Carroll and Chang⁵¹, the latter under the name of CANDECOMP (from CANonical DECOMPosition).

The basic rationale of the PARAFAC model is that the systematic variation occurring in an N -way data set $\underline{\mathbf{X}}$ of dimensions $I_1 \times \dots \times I_N$ can be adequately described by an N -linear model expressed as:

$$x_{i_1 i_2 \dots i_N} \cong \sum_{f=1}^F \prod_{n=1}^N a_{i_n f}^{(n)} \quad (2.1)$$

where $a_{i_n f}^{(n)}$ denotes the model parameters, i_n identifies the running index for the n -th mode and F is the number of fitted factors (components). The model parameters are typically grouped in loading matrices:

$$\mathbf{A}_n = \left\{ a_{i_n f}^{(n)} \mid i_n = 1 \dots I_n, f = 1 \dots F \right\},$$

one referring to each mode. For a three-way array $\underline{\mathbf{X}}$ of dimensions $I \times J \times K$, a PARAFAC

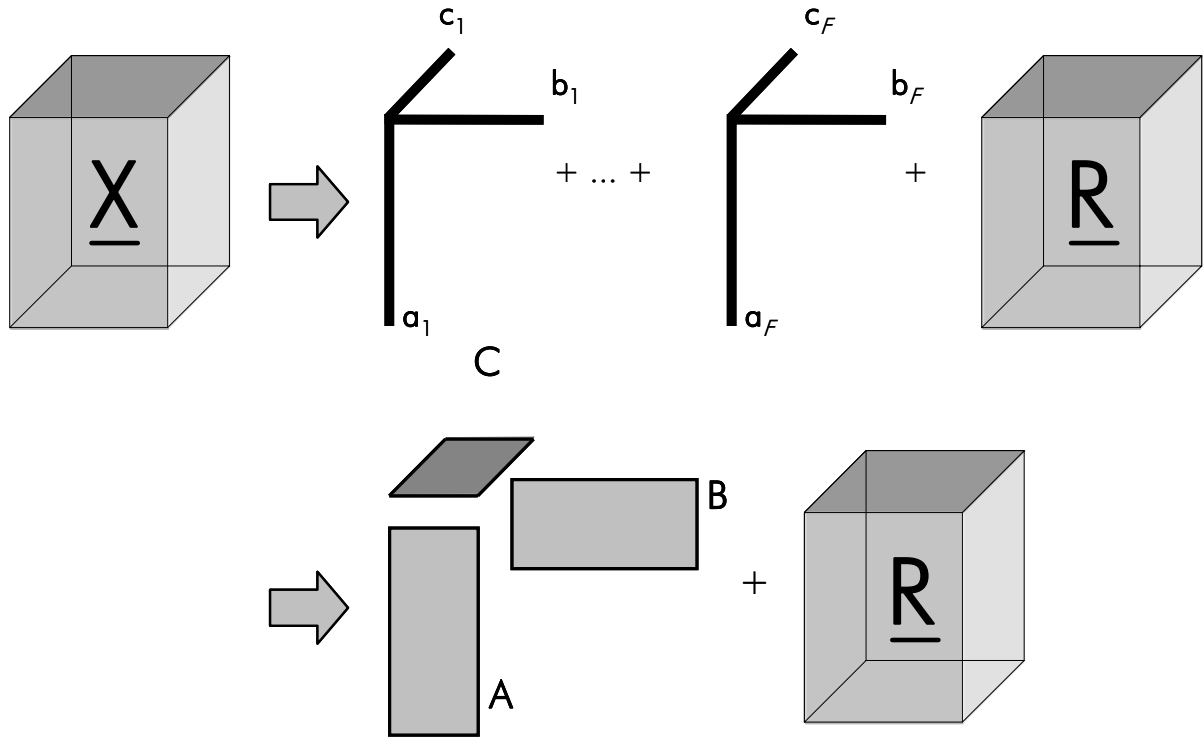


Figure 2.1. Graphical representation of the PARAFAC decomposition of a 3-way array $\underline{\mathbf{X}}$. The loading matrices $\mathbf{A} = [\mathbf{a}_1 \ \cdots \ \mathbf{a}_F]$, $\mathbf{B} = [\mathbf{b}_1 \ \cdots \ \mathbf{b}_F]$ and $\mathbf{C} = [\mathbf{c}_1 \ \cdots \ \mathbf{c}_F]$ have dimensions $I \times F$, $J \times F$ and $K \times F$ respectively.

model of rank F can be visualised as in Figure 2.1.

Note that the expression ‘ N -linear for an N -way array’ is more specific to PARAFAC than just multilinear; for example, Tucker models are multilinear, but may involve the products of a number of terms that is different from N for an N -way array^{178,278}. Since this chapter is dedicated to PARAFAC, the two terms are used as synonyms.

Several alternative notations, often functional to the algorithm employed for the fitting, have been developed over the years to express this model³¹. One that proved particularly useful for its compactness and to obtain efficient implementations^{P1-P111} expresses the model in terms of the matricised N -way array and loading matrices^{31,P111}:

$$\mathbf{X}^{(I_1 \times I_2 \cdots I_N)} \cong \mathbf{A}_1 (\mathbf{A}_N \odot \cdots \odot \mathbf{A}_2)^T, \quad (2.2)$$

where the superscript $(I_1 \times I_2 \dots I_N)$ refers to the way the array is matricised^{31,154,PIII} and \odot denotes the column-wise Khatri-Rao product²⁴⁴.

The smallest number of PARAFAC components whose sum is exactly $\underline{\mathbf{X}}$ is said to be the rank of $\underline{\mathbf{X}}$ ^{69,93,165,278}. This definition is the natural extension of one of the possible ways to define the rank of a matrix²⁷⁸. However, in spite of the similarities between ranks of N -way arrays and matrices, finding the rank of a multi-way array has proved to be a rather elusive task and some sparse and not general results are available only for three-way arrays^{93,166,292,293}. For example, the rank of a multi-way array is not upper-bounded by any of its dimensions and the rank of arrays of purely random numbers is not necessarily the theoretical maximum rank that an array of similar size would allow²⁷⁸. Thus, the concept of *typical rank* has been introduced to denote the rank of almost all the arrays of specific size^{292,294}. But even the typical rank need not be unique²⁹⁴, and for certain dimensions more than one rank can arise with non-zero probability from random numbers (e.g., $2 \times 2 \times 2$ arrays can have both rank 2 and 3 with non-zero probability in the real-valued case).

The question of the rank of a multi-way array is relevant, e.g., to establish the number of degrees of freedom for a statistical evaluation of the results^{270,278}. However, for practical purposes, the general assumption for real data sets is that the rank of the systematic part of the variation (which in chemometrics is often referred to as *pseudo-rank*²⁷⁸) is relatively small compared to the actual rank of the array (i.e., its typical rank) because of the presence of noise or other disturbances and because of the redundancy of the measurements.

2.2 Uniqueness

Multilinear models are in general not unique. However, some of the indeterminacies that cause non-uniqueness do not affect the interpretation of the model or its application apart from some minor practical aspects. Four types of ambiguities can be found in multilinear models: rotational, scaling, permutational and sign^{278,291}.

The last two are trivial and can usually be removed by setting appropriate conventions. Thus, permutational indeterminacy can be fixed by sorting the factors according to their norms, and sign indeterminacy can be eliminated by enforcing a positive sign for the largest element or for the sum of the third powers of all elements in all but one of the loading vectors forming a factor^{278,PI}. Both these ambiguities are in fact irrelevant for the convergence of the fitting algorithms and are typically addressed only once convergence is attained.

The scaling indeterminacy^{PI} may have to be explicitly addressed depending on the algorithm being used and its purpose. For example, it leads to rank deficiency of the Jacobian (i.e., the matrix holding the value of the first derivative of the residuals with respect to each model parameter¹⁹⁵) and thus to the lack of global convergence of fitting algorithms where an update for all parameters is reckoned in a single step at each iteration^{23,193,224} [e.g., *Gauss-Newton* (GN)^{224,PI-PIII} and *Preconditioned Conjugate Gradients* (PCG)^{225,322,PIII}]. Likewise, the Jacobian's rank deficiency (and the scaling indeterminacy) has to be removed when variance estimates for the model parameters are sought^{190,200,224}. On the contrary, scaling indeterminacy as such does not prevent convergence of most alternating fitting algorithms [e.g., *Alternating Least Squares*^{31,51,114} (ALS) and *Alternating Slice-wise Decomposition* (ASD)¹³⁹]. Hence, with the relevant exception of *Self-Weighted Alternating Trilinear Decomposition* (SWATLD)^{57,PI}, it is sometimes disregarded during the fitting. The scaling indeterminacy can be directly eliminated by fixing the norm of all loading vectors except for one in each factor^{190,225} or as a consequence of some constraints (e.g., closure²⁷⁹). Alternatively, the problems associated with the scaling indeterminacy can be removed indirectly using the same solutions employed to deal with rank deficiency, e.g., by means of a damping parameter in the Levenberg-Marquardt algorithm^{PI} or of an additional regularisation term in the loss function^{224,PI}. In any case, it is important to point out that the adopted scaling convention can have a remarkable impact on both numerical stability and speed of convergence of all fitting algorithms (thus including ALS) and the issue should be addressed carefully^{23,102,131,PI}.

Rotational indeterminacy means that there exist an infinite number of solutions that can be obtained from one another through non-trivial linear transformations (i.e., not lim-

ited to combinations of permutation and scaling matrices) and fit the data equally well²⁷⁸. When a model has no rotational indeterminacy, it is said to be (essentially) 'unique'^{278,294} or 'identifiable'^{224,225}. Even if numerous algorithms exist to find the best rotation according to a predefined criterion, rotational ambiguity often precludes a straightforward interpretation of the estimated parameters because it is necessary to ensure that the employed criterion is adequate for the problem at hand and is not only conceptually or visually appealing^{31,121}. Moreover, especially in exploratory studies, where the underlying model may not be known in advance, different criteria may lead to rather different conclusions, all equally acceptable^{31,121}.

In a number of cases, (essential) uniqueness can be attained by constraining the solution. However, while constraints (e.g., non-negativity, unimodality and selectivity³¹) are often used to yield estimates that are consistent with an underlying physical or chemical model, only a few of them (e.g., selectivity and symmetry³¹) are sufficient to yield unique solutions. In particular, non-negativity and unimodality can, in most cases, only produce feasible ranges for the factors^{100,173,278}. In general, for models that are not structurally unique, additional external knowledge about the underlying process (e.g., the spectrum of one or more of the chemical species involved) is necessary to obtain univocally interpretable solutions^{22,110,240,296}.

2.2.1 Uniqueness of PARAFAC

The PARAFAC model can be considered both as an extension to the N -way case of various bilinear models [e.g., *Principal Component Analysis* (PCA)^{31,278} and several *Self-Modelling Curve Resolution* (SMCR)^{100,173,291}] and as a special case of several Tucker models for arrays with an identical number of modes²⁷⁸. However, unlike them, the PARAFAC solution is essentially unique (hereafter often just termed unique) under rather mild conditions pertaining to the linear independence of the loading vectors²⁷¹. This uniqueness property was the main reason behind the introduction of the concept of parallel proportional profiles^{55,114,121,278}. The uniqueness conditions for the PARAFAC model are discussed at length in numerous publications^{114,116,158,166,190,271,295}, a general sufficient one is:

$$\sum_{n=1}^N k_n \geq 2F + N - 1, \quad (2.3)$$

where k_n (referred to as the k -rank) is the largest value of k such that every subset of k columns of \mathbf{A}_n has full column rank²⁷¹.

One can observe from (2.3) that simple rank deficiency of the loading matrices (e.g., arising from closure effects due to mass balance^{278,279}) is compatible with uniqueness. Conversely, uniqueness is lost in the three-way case if two columns in the same loading matrix are proportional (i.e., the matrix has k -rank one). However, even when the decomposition is not unique, some of the loading vectors can still be uniquely resolved^{36,116,158}; this allows for example the quantitative determination of the analytes of interest even in presence of multiple interferents^{158,278,280} or when the contribution of the single constituent does not have rank one^{282,327}.

It is significant that a PARAFAC model (intended as a low-rank N -linear decomposition for an N -way array) would still be appropriate even in the presence of proportional columns (cf. section 2.3), but, since the model is not identifiable, the algorithms may fail to converge to meaningful solutions^{110,157,282}. In this sense, loss of uniqueness may arise also on purely numerical basis and, if the factors are very collinear, uniqueness may be weakly determined^{129,163}. In this case, the enforcement of constraints (e.g., equality between some of the loading vectors or orthogonality) may be of some help to yield more interpretable solutions at the cost of a slightly worse fit^{36,86,163}.

The restricted Tucker models that are sometimes used in this situation operate exactly in this fashion as they are mathematically identical to PARAFAC models with equality constraints^{36,121}. With relatively few modifications (e.g., the projection of gradient and Hessian matrix on the null space of the constraint matrix¹⁰²), a Levenberg-Marquardt algorithm^{PI,PII} can be employed for these types of problems and may constitute an alternative to the ALS algorithm typically used¹⁵⁷.

Demonstrating analytically that a certain model is structurally unique involves rather complex and lengthy reasoning and general results are difficult to obtain^{157,165,278,295,296}. Even condition (2.3) has been recently found to be necessary and sufficient only when the

rank of the array is at most equal to three²⁹⁵. Numerical verifications are also possible and can be used when theoretical results are not available^{159,295}. Most of the approaches entail fitting the same model repeatedly to the same data set with different starting points or to different arrays of identical rank obtained from the original one through a selection of samples or variables (i.e., split-half analysis¹¹⁷). However, these methods can be computationally expensive and do not guarantee that the model is unique: most algorithms may converge to local minima and the solutions are affected by other aspects such as deviations from multilinearity, missing values and noise³⁶. Therefore, it is possible for the algorithm to converge always to the same solution even though the underlying model is not unique³⁶. Condition (2.3) can also be used to verify the identifiability of a specific solution, but this option can be rather expensive if F is large because the rank of all subsets of 2 to F columns of each loading matrix may have to be calculated in order to ascertain the k -rank of each of the loading matrices¹⁹⁰.

Since any right singular vector of the Jacobian corresponding to a zero singular value is a direction (and thus a transformation) along which the parameters can be updated without any loss of fit²²⁵, the number of zero singular values of the Jacobian has been advocated as a means to establish local uniqueness (i.e., pertaining to a specific minimum)²²⁵. Notice that in this case one must account for the fact that unless additional constraints are enforced, scaling indeterminacy leads to a rank deficient Jacobian with at least $(N-1)F$ trivially zero singular values^{31,225,PI}. Consequently, concise expressions for computing the Jacobian and its cross product may be some help in establishing uniqueness, both analytically and for practical purposes. For example, one can observe that the rank of the partition of the Jacobian relative to the N -th mode^{PIII}

$$\mathbf{J}_n = -\mathbf{M}_n (\mathbf{I}_{I_n} \otimes \mathbf{Z}_{-n}), \quad (2.4)$$

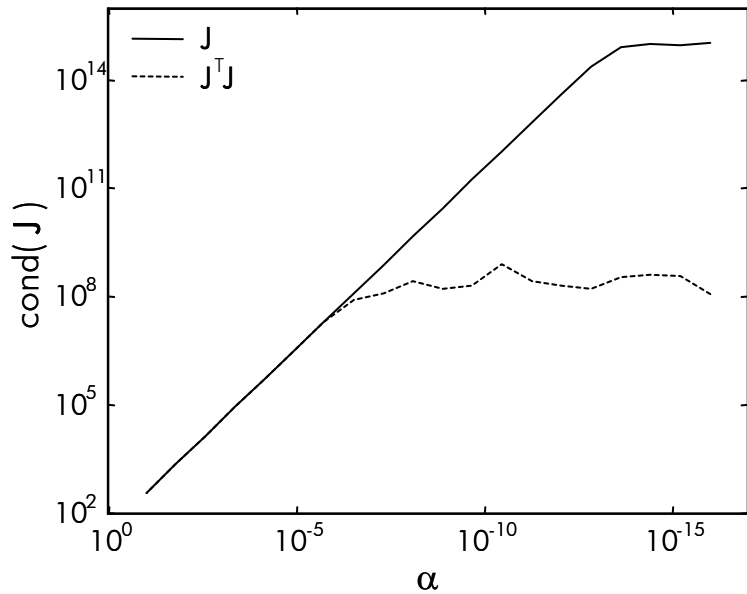
where \mathbf{M}_n is an appropriate permutation matrix, \mathbf{I}_{I_n} is an $I_n \times I_n$ identity matrix, \mathbf{Z}_{-n} denotes the Khatri-Rao product of all the loading matrices but the n -th and \otimes is the Kronecker product¹⁹⁵, depends solely on $(\mathbf{I}_{I_n} \otimes \mathbf{Z}_{-n})$ because \mathbf{M}_n is a permutation matrix and thus has full rank¹⁹⁵. Since $\text{rank}(\mathbf{A} \otimes \mathbf{B}) = \text{rank}(\mathbf{A})\text{rank}(\mathbf{B})$ ¹⁹⁵, if \mathbf{Z}_{-n} does not have full column rank, there are at least I_n additional zero singular values in the Jacobian. This necessary

condition for uniqueness was originally demonstrated in an alternative way by Liu & Sidiropoulos¹⁹⁰ and can be considerably less expensive to verify than (2.3) for large F . Moreover, since the rank of \mathbf{J} is the same as that of $\mathbf{J}^T\mathbf{J}$ and the latter matrix can be computed without the explicit formation of \mathbf{J}^{PIII} , the number of non-zero singular values (or eigenvalues) of $\mathbf{J}^T\mathbf{J}$ could also be used for problems that are too large for their Jacobian to be explicitly calculated. The main shortcoming for such an approach is that the squaring of the singular values of the cross-product compared to the original matrix increases the risk of diagnosing rank deficiency on a purely numerical basis (Figure 2.2)²³.

It is worth mentioning here that the same numerical techniques to assess uniqueness based on the Jacobian are valid and applicable to equality constrained PARAFAC models^{157,278} or if some of the array's elements are missing^{224,PII}.

Since an optimisation problem subject to linear equality constraints can be transformed to an unconstrained problem upon the projection of \mathbf{J} on the null space of a suitably defined constraints matrix¹⁰², any right singular vector associated to a zero singular value of the projected Jacobian would correspond to a direction that does not violate the equality constraints and leads to a different solution of equal fit. This can be rather straightforward to implement when very simple equality constraints are needed whereby one or more model parameters are fixed (e.g., to one in order to remove offsets in one or

Figure 2.2. Behaviour of \mathbf{J} 's condition number as a function of uniqueness. Two columns of \mathbf{A}_3 are defined as \mathbf{u} and $(1-\alpha)\mathbf{u} + \alpha\mathbf{v}$. For a 3-way array and α sufficiently small, uniqueness is lost and \mathbf{J} becomes rank deficient (scaling indeterminacy is fixed by imposing that the 2-norm of the columns for two loading matrices be constant).



more dimensions^{38,42,120,310}, to zero to impose selectivity³¹, or to a specific value when some model parameters are known in advance^{110,248}). Projecting the Jacobian on their null space simply corresponds to removing the columns relative to these parameters from **J**.

With respect to arrays with missing elements, these are given zero weights and the Jacobian for the associated model is obtained by removing the rows of **J** that correspond to the missing values. In this case, bad conditioning (and sometimes non-uniqueness) may ensue purely from the number and pattern of the missing values^{P11}.

In terms of uniqueness, there are obvious advantages in increasing the number of modes and in treating *N*-way arrays as such, without decreasing their order by collapsing one or more modes (e.g., limiting the number of slabs in such modes to one) or by simple rearrangement (e.g., matricising). For example, the decomposition can still be unique even if one of the loading matrices has proportional columns, when the array is at least of order four and the *k*-rank in the other modes is sufficiently high²⁷¹. Consequently, if another dimension is added to the array (e.g., the fluorescence decay), two fluorophores could still be uniquely identified and quantified even if they have identical emission or excitation spectra^{6,216}. Likewise, in the four-way case, it is possible to uniquely resolve the spectra of constituents having the same concentration profile in all the samples^{6,36,158,176}, something that is likely to happen if several fluorophores are attached to the same macromolecule in biological samples²⁵³. Some relatively novel curve resolution methods [e.g., the *Direct Exponential Curve Resolution Algorithm* (DECRA) and its generalisation, the ‘slicing’ procedure^{77,230,335}] take direct advantage of this by rearranging the data matrix and “artificially” increasing the order of the data array^{22,229,248,335}.

2.3 Appropriateness of the PARAFAC model

Every *N*-way array can be decomposed exactly in a PARAFAC model of sufficiently high rank^{69,93,165,278}. In this sense, a PARAFAC model is by definition appropriate. However, as mentioned in section 2.1, the typical rank of an array is affected by noise and is commonly much larger than its pseudo-rank²⁷⁸. Hence, the appropriateness of PARAFAC

is generally intended with respect to low-rank models and a PARAFAC model can be said to be appropriate if it approximates reasonably well the underlying phenomena associated to the data. From this point of view, it is necessary that these phenomena be also N -linear for an N -way array (for $N > 2$) and that no interactions exist between the underlying factors^{39,163,177}.

Spectroscopy, chromatography or their combination are typical cases where this type of data arises^{26,177,263,278}; and fluorescence spectroscopy provides perhaps the ideal example to show how a PARAFAC model can correspond to the underlying physical model¹⁷⁷. For sufficiently diluted solutions, Beer's law holds²⁷⁷ and the amount of light absorbed by a certain molecule (chromophore) at a specific wavelength can be expressed as:

$$x_i = \chi \varepsilon_i, \quad (2.5)$$

where χ and ε_i are respectively proportional to the chromophore's concentration and to its molar extinction coefficient at the i -th wavelength (i.e., the fraction of incident light absorbed per unit of concentration). For fluorescent species, chromophores are renamed fluorophores and part of the energy they absorb is reemitted by radiative decay²⁷⁷. Equation (2.5) can be modified accordingly, and the intensity of the light emitted by a fluorescent species at the j -th emission wavelength can be expressed as:

$$x_{ij} = \chi \varphi \varepsilon_i \lambda_j, \quad (2.6)$$

where, λ_j indicates the fraction of fluorescence photons emitted at the j -th emission wavelength and φ denotes the fluorescence quantum yield, i.e., the fraction of absorbed energy that is reemitted by radiative decay. If the solution is sufficiently diluted and there are no interactions between fluorophores, the signal from F different fluorophores is additive and equation (2.6) becomes:

$$x_{ij} \cong \sum_{f=1}^F \chi_f \varphi_f \varepsilon_{if} \lambda_{jf}, \quad (2.7)$$

where f identifies the fluorophore. Finally, when several solutions are analysed and by incorporating the quantum yield in the concentration term, one obtains for the k -th solution:

$$x_{ijk} \cong \sum_{f=1}^F \chi_{kf} \varepsilon_{if} \lambda_{if} . \quad (2.8)$$

Consequently, as equations (2.1) and (2.8) are identical, fitting a PARAFAC model to the measured intensities x_{ijk} corresponds to estimating the parameters for the physical model of the fluorescence signal. Moreover, the uniqueness property of PARAFAC models guarantees that, if model (2.8) is adequate, the correct number of factors (fluorophores) F is chosen and the global minimum of the loss function is attained, the estimated loading vectors for the f -th component are readily interpretable as concentration profiles (\mathbf{c}_f) and emission and excitation spectra (\mathbf{a}_f and \mathbf{b}_f respectively) of the f -th fluorophore^a. Many applications use EEM fluorescence data because they are reasonably simple and cheap to obtain, are well modelled by PARAFAC and are sufficient to solve a great variety of problems^{6,278,PI-PIII}.

Model (2.8) can be extended further. For example, one could take into account fluorescence decay^{47,171,201,255-257,268,300} or that quantum yield may be affected by the concentration of a quencher^{171,176,177,252,253,328}. If all these relations were considered at once, the underlying model would become¹⁷⁷:

$$x_{ijklm} \cong \sum_{f=1}^F \chi_{kf} \varepsilon_{if} \lambda_{if} \varphi_{lf} \tau_{mf} , \quad (2.9)$$

where φ_{lf} is the quantum yield at the l -th quencher concentration and τ_{mf} expresses the dependency on time. Likewise, one could use a spectrofluorometer to monitor a certain reaction in time^{216,217,289} or the elution from a chromatographic column^{10,11,34}.

Depending on the experimental conditions and the objectives of the analysis, various modes are often collapsed so that the data array becomes three-way and the final model

^a Given the permutational indeterminacy, in order to correctly assign a PARAFAC component to a fluorophore additional knowledge is necessary^{6,16}.

trilinear^{10,18,19,47,201,300}. However, such a reduction is often not necessary and several advantages can derive from maintaining the original number of modes or from increasing it when possible^{26,271} (cf. section 2.2). The reasons for the relative lack of applications with arrays with more than three modes are not obvious²⁶, but most often depend on reproducibility problems^{34,268,278}, size of the fitting problem^{216,224,225} or specific aspects pertaining to the algorithms (e.g., the fact that some methods only work for three-way arrays^{263,264} or cannot be straightforwardly generalised to the N -way case^{31,57,82,190,PI}, or their use for higher order arrays is cumbersome or inefficient¹¹⁹). Furthermore, the feasibility of a certain analysis also depends on the right choice of the fitting algorithm, which is not at all obvious (cf. section 2.7)^{23,94,102,193,PI-PIII}.

For a great variety of data, the parameters of the underlying physical or chemical model can be estimated via the PARAFAC model. Only to cite some that do not regard fluorescence: UV-VIS absorption spectrometry^{74,79,111,180,269}, HPLC-DAD^{65,332}, GC-MS/LC-MS^{49,89,97,276}, GC-GC^{91,92,307}, spectroscopic monitoring of chemical reactions^{22,79,126}, low- and high-field Nuclear Magnetic Resonance (NMR)^{112,222,223}, Multichannel Evoked Potentials for electroencephalographic (EEG) analysis^{5,86,205}, and signal processing for blind source separation^{190,272,273}. In addition, when the underlying factors in one dimension are exponentials or sum of exponentials, it is possible to increase the order of the data array (e.g., to rearrange a matrix in a three-array that can be described through a trilinear model)^{22,216,229,248,335}. It is noteworthy that this represents an exception to the fact that while one can always reduce the order of an array (e.g., by matricising it), the opposite is not possible²⁶.

The PARAFAC model can also be used to study data arising from designed experiments^{37,38,123}. GEMANOVA³⁸ (*GEneralized Multiplicative ANalysis Of VAriance*) treats all factors as qualitative and do not regress the response variable on the design factors as standard ANOVA. Instead, it seeks to describe the data with few higher order interactions assuming a multiplicative model. In other words, it decomposes the N -way array (where N is the number of design factors) holding in each element the value of the response variable using F factors (where F is the desired number of interactions or main effects). The resulting model is simply a PARAFAC one whereby main effects or interactions of order

lower than N (when required) are obtained by fixing the loading vectors in some of the modes for selected components³⁸. However, it is significant that, theoretically, enforcing this type of constraints leads to loss of uniqueness (specifically, if two different factors/effects, have columns of ones in the same mode¹²⁰) and problems in convergence to meaningful solutions³⁸.

2.4 Deviations from ideality

Ideally, one should expect that the pseudo-rank of a data array be lower than the mathematical rank and mostly equal to the number of sources of variation in the data, which in chemometrics is often referred to as the chemical rank²⁷⁸. Hence, deviations from ideality can be defined as those conditions arising in certain data that require the fitting of a PARAFAC model whose rank is different from the actual number of underlying phenomena (e.g., number of constituents in a solution of different compounds). Any such departure has an effect on the extracted components and can affect the interpretation as well as the quality of the predictions in regression models. Thus, one should take special care in verifying that the underlying conceptual model follows the same assumptions as the PARAFAC one^{113,121,278}.

Deviations from the ideal are often encountered in practice. Some of them can be accounted for by constraining the PARAFAC model (directly or indirectly); some others require the use of different models^{6,39,67,177,282,PVI}.

2.4.1 Presence of non low-rank background components

Some types of real data contain features that cannot be adequately modelled by low-rank PARAFAC models⁶ (e.g., Rayleigh and Raman scatter in EEM fluorescence measurements¹⁷¹). Several ways of dealing with this problem have been devised: using weights^{41,141,248}, subtracting a standard^{218,328}, setting the corresponding values to missing^{15,16,32}, using of zeros^{15,60,297}, low rank background modelling and removal through bilinear factors^{27,58,207,248}.

Not all these methods work equally well²⁴⁸. Thus, blank subtraction is often not sufficient to remove all the background, thereby affecting loadings and quality of the predictions^{6,PII}, and not all weighting schemes yield optimal results²⁴⁸. Missing values can also be problematic as they can slow down convergence and induce substantial distortions in the loading matrices, especially for specific patterns (cf. section 2.4.6)^{6,297,PII}.

With respect to background modelling, the results are still not conclusive²⁴⁸. However, it is worth mentioning that by fixing one (or more) loading vector to a constant value, thereby removing the corresponding columns from the Jacobian (thus columns and rows from the Hessian and elements from the gradient), bilinear factors and offsets can be efficiently fitted through the Hessian based methods (cf. section 2.7.4)^{PI-PIII}.

2.4.2 Rank overlap

Especially for curve resolution purposes and second-order calibration^{b,26,278}, one of the assumptions is that the contribution of each constituent in a mixture can be modelled by one fitted PARAFAC component (which is again a rank one array). In this respect, the uniqueness property of PARAFAC (when it holds) bears important consequences, as no additional constraints are necessary to yield meaningful and readily interpretable solutions (cf. section 2.2). On the other hand, for several analytical methods [e.g., tandem mass spectrometry (MS-MS)^{327,334} or multidimensional NMR^{112,222,223}] this condition is violated and the pseudo-rank of the array can largely exceed its chemical rank (i.e., the number of analytes). In these instances, each analyte is typically associated to more than one factor and complete uniqueness is lost (for three-way arrays) because such components have the same concentration profiles^{157,158,327}. If no additional external knowledge is available, only partial uniqueness can be attained in these cases and only the components relative to analytes with rank one contribution can be uniquely resolved^{158,296}. A similar problem emerges when the loading matrices are rank deficient and condition (2.3) does not hold (e.g., as a consequence closure effects linked to mass balance in a chemical reac-

^b Intended as the process of generating a regression model that allows to compute the concentration of one or more analytes in a sample from a two-way array of recorded signals²⁶.

tion^{157,247}). In this case, although single components may be associated with only one analyte, the solution is no longer unique and ordinary PARAFAC solutions are often not satisfactory^{36,278}.

Theory and formulation of models are quite more complicated in case of rank overlap^{36,157,158,278,282}, but it is important to remark that the PARAFAC model in itself is still appropriate even for this type of problems. An F component PARAFAC model is the array of rank F that best approximates the data^{43,93} and no additional restrictions are necessary on any of the loading matrices for it to be appropriate. Thus, for example, if one considers the vectorised array which underlying model has structure $\mathbf{c}_1 \otimes \mathbf{b}_1 \otimes \mathbf{a}_1 + \mathbf{c}_1 \otimes \mathbf{b}_2 \otimes \mathbf{a}_2 + \mathbf{c}_3 \otimes \mathbf{b}_3 \otimes \mathbf{a}_3$, one can easily see that its pseudo-rank is 3 and that a PARAFAC model of rank 3 can adequately approximate it. Indeed, owing to non-uniqueness, external additional constraints (e.g., non-negativity or equality if some of the other loading vectors are known) may be required to obtain directly interpretable solutions^{6,31,35,40,281}. However, non-negativity and other constraints need not be imposed during the fitting procedure and could be applied as postprocessing in those cases when the subspaces spanned by two (or more) components are uniquely identified rather than the components themselves^{112,222,223,281}. Thus, the problem of rank overlap pertains to the algorithm used to fit PARAFAC and how (and which) additional constraints are imposed to yield uniqueness or to narrow the range of possible solutions rather than the appropriateness of PARAFAC models per se.

Several algorithms exist and can be employed with various degrees of success to solve this type of problems: *Non-Bilinear Rank Annihilation*^{247,327,334} method (NBRA), which is an extension of the *Generalized Rank Annihilation Method* (GRAM)²⁶⁴ to fit the PARAFAC model; *Residual BiLinearization* (RBL)^{221,247,326}, restricted Tucker models^{157,280,282}, *PARAFac with LINear Dependencies* (PARALIND) models^{36,c} and *Multivariate Curve Resolution* (MCR)^{281,290}, which are all fitted by means of an *Alternating Least Squares* (ALS) algorithm¹⁵⁷. Moreover, an algorithm to fit PARAFAC on three-way arrays with two propor-

^c The restricted PARATUCK2 model that has been used to fit this type of data belongs to the PARALIND family, which is a more general class³⁶.

tional columns in one mode was already proposed by Krijnen as a means to study uniqueness¹⁶³ (cf. section 2.2).

The implementation of linear equality constraints in a PARAFAC context has been recently investigated^{31,36}, but the subject of which algorithm is the most efficient has not been addressed. For example, a Levenberg-Marquardt algorithm for PARAFAC using compression ^{P_I - P_{III}} and Jacobian projection on the null space of a suitably defined constraints matrix¹⁰² could provide a valid alternative to ALS for ill-conditioned problems ^{P_I , P_{II}} . More in general, in numerical analysis there is a wealth of different options in terms of equality constrained optimisation^{102,194} that could be applied to the PARAFAC case and could exploit its structure and the gains in efficiency associated with the properties of the Khatri-Rao product ^{P_{III}} .

2.4.3 Zero signal, nonlinear responses and deviations from multilinearity

One of the basic assumptions of low-rank multilinear modelling is that the variation in all the vectorised sub-arrays obtained by fixing the indices in one or more modes can be approximated by a weighted sum of the Khatri-Rao (or Kronecker) products of the loading vectors for the modes whose indexes are not fixed¹⁷⁸. For example, this infers that the emission spectrum of a fluorescent species is required to remain the same at all excitation wavelengths and for all the samples, and that the kinetic profile for a specific analyte must not change between samples when a reaction is being monitored.

Several features may be present in the data that can violate this condition. For example, owing to the fact that there is no emission at wavelengths lower than the excitation wavelength¹⁷¹, the corresponding measurements in an EEM matrix are zeroes (or just noise) and the recorded emission spectra would depend on the wavelength of the exciting light (because the number of zeros changes at each excitation wavelength). Thus, the loadings in the emission and excitation mode would need to accommodate for the fact that their outer products (i.e., the fluorescence landscapes for the single constituent) have to be zero in the area of no emission and considerable distortions could ensue⁶. Similar

distortions in the signal may occur because of a nonlinear response of a sensor (e.g., because it is close to saturation)^g.

More or less subtle deviations from exact N -linearity (e.g., minor shape changes in the recorded emission spectrum between different EEM measurements of the same specimen³⁹ or peak shifts due to matrix effects in chromatography^{25,97}) are typical of experimental data and have been suggested as the reason for larger variances in the concentration estimates for calibration problems¹⁸⁴. These deviations may give rise to problems because they contribute to the variation in the data and need to be accounted for by any best low-rank approximation of the data array³⁹. Increasing the model's rank to account for these additional sources of variation and improve the results can hardly be considered a solution to this problem because the supplementary components would likely be too small in magnitude compared to the bulk of the signal, too collinear with the "true" factors in some of the modes or too unstable to be reliably isolated³⁹. Thus, the model's rank is almost necessarily wrongly specified with respect to this type of systematic variation and it seems plausible that these departures from low-rank N -linearity could be partly responsible for slow the convergence of some algorithms and for the emergence of two-Factor Degeneracies (2FD's)^d, both temporary (i.e., as in the so-called 'swamps') or in the final solution. Moreover, these deviations necessarily induce some alteration in the loading vectors and their presence often shows as dents, shoulders, or even spurious peaks in correspondence with other components (Figure 2.3)^{6,PII}. A decrease in the diagonality of the core (i.e., of the core consistency³⁹) associated to a specific PARAFAC solution has also been observed in this case, which hints a possible explanation for the relatively good performances of algorithms that yield solutions with higher core consistency^{PI}.

^d 2FD's are regions in the convergence process where two or more factors grow increasingly collinear maintaining opposite sign and almost cancelling each other's contribution^{PI,168,245}. Sometimes the degeneracies slowly disappear and the algorithm starts converging again at a higher rate (these regions are typically referred to as 'swamps')^{203,245}. 2FD's and swamps have been the subject of numerous studies^{203,224,226,245,347} and are perhaps the main driving force for continuous introduction new algorithms for fitting the PARAFAC model^{22,50,56,57,81,82,93,138,139,155,179,181,224,225,245,289,322,342,343,PI,PII}.

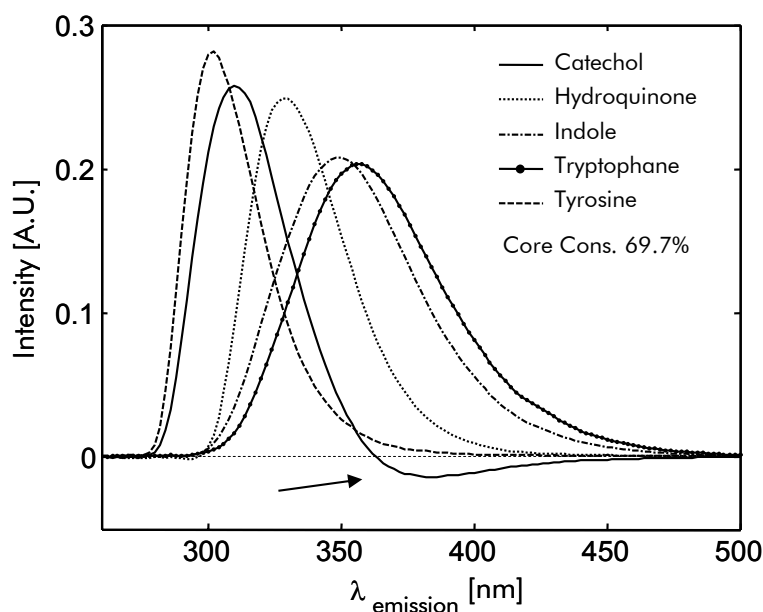


Figure 2.3 Emission spectra for EEM data set^{248,PII}. The concavity in the loading vector for catechol (pointed by the arrow) is due to subtle deviations from trilinearity. Note that the core consistency is relatively low.

One of the possible procedures to obviate the problems associated with the violation of N -linearity due to specific data elements is to treat them as missing values (or equivalently to assign them zero weights)^{6,8,PII}. However, pattern and fraction of missing values may themselves cause the formation of artefacts in the loadings and may have a considerable effect on the speed of convergence of the fitting algorithm^{PII}. Thus, more complex patterns of missing values and zeros have been devised recently in order to yield an acceptable trade off^{8,15,60,141,297}.

With respect to subtler deviations that cannot be ascribed to specific data values, it has been suggested that fitting according to a weighted least squares criterion (cf. section 2.5) may help reduce their effect¹²⁰. However, this possibility has not been thoroughly investigated.

2.4.4 Presence of interactions

A relevant condition for a (low-rank and unconstrained) PARAFAC model to be appropriate is that interactions between different underlying factors should be limited or absent^{32,119,177}. These can be real interactions between the physical elements corresponding to the PARAFAC components or may be artefactual. An example of the first type is the ‘inner-filter effect’ in fluorescence spectroscopy: the light emitted by a certain fluorophore

can be absorbed by another fluorophore in the same solution¹⁷¹. Per contra, specific patterns of missing values may allow interactions between factors that do not correspond to physical phenomena and which typically emerge during the fitting procedure^{6,P/I}.

Constraints often work well when interactions are artefactual; depending on problem and type of data, one could enforce non-negativity and/or unimodality⁶ on some loadings, can impute zeros in selected areas of the data array^{15,60,297} or can directly force certain model parameters to be zero³¹. It is noteworthy that the latter solution can be straightforwardly implemented in derivative based methods (cf. section 2.7) and entails a reduction in the size of the problem. Conversely, if a real interaction is occurring it may be necessary to opt for a different model than PARAFAC¹⁷⁷

2.4.5 Shift

Another possible violation of low-rank N -linearity may arise for shifted data, i.e., data whereby an identical phenomenon (e.g., a peak in a chromatographic run) occurs in separate samples (or, more generally, slabs) at different stages, or has different durations^{49,118,149}. In such cases, the change in the profiles between different slabs may become a relevant source of variation, entailing the use of additional components not related to the underlying phenomena with often detrimental effects on interpretation and usefulness of the model (cf. Chapter 3)^{9,196,344}. Typical examples of shift are found in chromatography^{34,49,91,P/IV-P/VI} and kinetic studies^{67,149}, but shift is common also for NMR^{299,316,337-339} and fluorescence signals¹⁷¹.

The shift problem can be pursued by modifying the model structure allowing for some deviations from strict multilinearity (e.g., PARAFAC2^{34,P/VI}, Tucker2⁶⁷, shifted factor models^{118,127,128} or constrained multivariate curve resolution^{67,265} – MCR) or by applying suitable preprocessing methods aimed at realigning the signal in the different slabs such as warping^{48,49} (cf. Chapter 3). An interesting alternative to these common strategies arises when the data array is at least of order four, because one could concatenate the slabs relative to the mode in which the shift occurs retaining at the same the uniqueness properties of PARAFAC. Per contra, if the original data set is three-way this procedure corre-

sponds to matricising it, thus resorting to bilinear models and likely loosing uniqueness^{67,290}.

2.4.6 Missing values

Missing values are a fairly common occurrence in chemometrics and can emerge for different reasons. In certain types of analyses, some values are intentionally not measured (e.g., in *Selected Ion Monitoring Mass Spectrometry* – SIM or because of irregular sampling schemes when monitoring a chemical process²¹¹). Alternatively, missing values may be purely incidental (e.g., because of some sensor failure²¹¹ or saturation^{319,320}). In other cases, certain elements of a multi-way array are set to missing because the actual values would lead to a violation of low rank N -linearity (cf. sections 2.4.1 and 2.4.3)^{8,248,278,PII}.

Although they do not directly determine a change in the rank of the array, missing values can exacerbate the problems related to some other deviations^{6,60,248,297,PII}.

In recent years, the effect of missing values on the quality of the final model and on the rate of convergence for different fitting algorithms has been the subject of substantial investigation and has been shown to be potentially quite dramatic for systematic patterns of missing elements^{15,41,60,248,297,PII}.

Two different algorithmic approaches to the problem of missing values are commonly used: weighted regression (cf. section 2.5) and single imputation^{31,PII}. In the former, the model is fitted only on the elements that are not missing; in the latter, new estimates for the missing values are obtained at each iteration using the values in the interim model^{PII}. Under some assumptions on the distribution of the residuals, the second method can also be referred to as ‘Expectation Maximisation’ and the imputed values represent an additional set of parameters that is estimated conditionally on the loading matrices^{31,71,202,PII}.

Even though upon convergence the two methods should yield identical solutions¹⁵⁶, the two approaches are not equivalent and there are some indications that the weighted least squares approach is more effective for large amounts of missing values and specific, problematic patterns of missing values^{PII}. Regardless of the method, the more elements are

missing, the slower the convergence. In this respect, exploiting certain properties of the Khatri-Rao product may prove beneficial^{PII,PIII}. Constraints are also helpful to improve convergence, whether applied directly to the loading matrices (e.g., non-negativity and unimodality) or indirectly, by substituting the missing value with some expected value (e.g., zeros in EEM fluorescence landscapes)^{6,15,297}.

The pattern of missing values has a much more visible effect than for example their fraction and it is remarkable that PARAFAC models can still be reliably estimated with as many as 70% of the array elements set as missing^{PII}. This is at variance with the two-way case, in which PCA becomes rather unstable already at 25-40% missing values^{107,319}. One of the reasons for this stability is clearly the fact that the number of data points for the fitting largely exceeds the number of model parameters. This ratio further increases with the order of the array and, for higher orders, one may be able to use several million data points to estimate a few hundreds model parameters^{PIII}. A second reason is likely to be the rigid model structure of PARAFAC and the requirement that the variation be multilinear.

This stability is not shared by all patterns of missing values and randomly missing values (RMV) are less problematic than the systematically missing spectra (SMS) pattern found in fluorescence spectroscopy^{PII}. In this respect, however, one could envision an algorithm based on the weighted least squares approach that deliberately ignores a large fraction of values (that is, treats them as missing) in a perfectly random pattern, thereby reducing the computational load per iteration. Such an algorithm could for example be used to provide initial estimates to fit the PARAFAC model on the whole array, although some care needs to be paid to avoid that the pattern of missing values could interfere with multilinearity^{PII}.

2.5 Fitting criteria

Fitting a model to a set of data typically implies certain assumptions on the distribution of the measurement errors (also referred to as noise)^{235,310}.

In general, noise is assumed normally distributed with zero mean and is identified with the fitted model's residuals²³⁵. Depending on variance and covariance of noise in the array's elements, Least Squares (LS), Weighted Least Squares (WLS) and Maximum Likelihood (ML) fitting criteria naturally emerge^{41,235,310}. The corresponding optimisation problems can be written respectively as:

$$\underset{\mathbf{A}_1 \dots \mathbf{A}_N}{\operatorname{argmin}} \left[\mathbf{r}(\mathbf{A}_1 \dots \mathbf{A}_N)^\top \mathbf{r}(\mathbf{A}_1 \dots \mathbf{A}_N) \right], \quad (2.10a)$$

$$\underset{\mathbf{A}_1 \dots \mathbf{A}_N}{\operatorname{argmin}} \left[\mathbf{r}(\mathbf{A}_1 \dots \mathbf{A}_N)^\top \mathbf{W} \mathbf{r}(\mathbf{A}_1 \dots \mathbf{A}_N) \right], \quad (2.10b)$$

and

$$\underset{\mathbf{A}_1 \dots \mathbf{A}_N}{\operatorname{argmin}} \left[\mathbf{r}(\mathbf{A}_1 \dots \mathbf{A}_N)^\top \mathbf{S}^{-1} \mathbf{r}(\mathbf{A}_1 \dots \mathbf{A}_N) \right], \quad (2.10c)$$

where $\mathbf{r}(\mathbf{A}_1 \dots \mathbf{A}_N) \equiv \operatorname{vec} \mathbf{X}_1 - (\mathbf{A}_N \odot \dots \odot \mathbf{A}_1) \mathbf{1}_F$ denotes the vectorised model's residuals, \mathbf{W} is a diagonal matrix of non-negative weights (e.g., the inverse of the variance of the single elements of \mathbf{r}) and \mathbf{S}^{-1} denotes the inverse of a generic variance/covariance matrix of the measurement errors^e.

Obviously, the three cases are equivalent when $\mathbf{S}^{-1} = \mathbf{W} = \sigma^{-2} \mathbf{I}$, where \mathbf{I} denotes an identity matrix of suitable size and σ^2 is the noise variance. Thus, upon convergence to the global minima and for the right choice of model rank, LS fitting yields the maximum likelihood estimates of the model parameters under the condition that the errors/residuals are normally distributed, uncorrelated and have identical variance (i.e., that noise is homoscedastic). Likewise, WLS yields estimates that are optimal in a maximum likelihood sense under the condition that $\mathbf{W} = \mathbf{S}^{-1}$, viz. that \mathbf{S} is diagonal and the weights are the inverses of the variance of the corresponding elements of \mathbf{r} . In this case, the criterion is often referred

^e Note that ML fitting also entails the estimation of the variance/covariance matrix \mathbf{S} ¹⁹⁵. Thus, the appellation of ML fitting for equation (2.10c) is correct only if one assumes that the noise variance-covariance matrix be known in advance. This is sometimes possible because the distribution of noise is known for a specific analytical technique (e.g., single photon counting fluorescence²²⁰). Alternatively estimates for \mathbf{S} can be obtained from replicate measurements³¹⁰.

to as χ^2 -fitting²³⁵. WLS fitting allows departures from the assumption of identical variance for all data elements and should be preferred whenever reliable estimates of \mathbf{S} are available or significant deviations from homoscedasticity are expected³¹⁰. Finally, the ML criterion with non-diagonal \mathbf{S} is the most appropriate when the measurement errors are correlated^{41,310}.

These considerations notwithstanding, LS solutions are often reasonably good for both qualitative and quantitative purposes even when there are deviations from the ideal conditions^{31,PI}, and the difference between the least squares solution and the actual maximum likelihood estimates is expected to be small if the signal to noise ratio is sufficiently high^{41,PI}. Furthermore, reliable estimates of \mathbf{S} are seldom available as they require a number of replicates that is often not feasible^{41,310}. Therefore, least squares algorithms have generally been favoured when fitting PARAFAC models. Besides, WLS and ML algorithms are often slower than their least squares counterparts because of their increased complexity^{41,129,310,PII,PIII}. In this sense, it is even more important to exploit as much as possible the structure of the PARAFAC model and of the noise variance-covariance matrix^{93,310,PIII}. For example, if \mathbf{S} is diagonal, the Jacobian matrix \mathbf{J} need not be explicitly computed in order to calculate $\mathbf{J}^T \mathbf{S}^{-1} \mathbf{J}$ (or the partitions used for, e.g., ALS algorithms³¹⁰) and matrix-matrix products of full matrices can be used instead^{PIII}.

Great savings can also be achieved when \mathbf{S} can be expressed as a Kronecker product, regardless of its being diagonal. For instance, given N (positive definite) variance/covariance matrices \mathbf{S}_n of size $I_n \times I_n$ for $n = 1 \dots N$ and if $\mathbf{S} \equiv \mathbf{S}_N \otimes \dots \otimes \mathbf{S}_1$, the general ML loss function $L(\mathbf{A}_1, \dots, \mathbf{A}_N)$ can be written as:

$$\begin{aligned} L(\mathbf{A}_1 \dots \mathbf{A}_N) &= \mathbf{r}(\mathbf{A}_1 \dots \mathbf{A}_N)^T (\mathbf{S}_N \otimes \dots \otimes \mathbf{S}_1)^{-1} \mathbf{r}(\mathbf{A}_1 \dots \mathbf{A}_N) = \\ &= \left\| \left(\tilde{\mathbf{S}}_N^{-1} \otimes \dots \otimes \tilde{\mathbf{S}}_1^{-1} \right) \left[\mathbf{x} - (\mathbf{A}_N \odot \dots \odot \mathbf{A}_1) \mathbf{1}_F \right] \right\|_2^2 = \\ &= \left\| \left(\tilde{\mathbf{S}}_N^{-1} \otimes \dots \otimes \tilde{\mathbf{S}}_1^{-1} \right) \mathbf{x} - \left(\tilde{\mathbf{S}}_N^{-1} \mathbf{A}_N \odot \dots \odot \tilde{\mathbf{S}}_1^{-1} \mathbf{A}_1 \right) \mathbf{1}_F \right\|_2^2, \end{aligned} \quad (2.11)$$

where $\tilde{\mathbf{S}}_n$ denotes the Cholesky factor²⁶⁷ (or, alternatively, the square root⁴¹) of \mathbf{S}_n and the last equality holds for a well known property of the Khatri-Rao product²⁷⁸. Thus, in this case, the ML estimates of the loading matrices can be found by fitting in an LS sense a

PARAFAC model to an appropriately pre-processed array^{f42,93,120,197}. Unfortunately, pre-processing as a means to yield ML estimates cannot be employed when the noise is correlated across several modes (whereby the word ‘across’ is used in the same way as in Smilde *et al*²⁷⁸)^{42,120,310}. In the latter case, only one simplification seems possible; that is, when the noise is independent across one or more modes, the variance/covariance matrix \mathbf{S} becomes block diagonal for some permutation of the modes of the array, with some advantages in terms of storage and when computing its inverse^{23,195,310,PIII}.

It is important to mention that \mathbf{S} and \mathbf{W} need not be specified on a statistical base and that the only condition for problems (2.10b) and (2.10c) to have a solution is that both matrices be positive semi-definite. Hence, they can also be derived from *a priori* knowledge and employed to downscale the effect of certain elements of the data array on the final model and in this sense they have been used extensively^{8,41,62,120,140,248,PV}. For example, weights can be used to prevent the fitted model from being heavily influenced by features that may deviate from low-rank N -linearity (cf. section 2.4.3)^{6,31,41,141,248,PII}. A WLS criterion is also used to deal with missing values (cf. section 2.4.6)^{31,107,211,319,320,PII}.

In recent years, a number of methods have been introduced that do not fit PARAFAC models according to any well established criterion and that are loosely based on the concepts underlying PARAFAC-ALS^{50,56-58,82,138,139,181,PI}.

The mathematical relation between the loss functions minimised by these algorithms, if any, and the least squares one is not always clear and does not necessarily lead to improved efficiency or robustness^{82,PI}. All these methods do not minimise only one loss function (i.e., a more or less penalised distance between a model, however related to the PARAFAC one, and the data), but solve a different minimisation problem depending on which loading matrix is being estimated at a specific sub-step. Thus, they lack any well-characterised convergence property in terms of PARAFAC least squares loss function, which is not bound to decrease (or, more precisely, not to increase) as it is for ALS^{56,57,82}.

^f viz. $\text{vec} \tilde{\mathbf{X}}_1 \equiv (\tilde{\mathbf{S}}_N^{-1} \otimes \dots \otimes \tilde{\mathbf{S}}_1^{-1}) \text{vec} \mathbf{X}_1$. Note also that $\tilde{\mathbf{S}}_N^{-1} \otimes \dots \otimes \tilde{\mathbf{S}}_1^{-1}$ need not be formed to calculate $\tilde{\mathbf{X}}$ thanks to the associativity of the Kronecker product and to its well known property: $(\mathbf{C} \otimes \mathbf{A}) \text{vec} \mathbf{B} = \text{vec} \mathbf{ABC}^T$ ¹⁹⁵.

Additionally, most of these methods have more restrictive requirements for identifiability than the theoretical ones; consequently, they are most often not applicable when the loading matrices are rank deficient^{165,271}.

On the other hand, at least one of these algorithms (namely SWATLD^{57,82,PI}) does yield models that are reasonably close to the least squares ones and could be used when high precision and statistical properties of the solution are not main requirements for the decomposition (for example, to provide initial estimates for a least squares algorithm)^{82,PI}. SWATLD appears to be a rather stable and fast algorithm (cf. section 2.7); however, the unorthodoxy of its fitting criterion makes it very difficult to ascertain what renders SWATLD so stable in many respects and to understand whether some of its advantageous features could be implemented in other, theoretically sounder, algorithms^{82,PI}.

2.6 Difficulty of the fitting problem

The performances of the different algorithms for fitting the PARAFAC model are influenced by some well known factors (e.g., collinearity between the underlying components and the signal to noise ratio^{129,155,PI,PII}) as well as some peculiar ones which are still not completely understood theoretically (e.g., factor degeneracies and swamps^{168,203,226,347}). Therefore, in order to correctly assess the behaviour of the different methods, it is very important that the data sets on which they are tested be as well-characterised as possible.

In general, time and effort necessary to generate appropriate real data sets favoured tests for PARAFAC fitting algorithms chiefly based on artificial data^{81,82,129,155,PI-PIII}. Simulated arrays are typically designed according to the features that are expected to yield an effect on both quality of the results and efficiency of the algorithms. Furthermore, they allow a more refined control over the varied factors than real sets and do not contain unwanted disturbances that may obscure the effects in which one is interested^{39,82,155,226,295,PI,PII}. Thus, methods to generate synthetic data sets for PARAFAC algorithms abound in the chemometrics literature^{39,80,82,129,155,159,PI,PII}.

In general, all methods boil down to creating loading matrices according to the requirements of the simulations, computing an N -way array based on the PARAFAC model and such loading matrices, and summing it with an array of random errors of appropriate distribution and magnitude^{Pf}. The loading matrices can be constructed to be rank deficient, or such that the PARAFAC model is not unique or that a PARAFAC decomposition of given rank would yield factor degeneracies^{168,226}. In most cases, this is accomplished by using simple linear transformations applied to column-wise orthogonal matrices. Hence, said \mathbf{U} an $I_n \times F$ column-wise orthogonal matrix, the loading matrix \mathbf{A}_n is calculated as $\mathbf{A}_n = \mathbf{U}\mathbf{R}$, where \mathbf{R} is a matrix of suitable size^{159,226,267,295,Pf}. For example, \mathbf{R} can be chosen as the Cholesky factor or as the square root of the desired cross-product for \mathbf{A}_n (e.g., having identical off-diagonal elements)^{159,248,267Pf,PfII}. Added noise, if any, is often normally distributed with zero mean. The single values of the array representing the noise may be multiplied by a scalar to yield a specific variance, or signal to noise ratio or contribution to the total sum of squares of the final array^{190,Pf,PfII}. Further manipulations are necessary when studying the effect of heteroscedastic or correlated noise on a least squares solution^{310,Pf}, the efficiency of an algorithm using a weighted loss function^{41,310} or the effect of model error on the parameters' estimates^{39,184}. In this sense, there are some indications that, in order for the simulated sets to be realistic, some small deviations from low-rank multilinearity should be allowed (e.g., giving a Tucker structure to the noise)³⁹.

Great attention should be paid to the procedure used to generate the data set. For instance, choosing the loading matrices solely on the basis of their resemblance with those commonly encountered when fitting PARAFAC models (e.g., excitation and emission spectra or chromatograms) may be advantageous when interpreting some results from the chemistry point of view and visually expedient, but is not sufficient to properly characterise the data set from a numerical point of view^{129,PfII}. Likewise, smooth loading vectors make the results more pleasant to the eye and may help detecting artefacts in the solution^{129,PfII}, but, *ceteris paribus*, they bear neither consequence nor advantage on the fitting procedure unless smoothing constraints are being tested³⁷.

Regardless of the method used to generate a data set, some indications about the conditioning of the problem and the magnitude and distribution of noise are necessary to

properly characterise the problem from a numerical point of view. Several diagnostic parameters have been used in the chemometric literature to measure the difficulty of the fitting problems, but many seem to fall short of giving consistent and concise information. Over the years, the condition number of the loading matrices or of their Khatri-Rao products^{82,155}, the congruence between factors (i.e., the cosine of the angle between the vectorised factors) or loading vectors^{82,PI,PII}, the selectivity⁸², the singular values for the loading matrices^{129,PI,PII}, a diagnostic referred to as the 'array's condition number'¹²⁹ and the Jacobian's condition number and singular values^{225,226,PI,PII} have been used to characterise the problem.

Not all of these are equally informative and some can provide conflicting evidences^{PII}. For instance, while the condition number of the loading matrices affects convergence rate and quality of the solution^{129,82,155}, it may vary considerably between the different modes and relatively high values for one or more loading matrices do not necessarily imply that the fitting problem is difficult to solve^{129,PI,PII}. In this sense, the conditioning of the Khatri-Rao products of all loading matrices but one (i.e., that of $\mathbf{Z}_{-n} \equiv \mathbf{A}_N \odot \dots \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \dots \mathbf{A}_1$) seems much more relevant for the numerical stability of most algorithms. This is easy to see for the ALS algorithm, which entails the computation of the pseudo-inverse of \mathbf{Z}_{-n} rather than that of the single loading matrices.

However, diagnostics that refer to fewer loading matrices than the order of the array may miss significant relations implying or excluding uniqueness, and thus ill-conditioning of the problem^{271,295}. Hence, if any single number should be used to determine the conditioning, the most natural choice among those listed seems the Jacobian's condition number, which is related to the convergence rate of several methods^{23,225,PI,PII}. Additionally, the Jacobian's condition number can directly account for the effect of the pattern of missing values^{PII}. This diagnostic parameter seems also preferable to the 'array's condition number'¹²⁹, which relies on the estimation of two separate models of different rank, one of which is necessarily wrongly specified.

The main drawback of the Jacobian's condition number is that it is not affected by amount and nature of noise in the array, which also influences the rate of convergence of some algorithms^{23,102,193}. In order to take this into account, one could use the Hessian

matrix instead^{23,226}. It is noteworthy then, that, since $\mathbf{J}^T\mathbf{J}$ approximates the Hessian matrix for least squares problems and the difference between these two matrices depends on the residuals, using the condition number of \mathbf{J} is reasonable for problems characterised by small residuals^{PI,PII}.

From the computational point of view, it is worth mentioning that the sparsity of \mathbf{J} and of the second-order contribution to the Hessian $(\mathbf{Q})^{PIII}$ and the availability of efficient routines to compute the products of \mathbf{J} (and \mathbf{J}^T) with a vector^{PIII} suggest the possibility of using iterative methods to calculate the extremal eigenvalues of the Hessian (or singular values of \mathbf{J}), overcoming storage limitations¹⁰⁵.

Another useful diagnostic for noise that is more specific to PARAFAC is the core consistency at the solution, which is again a measure of the appropriateness of a PARAFAC model of given rank³⁹. This becomes particularly important for experimental data sets, which almost necessarily depart from the basic assumptions for employing PARAFAC. Whether or not these deviations from low-rank N -linearity represent a problem is determined by their magnitude and nature (cf. section 2.4) and sometimes they are compatible with the computation of reasonable estimates of model's parameters or of some related values^{31,39,120,184}. Nonetheless, their presence often entails a worsening of the solution and slower convergence because the model's rank is almost inherently different from the rank for the systematic part of the variation^{168,184,226,PI}.

2.7 Algorithms for fitting PARAFAC models

The need for fitting PARAFAC models on data sets of increasing size and complexity and the use of resampling techniques (e.g., jack-knifing^{198,249} and bootstrapping^{75,113,278}) require efficient algorithms.

In spite of several well known shortcomings^{83-85,155,203,PI,PII}, most of the work pertaining to the computational aspects of fitting PARAFAC models has been restricted to two classes of algorithms: ALS^{51,114,278}, which is iterative, and direct procedures based on solving a

generalised eigenvalue problem [e.g., GRAM²⁶⁴ or the *Direct TriLinear Decomposition* (DTLD)²⁶³].

Over the years, several other methods have been tested, but only few of them appeared to be able to perform as well as PARAFAC-ALS^{82,PI}. For example, a steepest descent procedure^{102,193} was used for preliminary studies on the PARAFAC model, but was soon abandoned in favour of ALS as it was found unfeasible when the model's rank was relatively high and for larger arrays¹¹⁴. More recently, a modified version of the conjugate directions method²³⁵ based on fast computation of gradient and exact line search has been tested to refine an initial solution obtained with ALS and to attain a quadratic convergence in the final stages⁹³. It was concluded that the method is rather inefficient and that its convergence rate is long from being quadratic⁹³.

Consequently, most of the newly proposed methods for fitting PARAFAC models comprise more or less complex additions to the ALS basic procedure aiming at accelerating its convergence rate or loosely rely on an underlying ALS scheme whereby the fitting criterion is modified in an attempt to cope with some problems encountered with the least squares one (cf. section 2.5).

Per contra, many efficient methods for solving general optimisation problems^{23,102,193} have only seldom been employed in the PARAFAC context^{10,22,93,122,185,190,224,225,322,PI,PII} and the development of fitting procedures that are not based on the ALS idea has been rather limited. The fact that most of the implementations of such alternative procedures lack the generality and the simplicity of ALS based ones is likely one of the reasons why these methods received limited attention^{31,155,278,PI}.

In this section, a brief and as objective as possible account of the current state of the art on PARAFAC fitting procedures is rendered, and some alternatives are pointed out that would reap great benefit from the peculiar structure of PARAFAC and from the properties of the Khatri-Rao product^{PIII}.

The idea is not to exhaust the subject of which method is the absolute best, nor is it to compare all the available algorithms. There are simply too many of them (especially if one considers those made available from the numerical optimisation field) and, more im-

portantly, the matter has likely no ultimate (or trivial) answer¹⁰². For example, if one considers only the problem's size (i.e., the number of estimated parameters), some algorithms are more suited for small problems (e.g., Levenberg-Marquardt and Newton methods^{193,224,PI}) while others are expected to be faster for large scale ones (e.g., PCG, truncated-Newton and limited memory quasi-Newton methods^{102,193,225,322}).

Besides, the performances of each method are influenced by several aspects pertaining to the data array rather than the algorithm in itself^{129,PI}. Hence, factors like deviations from low-rank N -linearity^{39,PI} (cf. section 2.4), collinearity between the underlying components and model's identifiability [i.e., more simply Jacobian's rank deficiency (cf. section 2.2.1)]^{23,129,155,225}, adequate pretreatment^{49,117,120,347,PVI}, amount and type of noise^{102,193,PI-PIII} and presence and pattern of missing values^{129,278,PII} play an important role in determining which algorithm should be used^{PI,PII}. In this respect, it seems also advisable to properly characterise the data sets that are used for testing a new algorithm through some sensible diagnostics^{129,PI,PII} (cf. section 2.6).

The matter of choosing the most suitable algorithm is further complicated if one considers that other criteria than the least squares one may be more suited for specific types of data^{41,310} (cf. section 2.5) and that constraining a PARAFAC model is sometimes necessary or at least advantageous^{6,31,36,110,163,192,224,240,296}. Implementing such modifications or combining them in a single procedure may be overtly difficult or numerically unstable depending on the approach³¹. Hence, algorithms that work finely in the simple least squares case may be unsuitable *mutatis mutandis* for constrained problems or to minimise non-least squares loss functions (e.g., combining compression and non-negativity constraints is still an unresolved problem at this stage)³¹.

2.7.1 Alternating Least Squares

ALS is a relatively old method, known in the field of mathematical programming as nonlinear Gauss-Seidel algorithm^{23,70,346}. In this approach, the nonlinear problem represented by fitting a PARAFAC model is split in smaller linear least squares problems that are solved iteratively. At each step, a subset of the model parameters is estimated condi-

tionally on the remaining ones and a single iteration is completed when all the sub-problems have been solved. Hence, each complete iteration is comprised of a number of sub-steps that depends on the partitioning of the model parameters. Most often, the unknowns in the linear problems are the loading matrices and the number of sub-steps equals the order of the array, but, for specific constraints, also single loading vectors may serve this purpose and the number of sub-steps may increase^{31,52}.

There are numerous methods to solve linear (weighted) least squares problems²³, but the properties of the Khatri-Rao and opportune matricisations can help to reduce the computational expense for the single iteration of PARAFAC-ALS^{278,PI-PIII}.

For example, the estimate for the n -th loading matrix at the $(s+1)$ -th iteration in the least squares case is computed as:

$$\mathbf{A}_n^{(s+1)} = \mathbf{X}_n \left(\mathbf{A}_N^{(s)} \odot \dots \odot \mathbf{A}_{n+1}^{(s)} \odot \mathbf{A}_{n-1}^{(s+1)} \dots \odot \mathbf{A}_1^{(s+1)} \right)^{+\top} \equiv \mathbf{X}_n \left(\mathbf{Z}_{-n}^{(s)} \right)^{+\top} \quad (2.12)$$

where $+$ denotes the Moore-Penrose inverse²⁶⁷. One can easily verify that the most expensive steps of this algorithm are the computation of \mathbf{Z}_{-n}^+ (where the iteration number is dropped for clarity) and the products $\mathbf{X}_n \mathbf{Z}_{-n}^{+\top}$; accelerating these steps may significantly improve the speed of the algorithm. Therefore, \mathbf{Z}_{-n}^+ is often calculated as^{31,51,114,PII}:

$$\mathbf{Z}_{-n}^+ = \mathbf{Z}_{-n} \left(\mathbf{A}_1^\top \mathbf{A}_1 * \dots * \mathbf{A}_{n-1}^\top \mathbf{A}_{n-1} * \mathbf{A}_{n+1}^\top \mathbf{A}_{n+1} \dots * \mathbf{A}_N^\top \mathbf{A}_N \right)^+, \quad (2.13)$$

where $*$ denotes the element-wise (Hadamard) product and the iteration number is dropped for clarity.

Notwithstanding the reduction in the computational time, reckoning \mathbf{Z}_{-n}^+ in this fashion can induce some numerical instability because of the squaring of the condition number²³. Thus, unless some type of regularisation is applied, such approach should be avoided when the \mathbf{Z}_{-n} 's are ill-conditioned (e.g., because of factor degeneracies)^{23,31,245}. For the same reasons, one should not substitute the pseudo-inverse in the rightmost term of (2.13) with an ordinary inverse, something that is possible if \mathbf{Z}_{-n} has full column rank. Besides, this substitution would also bring a limited advantage because the model's rank

Table 2.1. Scheme of a naïve and an optimised formulation of the PARAFAC-ALS algorithm for a 3-way array $\underline{\mathbf{X}}$ of size $I \times J \times K^{PIII}$. s denotes the iteration number.

Without optimisation	With optimization
0) Set $s = 0$ and initialise \mathbf{B}_0 and \mathbf{C}_0	0) Set $s = 0$ and initialise \mathbf{B}_0 and \mathbf{C}_0
1) $\mathbf{A}_{s+1} = \mathbf{X}^{(I \times JK)} (\mathbf{C}_s \odot \mathbf{B}_s)^{+\top}$	1) $\mathbf{A}_{s+1} = \mathbf{X}^{(I \times JK)} (\mathbf{C}_s \odot \mathbf{B}_s)^{+\top}$
2) $\mathbf{B}_{s+1} = \mathbf{X}^{(J \times IK)} (\mathbf{C}_s \odot \mathbf{A}_{s+1})^{+\top}$	2) $\mathbf{A}_{s+1} = \left[\mathbf{X}^{(I \times JK)} (\mathbf{C}_s \odot \mathbf{B}_s) \right] (\mathbf{C}_s^\top \mathbf{C}_s * \mathbf{B}_s^\top \mathbf{B}_s)^{+}$
3) $\mathbf{C}_{s+1} = \mathbf{X}^{(K \times IJ)} (\mathbf{B}_{s+1} \odot \mathbf{A}_{s+1})^{+\top}$	3) $\tilde{\mathbf{X}}^{(F \times JK)} = \mathbf{A}_{s+1}^\top \mathbf{X}^{(I \times JK)}$
4) Check convergence	4) $\mathbf{B}_{s+1} = \left[\tilde{\mathbf{X}}^{(J \times KF)} (\mathbf{I}_F \odot \mathbf{C}_s) \right] (\mathbf{C}_s^\top \mathbf{C}_s * \mathbf{A}_{s+1}^\top \mathbf{A}_{s+1})^{+}$
5) If not converged repeat from 1)	5) $\mathbf{C}_{s+1} = \left[\tilde{\mathbf{X}}^{(K \times FJ)} (\mathbf{B}_{s+1} \odot \mathbf{I}_F) \right] (\mathbf{B}_{s+1}^\top \mathbf{B}_{s+1} * \mathbf{A}_{s+1}^\top \mathbf{A}_{s+1})^{+}$
	6) Check convergence
	7) If not converged repeat from 2)

is usually rather small (in the order of a few units) and gain is almost negligible compared to the cost of computing $\mathbf{X}_n \mathbf{Z}_{-n}$.

Each $\mathbf{X}_n \mathbf{Z}_{-n}$ product requires in theory $2F \prod_n I_n$ operations and a permutation of the array's modes. The permutation can be accelerated through appropriate matricisations^{PIII} or avoided altogether by expressing the algorithm in terms of frontal slabs of $\underline{\mathbf{X}}$ ²⁷⁸, but while computationally more efficient, both these procedures do not reduce the number of operations necessary to complete a single iteration. Per contra, computing $\mathbf{X}_n \mathbf{Z}_{-n}$ for $n = 2 \dots N$ as

$$\mathbf{X}_n \mathbf{Z}_{-n} = \tilde{\mathbf{X}}_n (\mathbf{I}_F \odot \tilde{\mathbf{Z}}_{-n}), \quad (2.14)$$

where $\tilde{\mathbf{X}}^{(F \times I_2 \dots I_N)} \equiv \mathbf{A}_1^\top \mathbf{X}_1$, $\tilde{\mathbf{X}}_n \equiv \tilde{\mathbf{X}}^{(I_n \times I_{-\{1,n\}}^F)}$ and $\tilde{\mathbf{Z}}_{-n} \equiv \mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \dots \odot \mathbf{A}_2$, allows a reduction that is proportional to the ratio between the largest mode (assumed here and without lack of generality to be the first) and the rank of the model. In fact, if one ignores the operations necessary to compute $(\mathbf{Z}_{-n}^\top \mathbf{Z}_{-n})^{+}$, the cost of one ALS iteration using (2.14) is approximately equal to the cost of computing $\mathbf{J}^\top \mathbf{v}$ using the chain rule^{PIII}. Again, appropriate matricisations or using frontal slabs may increase the efficiency. An immediate pos-

sibility to further accelerate the calculation of $\mathbf{X}_n \mathbf{Z}_{-n}$ is to operate column-wise. Namely, the f -th column of $\mathbf{X}_n \mathbf{Z}_{-n}$ is equal to $\tilde{\mathbf{X}}_{n,f} \tilde{\mathbf{z}}_{-n,f}$, where $\tilde{\mathbf{X}}_{n,f}$ is the matricised $(N-1)$ -array obtained by fixing to f the index for the first mode of $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{z}}_{-n,f}$ denotes the f -th column of $\tilde{\mathbf{Z}}_{-n}$. Table 2.1 reports the PARAFAC-ALS algorithm for a three-way array with and without the optimisations just illustrated.

ALS becomes less efficient than Hessian based methods (cf. section 2.7.4) as the order of the array increases^{PIII}. At the same time, its computational complexity grows linearly with the number of factors whereas for Gauss-Newton (or Newton) methods based on Cholesky decomposition it depends on the third power of the rank of the model and for quasi-Newton algorithms on its square^{94,102,193}. Therefore, ALS is likely to be faster for relatively high ranks^{PI-PIII}.

The convergence rate of PARAFAC-ALS is at most linear and in practice, this algorithm can be extremely slow, especially in case of high collinearity between the factors^{129,155,PI}, when factors degeneracies emerge^{203,226,245} or for large fractions of missing elements^{PII}. Several modifications have been proposed to accelerate convergence³⁷; amongst them: line search extrapolation/relaxation^{10,31,114,253,PI,PIII}, regularisation^{31,224,245,PI} and compression^{33,155,PI}.

Line search extrapolation and relaxation procedures work by extending the optimal steps by a certain factor α . α can be a fixed value, or can be calculated on the basis of the similarity between the last two updates or, more simply, can be based on the number of iterations^{10,31,114,235,253,PI}. The effect this procedure has on the number of iterations necessary to reach convergence can be quite dramatic and, although it adds to the complexity of the algorithm, the reduction in terms of time can be remarkable even for quite complicated extrapolation schemes^{129,226,251}. While in most PARAFAC-ALS implementations the choice of α is based on heuristics, a method that solves analytically the line search problem was proposed already in 1992⁹³ and it was recently implemented in MATLAB^{PIII}. Preliminary tests on three-way arrays indicate that the exact line search method yields a large reduction in the number of ALS iterations. Nonetheless, an ALS iteration using the procedure described in Paper III is approximately twice as costly as one of a standard ALS (for the three-way case) where α depends on the number of iterations³⁷. Thus, the exact line

search algorithm appears to be more time consuming than procedures based on simpler schemes. Clearly, these results have to be validated with a more extensive study, but they also suggest that it might be worthwhile to investigate on line search procedures that approximate the exact solution at lower cost.

ALS has been reported to be more sensitive than other algorithms (apart from direct methods) to over-factoring^{Pf}. In particular, when the rank of the model exceeds the rank of the systematic variation, the number of 2FD's recorded in the solution increases^{Pf}. This is consistent with theoretical considerations on the relation between rank of the model and of the array and emergence of factor degeneracies²²⁶ and suggests that modifications to the basic ALS algorithm that are intended to reduce the occurrence of 2FD's (e.g., regularisation) may be helpful also in improving the resistance to over-factoring. Regularisation is a common strategy when solving least squares problems²³ and can reduce the number of iterations spent in a swamp and accelerate convergence^{129,131,245}. However, it may also introduce some bias and yield solutions that visibly differ from the least squares one, thus some care should be paid in evaluating the results^{129,245}.

Compression (cf. section 2.7.6) allows a great reduction in the size of the problem and seems to improve the resistance of ALS (and of other methods) to mild over-factoring. However, more ALS iterations are required on the compressed array and the reduced size of the problem does not always imply a reduction of computational time^{Pf}.

In general, ALS seems to work quite well for three-way arrays so long as the rank of the model is known and the underlying factors are not too collinear. For relatively difficult problems, several restarts or better initial estimates (e.g., like those obtained from compression – cf. section 2.7.6) may be necessary to obtain fast convergence or convergence to a meaningful solution²⁰³.

2.7.2 Direct methods

The only alternative to PARAFAC-ALS that has received much attention is the GRAM-DTLD algorithm^{263,264}, which is based on a generalised eigenvalue problem and has fixed computational complexity⁸³⁻⁸⁵. Unfortunately, what is gained in speed is lost in accuracy,

and while in the noise-free case the model underlying GRAM-DTLD is identical to the PARAFAC model, this is not the case for real experimental data. In fact, the optimisation criterion of the GRAM-DTLD is not strictly well defined in terms of the PARAFAC loss function⁸², and the two solutions may differ considerably, especially when the signal to noise ratio is low. The GRAM-DTLD algorithm per se has been repeatedly proven to be inferior to PARAFAC-ALS in terms of quality of the solution^{82,179,204,PI}. Nonetheless, because of the fixed complexity and its speed, it has been often advocated as to provide initial estimates of the loading matrices for PARAFAC-ALS^{31,179,204}. In particular, since most iterative nonlinear least squares algorithms are sensitive to the initial estimates of the parameters and convergence is ensured only if the initial guess is reasonably close to the final solution^{94,102,193}, a rational start obtained, e.g., via GRAM-DTLD may lead to faster convergence compared to purely random values. However, since GRAM-DTLD gives the same solution every time, using it as the sole option does not give any information on whether the main algorithm has converged to a local minima³¹ and does not help in case the corresponding values lead to a degenerate solution or a swamp^{155,203,226}. Other algorithms can be used to provide further rational starts, e.g., curve resolution methods^{68,290,291}, NIPALS or SVD³¹, some alternative GRAM-DTLD like procedure¹⁷⁹ or iterative non-least squares algorithms (e.g., SWATLD⁵⁷ – cf. section 2.7.3)^{82,PI}.

Another direct method based on singular value decomposition (hereafter named PARAFAC-SVD) can be used for calibration purposes for three-way arrays. In PARAFAC-SVD, the loading vectors in the sample mode are assumed known and equal (apart from the different scaling) to the matrix (\mathbf{U}_1) holding the concentrations for the calibrations samples^{184,185}. In essence, PARAFAC-SVD approximates the loadings for each factor with the first left and right singular vectors of F suitably calculated matrices (viz. $\mathbf{V}_f = (\mathbf{d}_f^T \otimes \mathbf{I}_{l_2}) \tilde{\mathbf{X}}^{(l_2 F \times l_3)}$, where \mathbf{d}_f denotes the f -th column of $(\mathbf{U}_1^T \mathbf{U}_1)^{-1}$ and $\tilde{\mathbf{X}}^{(F \times l_2 l_3)} \equiv \mathbf{U}_1^T \mathbf{X}_1^g$)¹⁸⁵.

^g The notation has been changed to be consistent with the one employed herein.

Like DTLD-GRAM this method does not perform as well as a PARAFAC-ALS^{184,185}, but could provide good initial values for iterative algorithms. An interesting aspect of PARAFAC-SVD though is that it can be straightforwardly extended to higher orders, although in this case it becomes iterative. It is easily verified that for N -way $\underline{\mathbf{X}}$, \mathbf{V}_f is a matrixed $(N-1)$ -way array (i.e., $\mathbf{V}_f^{(I_2 \times I_3 \dots I_N)} = (\mathbf{d}_f^T \otimes \mathbf{I}_{I_2}) \tilde{\mathbf{X}}^{(I_2 \times I_3 \dots I_N)}$) and its best rank one approximation is a rank one PARAFAC model. Currently, neither references nor record exist on the application of this method for arrays of order higher than three.

2.7.3 Iterative Non-least squares methods

A relatively large number of new iterative methods for fitting PARAFAC has been proposed in recent years^{50,56,57,138,139,181,342,343} that do not solve (2.10a), but “related” problems containing various kinds of adjustments. In recent reviews, most of these algorithms have been found wanting, both theoretically and in practice, in terms of quality of the solution, reliability and computational expense^{82,Pf} (cf. section 2.5).

Of all these methods, only *Alternating Slice-wise Diagonalization* (ASD) and *SWATLD* (*Self-Weighted Alternating Tri-Linear Decomposition*) yield solutions of acceptable quality^{82,Pf}. However, ASD has turned out to be rather unsatisfactory with respect to most of the design factors taken into account in some Monte Carlo simulations (namely noise, degree of collinearity and mild over-factoring). Moreover, the original claim that ASD would avoid factor degeneracies¹³⁹ has been shown to be not true^{Pf}, and this method yields a much higher number of 2FD’s in the solution than several other methods (including ALS)^{Pf}.

SWATLD has been shown to be fast, relatively resistant to mild collinearity and surprisingly stable with respect to over-factoring^{Pf}. Moreover, it did not yield solutions with 2FD’s in test conducted on both real and simulated data sets^{Pf} and it has been reported in several instances (though not all) to lead to better predictions in calibration problems than the standard PARAFAC-ALS^{12,13,78,Pf}.

However, SWATLD hardly ever finds the least squares solution, even when it recovers the correct solution more often than least squares algorithms (e.g., in case of mild

over-factoring)^{Pf}. It has been reported that SWATLD yields solutions that are slightly biased towards higher core consistency^{Pf}, which is a measure of how appropriate a PARAFAC model of given rank is³⁹. This may indicate that SWATLD finds a solution that is less affected by model error³⁹ and would at least partly justify the improved predictions for some calibration problems^{12,13,78,Pf}. The idea that some limited bias may be beneficial is also corroborated by the fact that, for at least one fluorescence data set, ASD yields a solution with higher core consistency than SWATLD's and improved calibration models for two (three when the rank is over-estimated) analytes out of four^{Pf}.

Neither SWATLD nor ASD explicitly penalise lower core consistency and introducing an explicit penalty in this sense to the least squares loss function seems rather problematic as the expression to compute the core consistency includes the pseudo-inverse of all the loading matrices^{23,39}. In fact, there is an algorithm, namely PARAFAC-PDE (PARAFAC with *Penalty Diagonalization Error*) that purports a similar goal by introducing a penalty term in the loss function regulated by a user-defined parameter λ ⁵⁰. Unfortunately, PARAFAC-PDE has analogous conceptual flaws of all the other methods that at each sub-step (and sought loading matrix) optimise a distinct loss function⁸². Hence, its reliability and the quality of its results suffer from the ill-defined optimisation criterion^h.

It has been suggested that ASD could be preferred to SWATLD because it is much faster⁸². On the other hand, when the data array is sufficiently small, a carefully implemented SWATLD has been found to be as fast as ASD^{Pf}. Thus, the compression step embedded in ASD could be regarded as the likely reason for the good performances this algorithm yields on larger arrays. However, compression can be used with success together with many algorithms (cf. section 2.7.6), which suggests that also SWATLD could be made faster. Given the unknown properties of the SWATLD fitting criterion, additional studies will be necessary to verify this claim.

^h Albeit the convergence properties of this algorithm have not been studied in detail, preliminary tests on both EEM fluorescence and simulated data sets showed that PARAFAC-PDE converges to different minima and visibly different solutions depending on the initial value; that the core-consistency of its solutions does not change monotonically with the value of λ as one would expect; and, more importantly, that this method hopelessly diverges for certain values of λ .

Table 2.2a. Original SWATLD algorithm. \mathbf{D}_A , \mathbf{D}_B and \mathbf{D}_C are diagonal matrices holding the norms of the columns of \mathbf{A} , \mathbf{B} and \mathbf{C} respectively; $\bar{\mathbf{a}}_i$ denotes the i -th row of \mathbf{A} , $\bar{\mathbf{b}}_j$ the j -th of \mathbf{B} and $\bar{\mathbf{c}}_k$ the k -th of \mathbf{C} ; \mathbf{X}_i , \mathbf{X}_j and \mathbf{X}_k are the $(J \times K)$, $(I \times K)$ and $(I \times J)$ slabs of \mathbf{X} obtained by fixing the running index in the first, second and third mode respectively, and $\text{diag}(\mathbf{M}) = [m_{11} \ \cdots \ m_{FF}]$ for an $F \times F$ matrix \mathbf{M} .

-
- 0) Initialise \mathbf{A}_0 and \mathbf{B}_0
 - 1) $\bar{\mathbf{c}}_k = 0.5 \text{diag}[(\mathbf{B}^+ \mathbf{X}_k^T \mathbf{A}) * \mathbf{D}_A^2 + (\mathbf{A}^+ \mathbf{X}_k \mathbf{B}) * \mathbf{D}_B^2]$ for $k = 1 \dots K$
 - 2) $\bar{\mathbf{b}}_j = 0.5 \text{diag}[(\mathbf{C}^+ \mathbf{X}_j^T \mathbf{A}) * \mathbf{D}_A^2 + (\mathbf{A}^+ \mathbf{X}_j \mathbf{C}) * \mathbf{D}_C^2]$ for $j = 1 \dots J$
 - 3) $\bar{\mathbf{a}}_i = 0.5 \text{diag}[(\mathbf{C}^+ \mathbf{X}_i^T \mathbf{B}) * \mathbf{D}_B^2 + (\mathbf{B}^+ \mathbf{X}_i \mathbf{C}) * \mathbf{D}_C^2]$ for $i = 1 \dots I$
 - 4) Check convergence.
 - 5) If not converged repeat from 1)
-

Table 2.2b. SWATLD algorithm using the Khatri-Rao product.

-
- 0) Initialise \mathbf{A}_0 and \mathbf{B}_0 so that they are column-wise normalised
 - 1) $\mathbf{C} = 0.5 \mathbf{X}^{(K \times I)} (\mathbf{B}^{+T} \odot \mathbf{A} + \mathbf{B} \odot \mathbf{A}^{+T})$
 - 2) Column-wise normalise \mathbf{C}
 - 3) $\mathbf{B} = 0.5 \mathbf{X}^{(J \times K)} (\mathbf{A}^{+T} \odot \mathbf{C} + \mathbf{A} \odot \mathbf{C}^{+T})$
 - 4) Column-wise normalise \mathbf{B}
 - 5) $\mathbf{A} = 0.5 \mathbf{X}^{(I \times J)} (\mathbf{C}^{+T} \odot \mathbf{B} + \mathbf{C} \odot \mathbf{B}^{+T})$
 - 6) Check convergence.
 - 7) If not converged column-wise normalise \mathbf{A} and repeat from 1)
-

SWATLD is exemplary in showing the conciseness and the efficiency achievable through the Khatri-Rao product. Table 2.2a shows the original algorithm⁵⁷ and Table 2.2b one based on the Khatri-Rao and used in *P/*. The latter formulation is highly vectorised and particularly efficient in the MATLAB[®] environment and may also partly justify the discrepancies in time expense between *P/* and Ref. [82].

Even if the advantage in terms of time expense compared to least squares algorithms is apparent many problems remain unsolved. Apart from the lack of proper understanding of the workings of SWATLD, it is not clear how one could implement weighted

least squares fit and how the algorithm would behave, should the data array contain some missing values. Most importantly, there is no mathematical characterisation of the SWATLD solution and of its bias. This severely limits the use of this method, but seem to justify its use to provide initial estimates for other algorithms.

2.7.4 Hessian based methods

Most of the algorithms for solving nonlinear least squares problems are based on a quadratic model of the optimised loss function (i.e., on a Taylor series of the loss function truncated at the quadratic term)^{94,102}. Thus, given the Hessian \mathbf{H} and the gradient \mathbf{g} of the loss function at the current point, an update $\Delta\mathbf{p}$ for the vector of model parameters \mathbf{p} is found as a solution to the linear least squares problem

$$\mathbf{H}\Delta\mathbf{p} = -\mathbf{g} . \quad (2.15)$$

The different methods differ in the way \mathbf{H} is computed and how (2.15) is solved. Thus, in the Newton method, \mathbf{H} is the true Hessian; in quasi-Newton methods, an approximation of \mathbf{H} (or of its inverse) is obtained via low rank updates to the same matrix at the previous iteration; and in Gauss-Newton algorithms \mathbf{H} is approximated by $\mathbf{J}^T\mathbf{J}$ ^{94,102}.

The Gauss-Newton method is one of the most effective algorithms for nonlinear least squares problems^{23,102,193}. It has been employed (albeit in a slightly modified version) to fit a PARAFAC model to 3-way array as early as in 1982¹²² and a first comparison with ALS was published a few years later (in Japanese¹⁶⁴). No further mention seems to have been made until the introduction of PMF3 (*Positive Matrix Factorization for 3-way arrays*) some fifteen years later^{224,Pl}.

PMF3 uses in fact a damped Gauss-Newton¹⁹³ algorithm that both employs soft line search and a Levenberg-Marquardt scheme to cope with high residuals and model non-identifiability (cf. section 2.2)^{23,102,193,224,Pl}. PMF3 also allows weighted least squares fitting and implements non-negativity constraints, a regularisation scheme and a so-called nonlinear update that is meant to improve the convergence rate in case of large residuals^{224,Pl}. In its original formulation, PMF3 was limited to 3-way arrays^{31,224}, but this restric-

tion has been recently overcome^{PI,PIII}. PMF3 has been shown to be more efficient than ALS in presence of highly collinear underlying factors, but only for relatively small arrays and ranks¹²⁹.

In fact, although PMF3 normally requires fewer iterations to converge, it appears to be slower (in terms of computation time) than a standard Levenberg-Marquardt algorithm and requires a more experienced user because of the numerous settings and additional parameters^{31,155,PI}. Since performances in terms of stability and quality of the solution are very similar, this clearly advocates for the simpler method.

The main drawback of the Gauss-Newton method is that, especially for larger arrays, it is more demanding than PARAFAC-ALS in terms of memory and number of operations^{31,224,PI}. However such assertion is rather simplistic, as there are several factors to take into account^{PIII}.

First of all, so long as \mathbf{H} can be stored in the physical memory, the Levenberg-Marquardt algorithm can be approximately as fast as (or faster than) a single ALS iteration when the order of the array is sufficiently large^{PIII}. The simple explanation for this is that the number of array's elements increases with the product of the sizes in the different orders, whereas the number of model parameters grows linearly with their sum. This also implies that GN is likely more suitable for higher order arrays with similar sizes in the different dimensions while its advantage decreases when some dimensions are larger than the othersⁱ.

Secondly, the fact that solving (2.15) via direct methods has a computational cost proportional to the cube of the model's rank whereas for ALS the cost grows linearly with it indicates that one may expect that Hessian based methods should be faster if the rank of the model is not large^{129,PIII}. However, the solution of (2.15) represents a problem in its own right and certainly deserves further investigations. For instance, owing to the sparsity of \mathbf{J} (and of the second order part of the Hessian^{23,102,PIII}), iterative methods to solve the

ⁱ This is easy to see if one considers that a $20 \times 20 \times 20 \times 20 \times 20$ array requires 100 parameters per unit rank while a $10 \times 10 \times 10 \times 20 \times 160$ array (which has the same number of elements) requires 210 parameters per component.

system of normal equations may be much faster^{23,306}. Indeed, this option leads to the truncated Newton methods¹⁰², whereby an approximate solution to (2.15) is found by limiting the accuracy or the number of iterations of the linear least squares solver¹⁰². In this case, preconditioning becomes crucial to obtain good approximations of the update direction within few iterations. If the number of estimated parameters and the residuals are large, quasi-Newton methods, and particularly limited memory quasi-Newton methods, should be preferred^{94,102,193}, because at each step they require a number of operations that is proportional to the square of the number of parameters. Such methods are mentioned only briefly in the PARAFAC literature^{10,253} and, although they are among the most efficient algorithms for nonlinear optimisation, they were found to be slower¹⁰ or approximately as fast²⁵³ as ALS with line search extrapolation. However, substantial improvements have been accomplished for quasi-Newton methods since these results were obtained^{162,193} and further analyses are indicated.

Thirdly, Hessian based methods typically require fewer iterations to converge (in the worst cases few hundreds) than ALS, which may need several thousands, even when line search extrapolation is used^{129,PI,PII}. This gap is broadened for ill-conditioned problems and when the data array contains larger fractions of missing elements, in which cases Gauss-Newton is likely to be much faster than ALS even if the single iterations are more expensive^{129,155}. For example, Table 2.3. reports the performances of the Levenberg-Marquardt and ALS for two high collinearity problems¹²⁹ and a relatively large EEM fluorescence data set³⁷.

Finally, the explicit formulation of the nonlinear part of the Hessian and its being relatively inexpensive to compute, may represent a great advantage for damped Newton methods, particularly in the solution of relatively small problems (e.g., the difficult GE-MANOVA models described in section 2.3³⁸). It has been reported that, for PARAFAC, Newton methods do not determine a reduction in the number of iterations compared to damped Gauss-Newton²²⁴. An interesting possibility, then, would be to use a hybrid method based on Levenberg-Marquardt that switches to Newton (or quasi-Newton) only when this is expected to perform better (e.g., to attain quadratic convergence at the final stages even in case of large residuals).

2.7.5 Preconditioned Conjugate Gradient

Preconditioned Conjugate Gradient (PCG) algorithms have been advocated already in 1995 to fit PARAFAC models entailing the estimation of thousands of parameters²⁵³; two general purpose software packages currently use PCG algorithm for fitting user-defined models and amongst them PARAFAC (viz. the ‘Multilinear Engine’²²⁵ and the ‘Equation Oriented System’³²²). The convergence rate of this method is at most superlinear¹⁰² and the number of necessary iterations tends to be in the same order as the number of estimated parameters unless the Hessian’s eigenvalues are clustered¹⁰²; modifications and additional constraints are also required when the Hessian (or $\mathbf{J}^T\mathbf{J}$) is not positive definite^{102,225}. A suitable preconditioning step is crucial to obtain fast convergence with PCG⁹⁴. An immediate and simple preconditioner corresponds to column-wise normalisation of \mathbf{J} ^{23,102,225,322}. Since the nonzero elements of each column of the Jacobian are formed as a Kronecker product, it is easy to see that this type of preconditioning simply entails the column-wise normalisation of all the loading matrices, with some savings in

Table 2.3. Median (over five random starts) of number of iterations and computation time for three data sets from literature¹⁵⁵. The relative fit decrease convergence criterion was set at 10^{-8} . κ denotes the Jacobian’s condition number. Fluorescence is an EEM fluorescence data set of size $5 \times 201 \times 61$ and rank 3, PP1 and PP2 are synthetic data sets having size $10 \times 8 \times 5$ and rank 3 and 4 respectively. ALSc denotes alternating least squares with compression, LM is the Levenberg-Marquardt algorithm and LMc is the Levenberg-Marquardt method with compression.

Data set	κ		Algorithm			
			ALS	ALSc	LM	LMc
Fluorescence	9.7	Iterations	144	98 (4)	9	13 (2)
		Time (s)	5.46	0.83	5.36	0.86
PP1	206.2	Iterations	3188	3400 (162)	18	15 (4)
		Time (s)	19.97	20.19	0.44	0.48
PP2	331.1	Iterations	8064	15068 (2)	25	31 (2)
		Time (s)	48.44	105.61	1.03	1.45

terms of computational load.

Conjugate gradients methods have stricter requirements for the line search procedure than other nonlinear least squares solvers^{94,102}. Although soft line search can still be used^{94,225}, it has been shown that the line search problem for an N -way PARAFAC model can be solved by finding the minimum of a polynomial of degree $2N$ ^{93,PIII}. The coefficients of such polynomial can be found by fitting one of the same degree to $2N+1$ points (at the cost of an identical number of loss function evaluation), or directly⁹³. Using some properties of the Khatri-Rao product allows one to reduce the cost of the direct procedure of the original 2^N loss function evaluations to N function evaluations and 2^{N-1} Khatri-Rao products of $N-1$ loading matrices. Even though this makes the direct option preferable for arrays of order 3 or 4, the fitting approach is faster for higher orders^{PIII}.

Some preliminary tests have shown that a standard implementation of Preconditioned Conjugate gradients with exact line search requires more iterations than ALS with line search extrapolation, but to this day, apart from some theoretical observations in the original publications concerning only PCG and PARAFAC-ALS^{225,322}, there has been no systematic and thorough comparison between PCG and other methods.

2.7.6 Compression

Compression is a procedure that can render Hessian based methods applicable for large problems and most other algorithms more well-behaved with respect to overfactoring^{33,131,155,PI} (see Table 3). The theoretical base for compression is the CANDELINC theorem⁵³. In essence, given an orthogonal basis \mathbf{U}_n that spans the systematic variation in the n -th mode, the loading matrix \mathbf{A}_n for the same mode can be expressed as $\mathbf{A}_n = \mathbf{U}_n \mathbf{T}_n$, for some \mathbf{T}_n , and the PARAFAC model for the N -way array \mathbf{X} becomes:

$$\begin{aligned} \text{vec} \mathbf{X}_1 &= (\mathbf{U}_N \mathbf{T}_N \odot \dots \odot \mathbf{U}_1 \mathbf{T}_1) \mathbf{1}_F + \text{vec} \mathbf{E}_1 = \\ &= (\mathbf{U}_N \otimes \dots \otimes \mathbf{U}_1) (\mathbf{T}_N \odot \dots \odot \mathbf{T}_1) \mathbf{1}_F + \text{vec} \mathbf{E}_1. \end{aligned} \quad (2.16)$$

Premultiplying by $\bar{\mathbf{U}}^T \equiv (\mathbf{U}_N \otimes \dots \otimes \mathbf{U}_1)^T$ one obtains:

$$\bar{\mathbf{U}}^T \text{vec} \mathbf{X}_1 = (\mathbf{T}_N \odot \dots \odot \mathbf{T}_1) \mathbf{1}_F + \bar{\mathbf{U}}^T \text{vec} \mathbf{E}_1. \quad (2.17)$$

The CANDELINC theorem demonstrates that $\|\bar{\mathbf{U}}^T \text{vec} \mathbf{E}_1\|_2^2 = \|\text{vec} \mathbf{E}_1\|_2^2 + C$, where C is a constant independent of $\mathbf{T}_1 \dots \mathbf{T}_N$ ⁵³. Thus, the loading matrices of a PARAFAC model fitted on $\bar{\mathbf{U}}^T \text{vec} \mathbf{X}_1$ identify the linear transformations $\mathbf{T}_1 \dots \mathbf{T}_N$ and the loading matrices \mathbf{A}_n give the least squares solution for the projection of the array on the selected bases. In order to obtain the least squares solution on the original array (and not only on its projection), a refining step is required whereby the loading matrices obtained in this fashion are used as initial values^{33,Pl}.

The various compression algorithms differ on how the bases for compression are found and how many modes are compressed^{33,131,155}. For example, the first F left singular vectors of \mathbf{X}_n can be used to compress all modes^{10,255} or only some of them^{139,263,334}; similarly, a Tucker2 model has been fitted on a three-way array to obtain compression bases for two modes^{47,300}; Bro and Andersson explored some of these alternatives and concluded that the best results are obtained using a compression based on Tucker models and operating on all modes³³. More recently a compression procedure has been proposed that is specifically designed to deal with high collinearity between factors¹⁵⁵. No comparison has been made between such an algorithm and one that retains orthogonality of the compression bases but uses, e.g., Hessian based methods.

An interesting case arises when one desires to extend compression to reduce the size of χ^2 and ML problems^{195,Pl}. Several possibilities have been submitted^{31,255-257}. Under the assumption that the measurement errors are normally distributed there is a simple solution to this problem and the variance/covariance matrix \mathbf{S}_G for the fitting errors on the elements of a compressed N -way array $\underline{\mathbf{X}}$ is

$$\begin{aligned} \mathbf{S}_G &= \text{var} \left[\bar{\mathbf{U}}^T \text{vec} \mathbf{X} - (\mathbf{T}_N \odot \dots \odot \mathbf{T}_1) \right] = \\ &= \text{var} \left[\bar{\mathbf{U}}^T \text{vec} \mathbf{X} - (\mathbf{U}_N^T \mathbf{A}_N \odot \dots \odot \mathbf{U}_1^T \mathbf{A}_1) \mathbf{1}_F \right] = . \\ &= \text{var} \left[\bar{\mathbf{U}}^T (\text{vec} \mathbf{X} - \mathbf{Z} \mathbf{1}_F) \right] = \bar{\mathbf{U}}^T \mathbf{S} \bar{\mathbf{U}} \end{aligned} \quad (2.18)$$

where the last equality is applicable because $\bar{\mathbf{U}}^T$ has full row rank¹⁹⁵. The computation of \mathbf{S}_G can be further simplified if \mathbf{S} is a Kronecker product of matrices of appropriate size³¹⁰. One can immediately see that even if \mathbf{S} is diagonal, \mathbf{S}_G need not be so. Consequently,

apart from the trivial case where $\mathbf{S} = \sigma^2 \mathbf{I}$ (which implies that \mathbf{S}_G is also a multiple of an identity matrix), the residuals for the compressed array are not independently distributed even when independence is assumed for the fitting errors in the original array. It is noteworthy that this approach circumvents possible problems with orthogonality between the truncated bases used for compression and the weights array $\underline{\mathbf{W}}$ ³⁷.

3. Alignment and Dynamic Programming

Low-rank multilinear models are widespread in the scientific community both for qualitative and quantitative analyses^{31,143,199,278}, but, like any model, they are only a simplified representation of the data and work under a number of assumptions^{67,196,278,291}. In order for a model to be useful, though, it is not necessary that all the underlying conditions be exactly met and limited departures from ideality are often tolerated^{31,235}. In fact, such deviations are almost unavoidable in common practice and sometimes their magnitude and nature require the choice of a different model that can accommodate for them^{31,143,195,235,267,278} (cf. section 2.4 for PARAFAC). Alternatively, appropriate data pre-treatments can be applied as to remove them or at least reduce their influence on the results.

Shift is a well known issue in chemistry and in signal processing and, in many problems, entails a violation of one of the assumptions of low-rank multilinear modelling, namely, that one variable must relate to the same phenomenon in all samples. When fitting low-rank multilinear models, the presence of shift in the data constitutes an additional source of systematic variation that needs to be accounted for^{118,196,317,PIV-PVI}. Hence, unless appropriate preprocessing is applied, the pseudo-rank of the data array is

increased by shift and the chemical information may be confounded with it, preventing the useful interpretation of the extracted multilinear factors or their application for calibration purposes^{196,317,PIV,PV}. The aim of alignment procedures, which are also referred to as 'warping'^{150,214,PIV-VI} or 'registration'²⁴² methods, is to correct the shift so that the data analysis is limited to the variation relevant to the study. Such procedures, and especially those based on dynamic programming¹⁷, are the main subject of this chapter.

Dynamic programming (DP)^{17,125} is an algorithmic technique that allows to efficiently solve optimisation problems that can be expressed through a recursive relationship and which meet certain assumptions (cf. section 3.3.1). The peculiarities of DP-based alignment methods have thus been investigated in order to establish the feasibility of two such algorithms, namely Correlation Optimised Warping (COW)²¹⁴ and (rigidly) slope-constrained Dynamic Time Warping (DTW)²⁵⁹, for the alignment of chromatographic data as a pretreatment step before the fitting of low-rank multilinear models^{PIV-VI}.

One important remark is that, while shift may be a problem when one desires to fit a multilinear model, it can also provide useful information (e.g., the chemical shift in NMR spectrometry^{66,305}, the shift due to molecular structural changes in fluorescence¹⁷¹). Therefore, pretreatment algorithms to remove it should not be used indiscriminately and, especially in those cases where part of the shift is meaningful for the purpose of the analysis, one should be as conservative as possible and concentrate only on removing the fraction of shift that represents an actual disturbance³⁰⁵. The conservativeness of alignment algorithms has been, perhaps, the main criterion used in this work to evaluate the performances of registration methods based on dynamic programming^{PIV-VI}.

Finally, it is necessary to point out that this chapter treats the case in which shift occurs only in one mode. For some types of chemical data (e.g., obtained through tandem gas chromatography or 2D electrophoresis) shift may be present in two modes and different algorithms should be employed^{91,103,147,301,302,307}.

3.1 Shift

‘Shift’ denotes the change in place or position of a certain feature (e.g., a spectral line or band in spectroscopy or a peak in chromatography) in a measurement. Thus, for any two samples, the values in the $(N - 1)$ -way array obtained by fixing the index in the mode along which shift is observed should not be directly compared as they may contain the signal of unrelated features.

Borrowing the terminology from functional data analysis, given two curves $x(t)$ and $y(t)$, where t may identify time, wavelength, chemical shift or even a geographical coordinate, one can distinguish range and domain variation. Range variation refers to the fact that x and y may differ at values of t at which they can be compared (viz. the intensity of the signal at a specific value of t may differ). Domain variation, or shift, denotes the fact that x and y should not be compared at a fixed value t_0 , but at two different ones (t_1 and t_2) at which the two curves are in comparable states (for example at the same stationary point). In many cases, the focus of the analysis is solely on range variation and consequently the unwanted domain variation represents a disturbance^{242,243,274}.

In general, shift occurs in *sequential data*, i.e., data where ‘the order or relative position of the measurements carries important meaning’¹¹⁸. Spectra and time series are typical examples sequential data encountered in chemistry. Limiting oneself to these types of data, shift problems can emerge when monitoring environmental, chemical or biological processes^{3,64,86,149-151,241}, performing chromatographic or electrophoretic separations^{14,59,109,142,196,214,246,250,277,299,321} and in certain spectral measurements (e.g., Fluorescence, Raman, NIR and NMR spectroscopy)^{28,87,171,233,299,316,317,337-339,344}.

Retention time shift in chromatographic data is a recurrent problem in chemometrics and can be used to outline the different problems associated to shift and some possible solutions. Due to column deterioration, nonlinear variations in flow, pH or composition of the mobile phase, temperature fluctuations, etc., the peak associated to a compound seldom appears at the same retention time in different chromatograms, even when these are obtained from the same sample (see Figure 3.1)⁶¹.

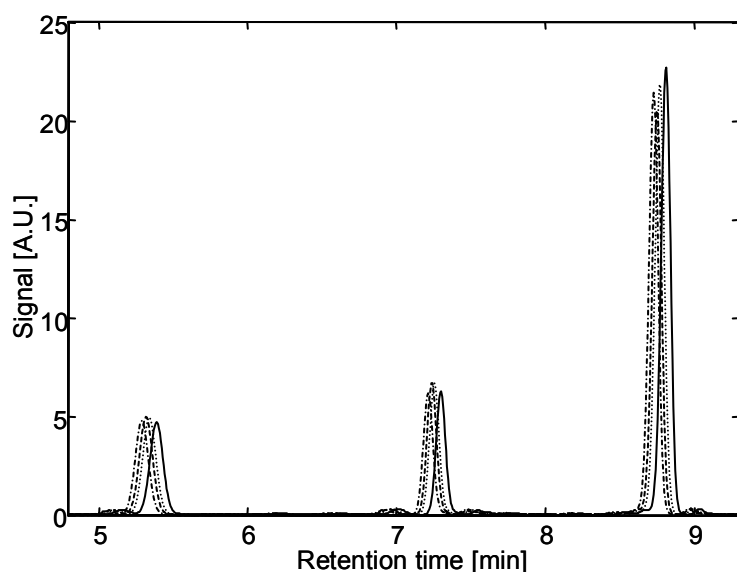


Figure 3.1. Example of shift in chromatographic data. Detail of the chromatograms of four samples of coffee powder with identical composition but having slightly shifted peaks because of column aging. The maximum shift recorded for these measurements is in the order of 20 data points, compared to a peak width at the baseline of approximately 30 points^{P/V}.

Traditional treatment of chromatographic data is comprised of peak identification, normalization (e.g., to correct for changes in sensitivity of the detector or for the amount of sample injected in the chromatograph), and integration^{62,63,73,96}. Once the quantitative information (e.g., concentration or peak area) is thus extracted, multivariate methods can be employed to further analyse the data if deemed useful for the purpose of the investigation⁶¹. Untreated (or insufficiently treated) shift may lead to incorrect peak identification and integration. Failing to assign a peak to the correct analyte (hence, column in a data matrix) may also imply that the corresponding concentration is assigned to a different and most likely adjacent column (analyte), thereby introducing spurious information in a data set. Obviously, this alters the results of multilinear models applied to the quantitative data^{63,PV} and also shows how the shift problem can carry over to data that do not contain actual curves or strictly sequential data.

Even though several attempts have been made to automate the standard procedure and technological advancement has improved instrumental precision and experimental repeatability^{63,73,142,314,315}, traditional chromatographic data analysis usually relies on heuristic and on skill and experience of the analyst as means to cope with retention time shift. This is true especially in case of univariate signals, but also extends, albeit to a lesser degree, to second- (or higher) order instruments (e.g., using diode array or mass spectrometry detection instead of univariate sensors), which are now widely available and

can facilitate peak identification^{96,234,278}. Hence, poor repeatability in terms of retention time may introduce much subjectivity in the subsequent data analysis, which also becomes rather time consuming and expensive, and may induce considerable errors. Therefore, automated procedures as well as analysts would greatly benefit from a reduction, by appropriate pre-treatment, of the time shift between two chromatograms^{61,PV} (cf. section 3.2).

An aspect that adds to the complications of classic chromatographic data analysis is that the peaks of interest are not always known at the time of the investigation and their identification may actually be the purpose of the study^{62,124,333,PV-VI}. Since analysts can only dedicate a limited amount of time to each sample, their attention often concentrates on a restricted number of peaks. This sometimes leads to disregarding substantial amounts of information that could be useful for exploratory purposes^{15,31,62,199,278,PV,PV}. Moreover, even when peaks can be identified, it is not always possible to reliably integrate all of them because of some assumptions on their shape (e.g., Gaussian, exponentially modified Gaussian, Lorentzian, etc.) that are indispensable to integrate insufficiently resolved peaks^{21,63,314,315}.

A natural way to cope with these problems is to analyse the entire chromatogram (or subsections) by means of low-rank multilinear models, whose usefulness for both calibration and qualitative studies is well documented and which can improve the objectivity of the analysis^{31,62,96,199,278}. Unfortunately, their application is severely affected by shift, which leads to a violation of a basic assumption of low-rank multilinearity^{118,196}. Figure 3.2 shows the effect of shift on a PARAFAC model. The landscapes in Figure 3.2a were produced using the spectrum and the slightly shifted chromatographic profiles reported in Figure 3.2b. No noise was added. Without shift, the rank of this simple array would be one, but it is apparent that it is not possible to model such data using only one component. A rank one model explains only 88.48% of the variation in an array with just two replicates and the peak in the time loading does not correspond to any of the peaks in the different replicates (Figure 3.2c-d). Only by adding a second factor, (Figure 3.2e-f) can one explain 100% of the variation.

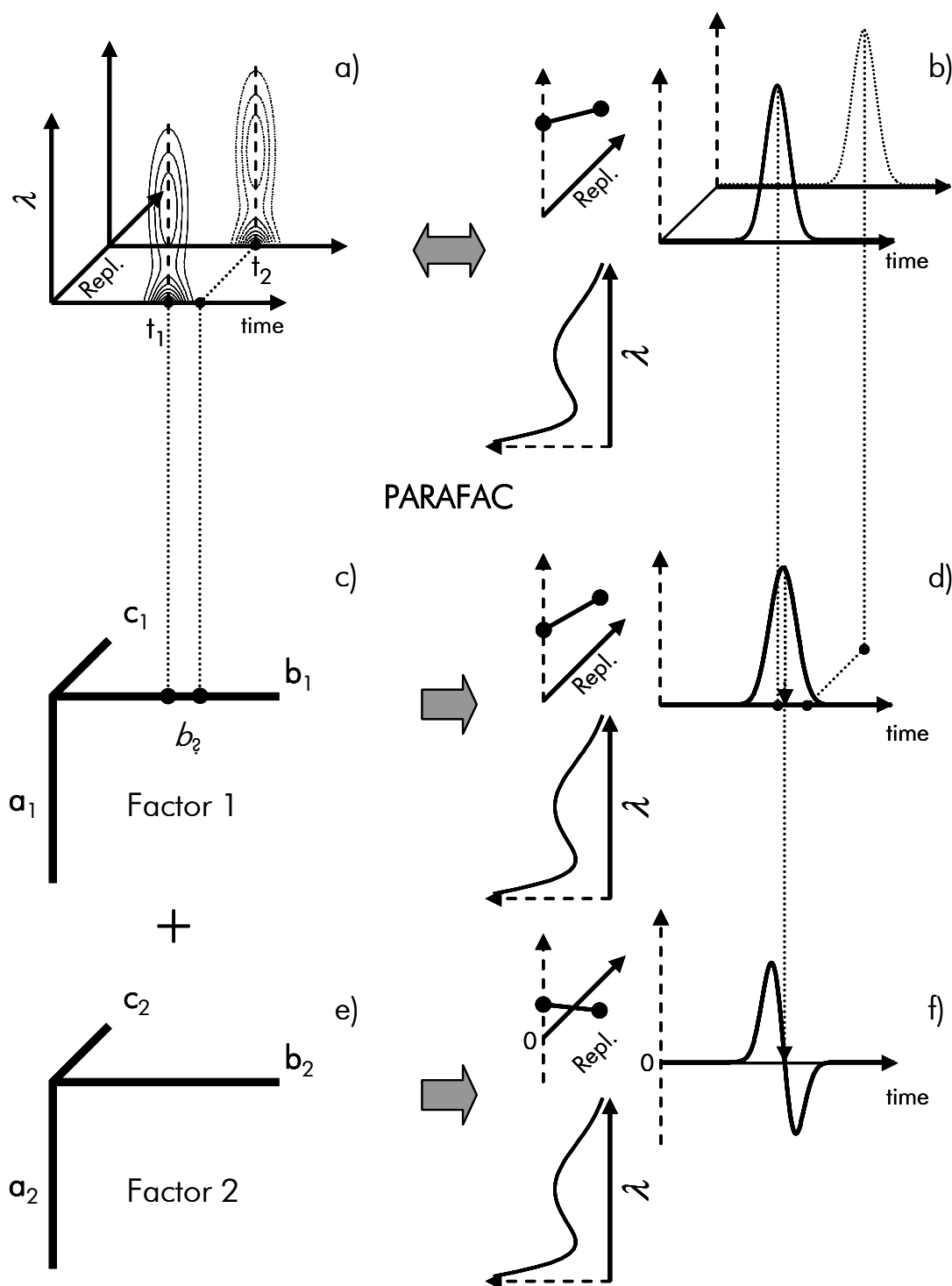


Figure 3.2. Effect of shift on a PARAFAC model. a-b) Two landscapes are generated using two Gaussian time profiles centred at t_1 and t_2 , the same absorption spectra and different "concentrations"; c-d) a rank-1 PARAFAC model extract a component whose time loading is a Gaussian centred somewhere in between t_1 and t_2 ; e-f) a second component is necessary to explain 100% of the variation.

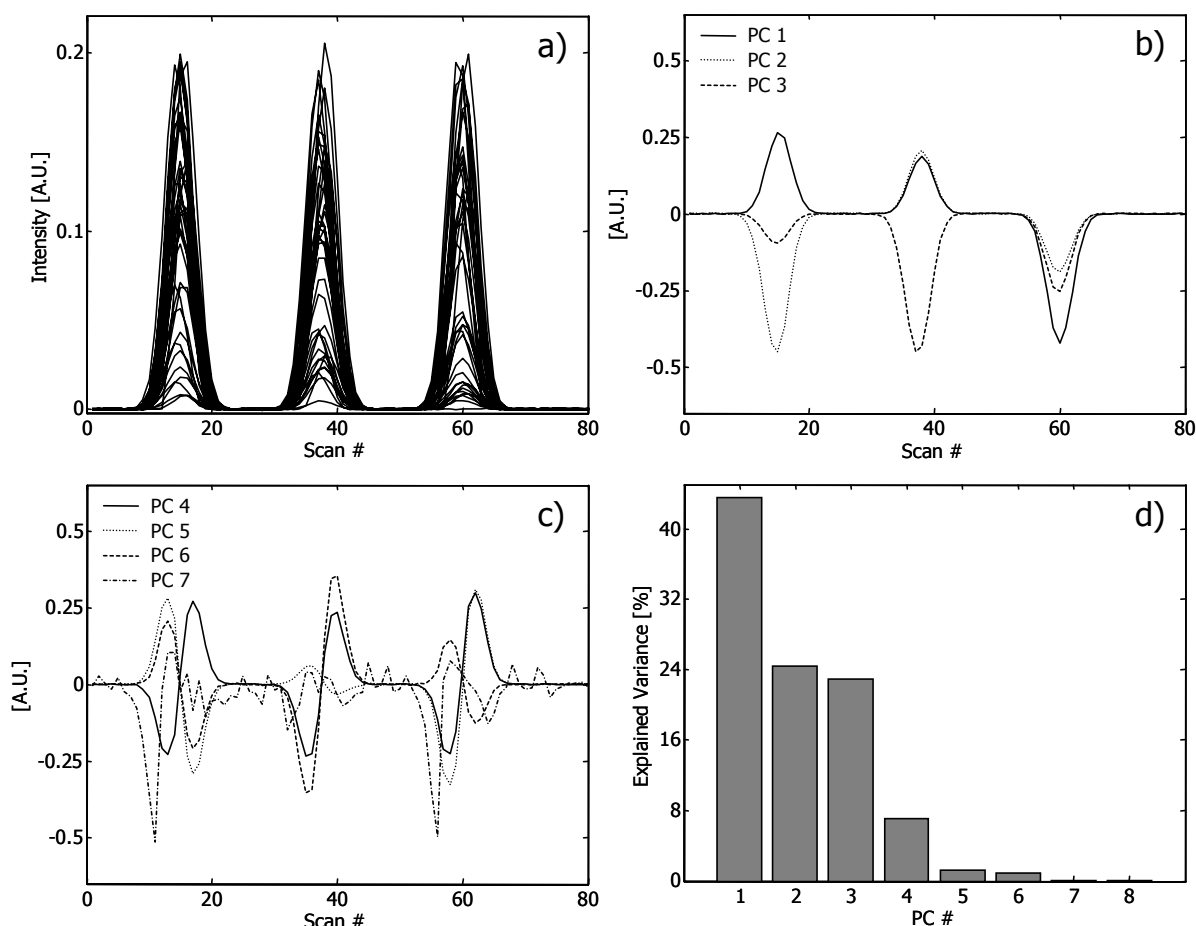


Figure 3.3. Effect of residual shift (± 1 point at the peak tops) on principal components analysis. a) data (40×80) prior to centring. b) The first three PC's of centred data do not contain any visible effect of shift. c) PC's 4 to 6 describe the residual shift, which still explains slightly less than 10% of the variance. Note that even though PC 7 mostly describes noise, the derivative like pattern is still visible. d) PC 4 describes more than 7% of the variation. Hence, in this case, shift related patterns in the loading vectors seem more informative with respect to the pseudo-rank than explained variance.

In this specific case, the second factor has the same loading in the spectral mode as the first, whereas its loading in the time mode looks like the first derivative of the time loading of the first. Thus, the maximum in the time loading of the first component corresponds approximately to the zero in the second factor (Figure 3.2f). It follows that the second factor describes the shift and its loadings in the sample mode change sign depending on whether the shift is towards the left or the right for the actual peak compared to the first factor^{196,PV}.

The first derivative aspect of additional factors in case of shift is well documented and has been proposed as a means to correct for shift in two-way data^{45,86,196,287,344} (cf. section 3.2.1), but is applicable only when the shift is limited^{196,287,337}. Moreover, it should be noticed that, for an unconstrained PARAFAC model, the decomposition for the array described in the figure is not unique, because the two factors have the same loading in the spectral mode (viz. the displayed solution was preferred because it represents a common result when shift is present). Therefore, the extension of such registration methods to the general N -way case may be problematic. However, the idea of aligning a signal using its derivative with respect to the shifted variable (i.e., time in case of chromatography) is also present in other pretreatment methods, which appear more amenable to extensions to higher orders^{1,106,337}.

When the shift is not one of the most relevant sources of variation, it may not appear in low-rank models (e.g., within the first few principal components), but may become visible if the rank of the model is sufficiently high (Figure 3.3a-c)^{61,317,PV}. This often occurs for the so-called ‘coarse alignment’ methods, whose maximum precision in the alignment is limited to the sampling rate of the instrumental scanning^{299,PV}. In these cases, explained variance or related parameters like the *Root Mean Squared Error* (RMSE) may be less useful for determining the correct model rank than observing in which component the effect of shift is first visible (Figure 3.3d)^{61,PV}.

The length of each chromatographic run may vary from sample to sample (e.g., in order to let all constituents elute from the column, certain runs may need to be unpredictably longer than others), and, more generally, between different batches of analysis, where the instrument may have been cleaned, or the column changed or cut, etc.. A similar problem occurs in batch process monitoring, where changes in the initial recipe may induce significant variations in the length of process times, and is further aggravated by sampling rates that vary from batch to batch^{98,149,219,241}. Different length and unequal sampling, if left untreated, impede the very formation of a data matrix/array comprising measurements for several samples and thus the application of the vast majority of the multivariate and multilinear models.

Because of the generality of the shift problem, different methods have been developed to deal with the specific aspects depending on the field of application and type of data (e.g., univariate or multivariate signal, existence of features – landmarks – that can be used for the alignment, etc.). Two main strategies exist to account for shift in the data: modifying the structure of the model in order to accommodate for the shift (e.g., PARAFAC2^{31,96,115,336,PVI}, ‘shifted factors’ models^{64,118,127,128} and, in some instances, Multivariate Curve Resolution^{67,68}) or pre-process the data as to align the trajectories (or the spectra). Either strategy has its drawbacks and it seems unlikely that there is a better method overall. As it is often the case, there are problems and algorithms (or models) that are more or less adequate to deal with them, rather than one method that always works.

3.2 Alignment procedures

Given two curves $x(u)$ and $y(v)$, where $u \in [0, U]$ and $v \in [0, V]$, a *warping path*

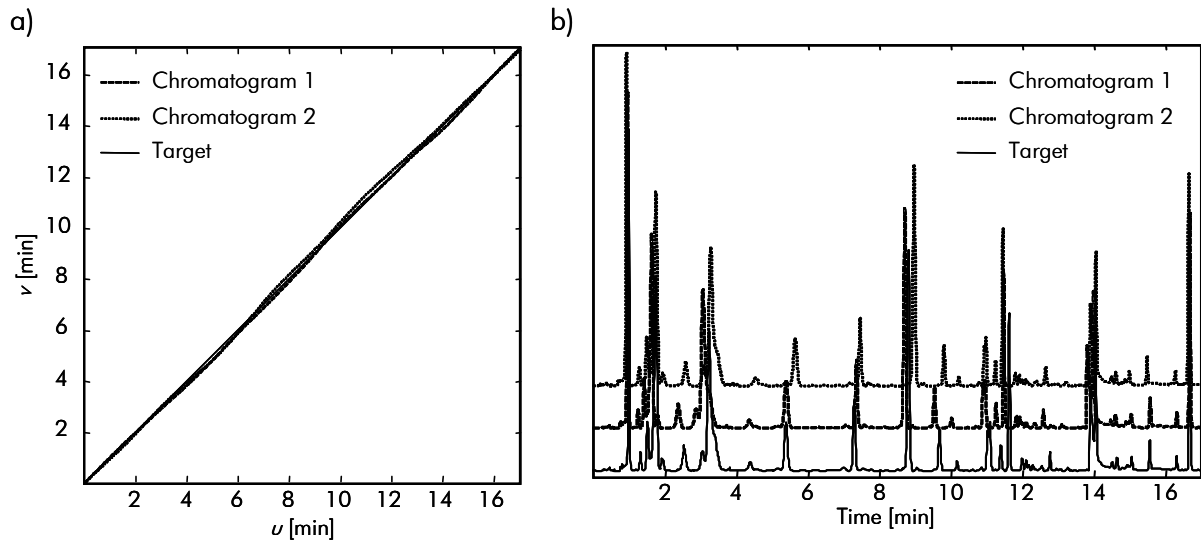


Figure 3.4 a) Warping paths are typically shown in a graph u versus v . Here u here is the time of the target and v is the time of the sample that is being aligned with the target. b) GC-FID chromatograms of coffee powder extracts^{PVI} relative to the warping paths of a). The chromatograms are 2550 points long and are shown with an artificial offset on the ordinates for clarity. For the same reason, shifts are artificially induced and have been accentuated compared to those of original publication.

or *warping function* \mathcal{F} is a mapping:

$$\mathcal{F} \equiv (f_x(t), f_y(t)) \quad \text{for } t \in [0, T], \quad (3.1)$$

where $f_x(t)$ and $f_y(t)$ are strictly increasing functions of t in the interval $[0, T]$ (for arbitrary T), that connects the points of the two curves. In general, $f_x(t)$ and $f_y(t)$ are constrained so that the initial and final points in the curves correspond (that is, $f_x(0) = f_y(0) = 0$, $f_x(T) = U$ and $f_y(T) = V$), although these constraints can be relaxed^{49,149,239,242,324}. Without loss of generality, u and v will henceforth be referred to as *time* (axes) and t as the *common time* (axis). Note that this definition also applies when the two curves take values in a multidimensional space and the single points $x(u)$ and $y(v)$ are vectors, matrices or even tensors.

The purpose of alignment (or registration) algorithms is to remove the unwanted domain variation (i.e., over u and v) so that the subsequent data analysis may focus on the range variation of x and y ²⁴². The idea is thus to seek a warping path \mathcal{F}^* , optimal in some sense, such that

$$x(f_x^*(t)) \cong y(f_y^*(t)). \quad (3.2)$$

This is typically done by minimising some distance measure Q (or, equivalently, by maximising some similarity one) between the two curves:

$$\mathcal{F}^* = \underset{(f_x, f_y)}{\operatorname{argmin}} Q(x(f_x(t)), y(f_y(t))) \quad \text{for } t \in [0, T]^{169}. \quad (3.3)$$

Note that the objective of the warping problem as expressed in (3.2) does not explicitly consider the fact that the two curves may differ because of range variation instead of domain variation. In fact, simply solving (3.3) without limiting the behaviour of the warping path may lead to unsatisfactory results, especially when the two curves are very different from one another (cf. section 3.3.2)^{169, PIV}.

Warping paths can be represented as curves on the (u, v) plane (Figure 3.4a). The region in this system of axes in which the warping paths are sought is referred to as the *search area* (*grid* in the discrete case)^{208, PIV}. When only standard end-point constraints (viz.

$f_x(0) = f_y(0) = 0$, $f_x(T) = U$ and $f_y(T) = V$ are imposed to the warping path, the search area is the rectangle delimited by the lines $u = 0$, $v = 0$, $u = U$ and $v = V$. The dimension of the search area has a great influence on the computational cost of the alignment algorithms and additional constraints can be imposed on \mathcal{F} to reduce it (cf. section 3.3). The main diagonal of the search area denotes a linear relation between the time axes of the two curves and, if these have the same length, the absence of correction.

The solution to the warping problem is inherently non-unique because of the arbitrariness of the variable t and of the interval $[0, T]$ that are used to define \mathcal{F} ^{169,188,324,325}. More precisely, given $h(t)$ a strictly monotonic invertible function such that $h(0) = 0$ and $h(T) = R$, the optimal warping path $\mathcal{G}^* \equiv (f_x^*(h^{-1}(r)), f_y^*(h^{-1}(r)))$ for $r = h(t)$ and $r \in [0, R]$ yields the same alignment as \mathcal{F}^* ¹⁶⁹. In other words, \mathcal{F}^* and \mathcal{G}^* result in the same correspondence between time points in the two curves x and y and are referred to as ‘equivalent warpings’¹⁶⁹.

Hence, while equations (3.2) and (3.3) express the warping problem in a symmetric fashion (viz. warping x towards y and y towards x yield the same results), if $f_x^*(t)$ is also invertible, an asymmetric formulation in terms of u alone can be obtained by setting $t = f_x^{*-1}(u)$. The optimal warping path would then be $\mathcal{G}^* \equiv (u, f_y^*(f_x^{*-1}(u)))$. The latter formulation is particularly useful to obtain an aligned curve \tilde{y} :

$$\tilde{y}(u) \equiv y(g^*(u)) \quad \text{for } u \in [0, U], \quad (3.4)$$

where $g^*(u) \equiv f_y^*(f_x^{*-1}(u))$. One can express *mutatis mutandis* the optimal warping path as a function of v alone, only in this case the result would be a curve $\tilde{x}(v)$ for $v \in [0, V]$ that is aligned to the original y and defined on its domain. Therefore, the intrinsic asymmetry of alignment procedures is apparent, as are the distinct roles of the curves involved: the *target* (or *reference*), which is the curve towards which the other is aligned, and the *sample*, which is the curve that is being alignedⁱ. When several curves need to be aligned,

ⁱ Note that asymmetry may also emerge from some choices of Q or from the constraints imposed to the warping path (cf. sections 3.2.3 and 3.2.1 respectively)^{PV}.

each sample is warped towards the same reference, which may be an actual measurement or constructed from the original curves^{4,149,324,PIV-PVI}.

A formulation of the warping problem that does not require an actual measurement as a reference curve is common in statistics^{95,101,174,188,242,243,250,324,325}. More specifically, it is assumed that all the observed curves are expressions of a common shape function and that a functional relationship exists between the time of this template and the time of the single curves. Thus, if the data set is comprised of I curves y_i for $i = 1, \dots, I$, the following model is expected to hold:

$$y_i(t) \cong a_i \psi(h_i(t)), \quad (3.5)$$

where ψ is the common template, a_i accounts for the signal's intensity, and the *time-synchronising mapping* (or *shift function*) $h_i(t)$ corresponds in essence to the inverse of the warping function as defined in equation (3.4)^{188,324,325}. Hence, ψ has a role similar to the reference in the standard formulation of the warping problem. However, from a statistical point of view, ψ also constitutes a set of parameters that need to be estimated and alignment methods differ in how this estimation is performed and according to which criterion (cf. sections 3.2.4 and 3.2.3 respectively)^{95,135,250,325}.

Equations (3.2) to (3.4) formulate the warping problem in a continuous setting, while the data one works with are most often obtained through the sampling of the curves and are thus discrete. The modifications required to go from a continuous formulation to a discrete one are simple and are well described in the literature^{169,324}; therefore, they will not be shown in detail. Nonetheless, it is important to mention that in the discrete case neither $f_x(t)$ nor $f_y(t)$ is invertible as they need no longer be strictly monotonous^{169,324}. Consequently, a different procedure may be required to obtain \tilde{y} (or \tilde{x}) from the original signal. Particularly in chemometric literature, such step goes under the name of *synchronisation* (cf. section 3.3.2)^{149,233,PIV,PVI}.

In essence, alignment algorithms differ in few basic aspects: the way \mathcal{F} is defined (e.g., parametrically or non-parametrically – cf. section 3.2.1), whether landmarks are used or not to recover it (cf. section 3.2.2), the distance or similarity measure that is

optimised (e.g., Euclidean distance or correlation coefficient – cf. section 3.2.3) and the algorithmic technique that is used to find the optimal \mathcal{F} (for example, dynamic programming^{20,125,283}, graph searching algorithms^{24,125} or simplex optimisation²³⁵ – cf. section 3.2.4).

3.2.1 Models for the warping path

Different ways of expressing \mathcal{F} have been proposed and several aspects can be considered for their systematic description.

A first distinction can be made based on whether the method is parametric or non-parametric. In parametric models, the warping path is an explicit function (e.g., linear or polynomial) of the time and of some parameters (e.g., shift or amplitude of the signals), whose optimal values are found via appropriate estimation algorithms^{95,250}. Depending on the optimisation criterion (e.g., maximum likelihood), parametric models for \mathcal{F} 's may require additional assumptions about the statistical distribution of some parameters^{95,250}. Conversely, in non-parametric methods, no assumptions are made about this functional relationship and the definition of \mathcal{F} is parameter free^{242,250,324,325,PIV}.

However, this classification becomes more complex if one takes into account the estimation of the reference or of the common template that is assumed to have given rise to the data^{101,174,189,250,324,325}. Alignment methods that combine a parametric part (e.g., for \mathcal{F}) with a non-parametric one (e.g., estimating the target curve based on the aligned signals) are referred to as semi-parametric^{174,250,324}. Most DP-based alignment methods are entirely non-parametric^{250,324,325,PIV-VI}. A known exception is the Continuous Profile Model (CPM)^{186,187}, which is based on hidden Markov models (HMM)^{236-238,275} and requires the estimation of a number of distribution parameters.

Another distinctive characteristic of warping paths is whether they are linear or not. Linear warpings are possibly the simplest choice and, in the parametric form, the corresponding warping path is typically defined by $f_x(t) = t$ and $f_y(t) = bt + a$, where a results in a time translation and b is a time scaling factor. These warpings are sometimes referred to as 'affine transformations'⁹⁵. The simple linear warping defined by $b = VU^{-1}$

and $a = 0$ can be used to obtain signals of identical length while by imposing $b = 1$, one avoids stretching and compression of the time axis and limits oneself to time translations^{95,144,250,331}. Especially the latter considerably simplifies the estimation procedures and can lead to great improvements in terms of computation time^{95,250,341}. However, linear warpings are often insufficient for practical purposes even when rather complex estimation procedures are employed (Figure 3.5b-c)^{95,250}. Therefore, they often represent only an initial correction that is further refined using more flexible schemes^{7,15,95,196,305,341}.

Linear warping has a strict relationship with the ‘shape invariant model’ when this is applied to the entire curve or signal^{174,324}. This model assumes that the i -th curve in a data set y_i can be expressed as:

$$y_i(t) \cong c_i \psi\left(\frac{t - a_i}{b_i}\right) + d_i, \quad (3.6)$$

where $\psi(t)$ is again the common template, c_i denotes the intensity variation and d_i the baseline for the i -th curve. Obviously, equation (3.6) is a case of the more general model given by equation (3.5) and consequently the function that relates the time in actual curves to the time in the template is the inverse of the linear warping function.

In chromatography, linear warping is implicitly used when the retention time for the single peaks in a chromatogram is corrected based on the retention time of a known substance (viz. an internal standard) that can be already present in the sample or that has been added for this specific purpose^{59,196}. The normalization parameters a and b are found so that the peak relative to the internal standard always occurs at the same time in all samples. More generally, the optimal parameters are estimated employing ordinary nonlinear least squares solvers or expectation maximisation algorithms^{9,59,95,243,250}.

Also some relatively recent methods that can correct for shifts in spectral data (e.g., NMR or Raman^{45,287,337-339}) with a precision greater than the instrumental one are in fact instances of the shape invariant model where $b_i = 1$ and $d_i = 0$. The novelty of these methods pertains to the algorithm that is used to calculate a_i rather than the underlying

model for \mathcal{F} . In particular, a_i is sought via standard multiple least squares based on the Taylor expansion of equation (3.6):

$$y_i(t) \cong c_i \psi(t + a_i) \cong c_i [\psi(t) + \psi'(t) a_i + \dots], \quad (3.7)$$

where the template ψ is either the loading vector of the first principal component of the data matrix, an actual measurement or an analytical function suitable for the data (e.g., Gaussian or a Lorentzian peak functions)^{45,287,337-339}. Better approximations of the aligned curves (and thus of the template, where it applies) are obtained iterating the procedure aligning the y_i 's with the (provisional) template until the shift parameter a_i is sufficiently small³³⁷.

Nonlinear warping paths are often necessary to correct for complex shift patterns (Figures 3.5a) and several possibilities have been investigated. *Inter alia*: quadratic or higher order polynomial relationships⁷⁶, splines^{101,134,135,242,243,274} (Figures 3.5f and 3.6b) and piece-wise linear warpings^{9,87,142,214,231,262,305,333,341,PIV-PV} (Figures 3.5d-e and 3.6b). Particularly the latter are very common in chemistry applications, both in the parametric (e.g., time normalization according to several internal standards^{9,59,196}) and non-parametric form (e.g., DTW^{15,124,149,246,259,PIV,PVI}, COW^{124,214,233,308,309,PIV-V} and Peak Alignment by Fast Fourier Transform – PAFFT³⁴¹).

A special case of nonlinear (and non-parametric) warping method is encountered in batch process monitoring when the observed curves are multivariate and one of the observed variables is strictly monotonic. If one such variable exists, it can be used to define the warping path and goes under the appellation of *indicator variable*^{98,219}. More formally, let the two monotonic invertible functions $f_a(u)$ and $f_b(v)$ with image $[0, S]$ and domains $[0, U]$ and $[0, V]$ give the value of the indicator variable as a function of time for batches a and b respectively, then \mathcal{F} is straightforwardly defined as $(u, f_b^{-1}(f_a(u)))$, if batch a is used as a reference or $\mathcal{F} \equiv (f_a^{-1}(f_b(v)), v)$ if batch b is the target (Figure 3.7b). Hence, indicator variables correspond to the time-synchronising mappings mentioned in the previous section. Examples of indicator variables are the culture volume in an industrial bioprocess³⁰⁴ or the reaction extent²¹².

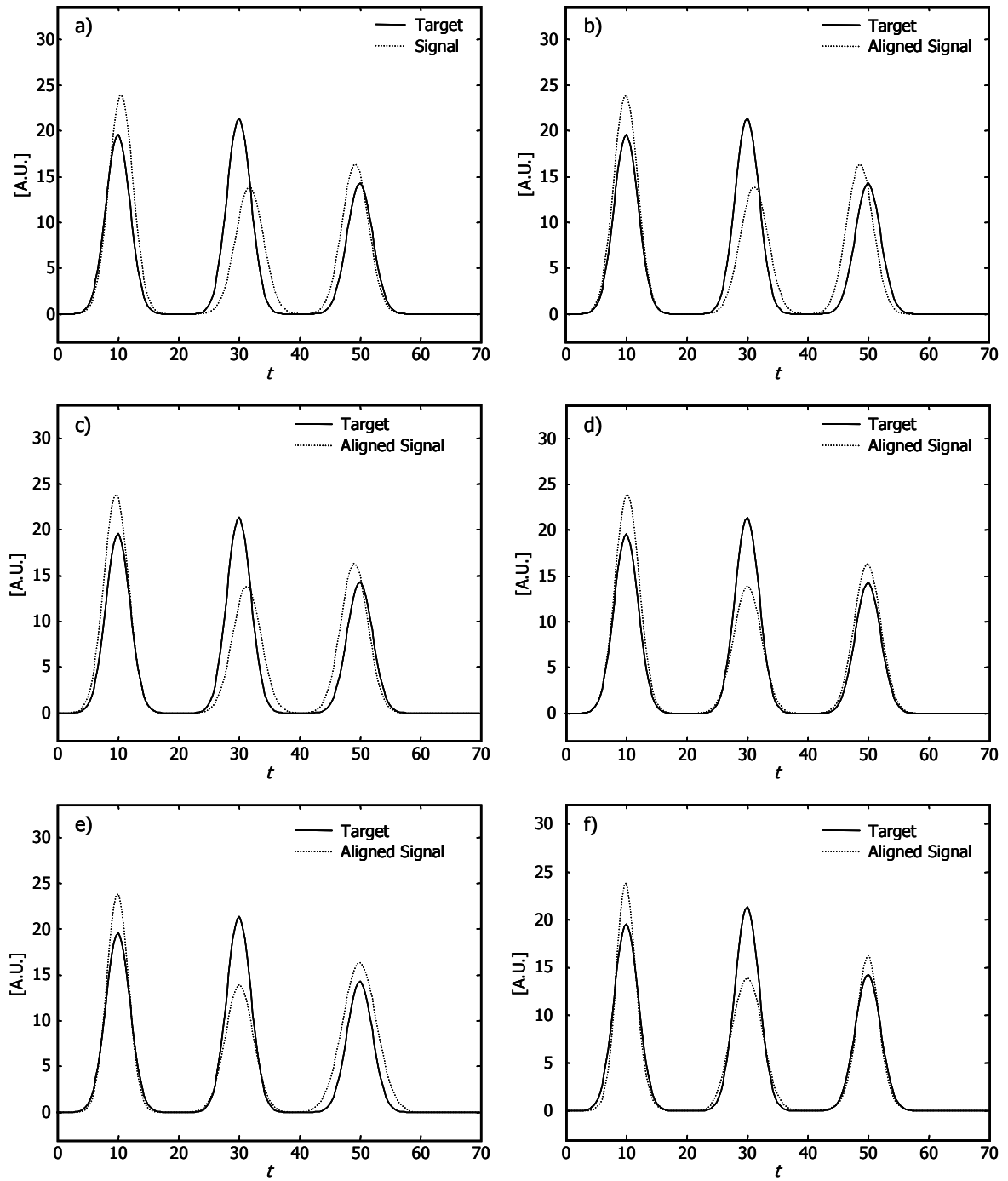


Figure 3.5. a) Simulated target and signal with three Gaussian peaks centred at $t_1 = 10, t_2 = 30, t_3 = 50$ and $t_1 = 10.5, t_2 = 31.7, t_3 = 49.2$ respectively. Best alignments with b) time translation; c) shape invariant model; d) piece-wise linear alignment with local time translation and deletion/insertion; e) piece-wise linear alignment with local time scaling and without deletion/insertion; f) monotone nonlinear regression²⁴²

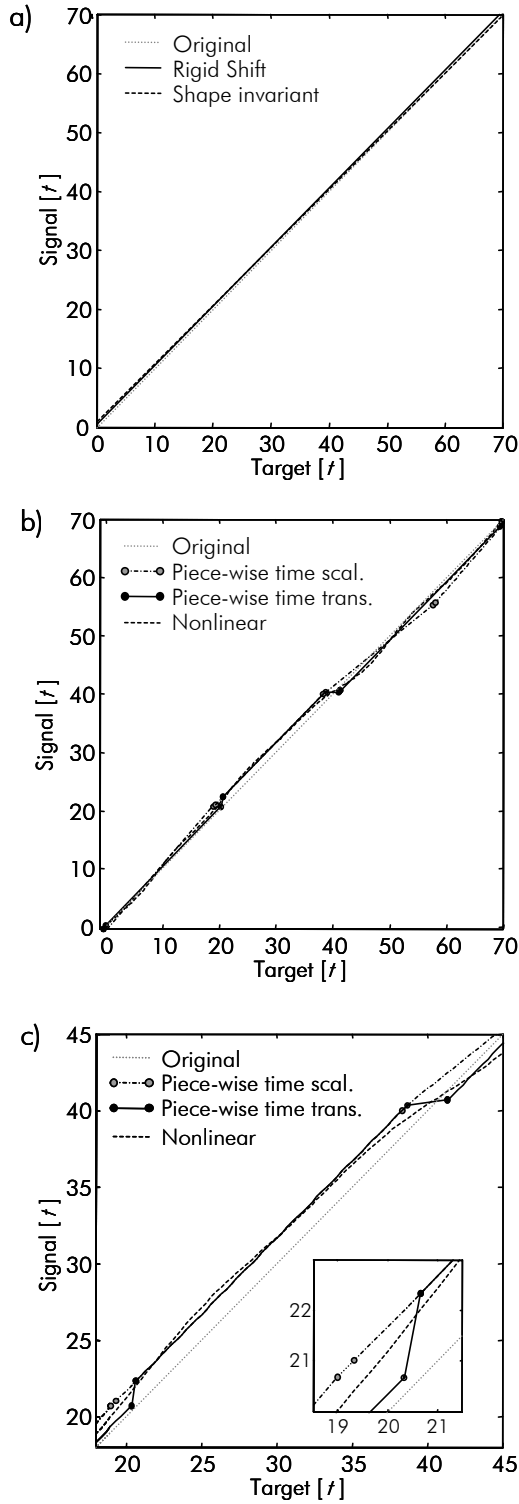


Figure 3.6. Examples of a) linear and b) nonlinear warping paths corresponding to the raw and warped data displayed in Figure 3.5b-f; c) detail of the nonlinear warping paths, in the box the local differences are further highlighted.

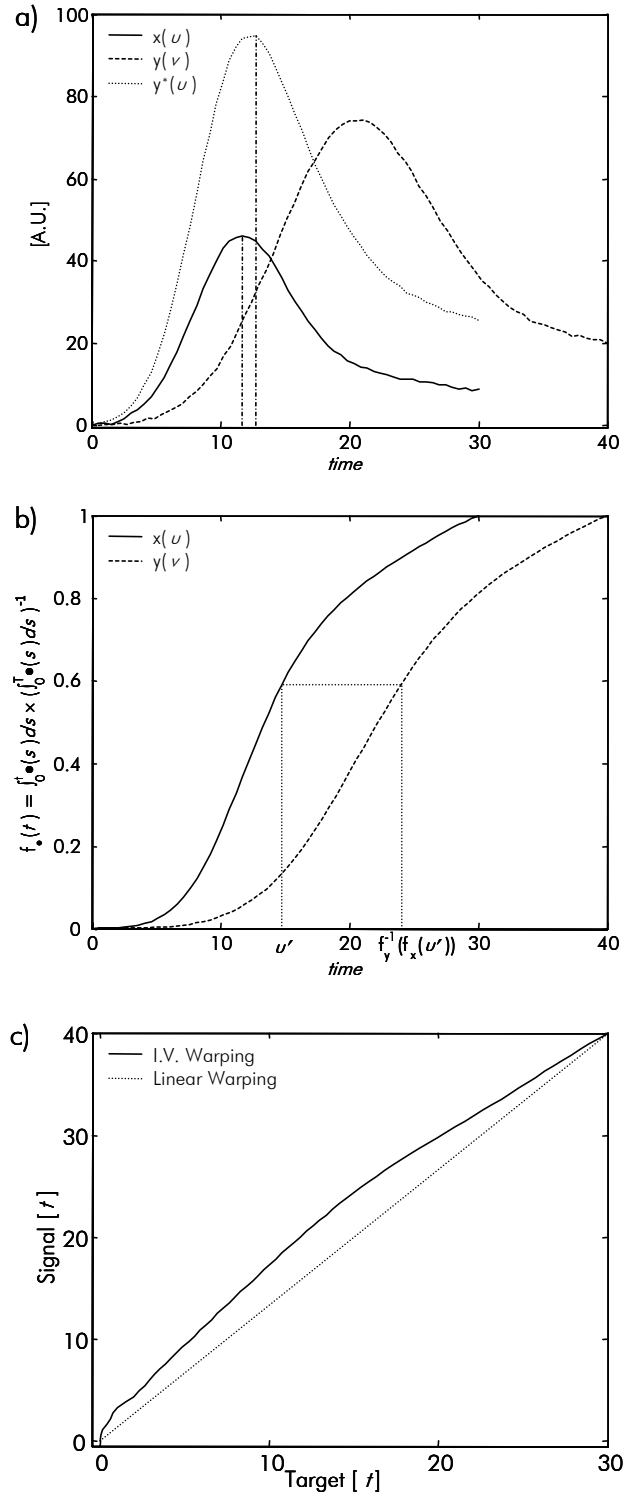


Figure 3.7. Nonlinear warping using the normalised area under the curve as indicator variable. a) Target (x), signal (y) and aligned signal (y^*) y^* is normalised to the same integral as y ; b) indicator variables and correspondence; c) warping path.

The indicator variable need not be an actual one, it can also be calculated from the data. The normalized area under one curve

$$f_x(t) = \int_0^t x(s)ds / \int_0^T x(s)ds \quad (3.8)$$

is a simple example of an indicator variable (Figure 3.7)¹⁸⁸. Note however that even though an indicator variable like (3.8) is also defined on chromatographic data, it is not generally applicable to complex chromatograms when the composition, and thus the height of the peaks, changes (Figure 3.8).

Another possibility for nonlinear warping that has been investigated in the area of process monitoring is to predict the time at which a measurement is taken using the information contained in the measurement itself using a simple linear model^{145,304,340}. More in detail, let \mathbf{X}_k denote the $I_k \times J$ matrix of the I_k measurements of J variables for the k -th batch (sample) and \mathbf{t}_k the $I_k \times 1$ vector of times at which such measurements were taken. Moreover, let \mathbf{t}_k be monotonic and scaled to the closed interval $[0, T]$ (i.e., $t_1 = 0$ and $t_{I_k} = T$). Then, a linear model is sought that predicts \mathbf{t}_k from \mathbf{X}_k for $k = 1 \dots K$:

$$\begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_K \end{bmatrix} \cong \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_K \end{bmatrix} \mathbf{b}, \quad (3.9)$$

where \mathbf{b} is the $J \times 1$ vector of regression coefficients and is typically calculated using Partial Least Squares Regression^{145,304,340}.

The predicted time for the k -th batch, $\hat{\mathbf{t}}_k = \mathbf{X}_k \mathbf{b}$ constitutes then a time-synchronising mapping (indicator variable) that can be used to align the signal. In batch process monitoring, \mathbf{b} is found on a calibration set comprised of batches with normal operating conditions and can be applied to any new measurement, even single data vectors. The main problem of this approach is that the model that is used to predict the process time may induce reversions in the time axis complicating the analysis^{145,330}. An obvious way to overcome this problem would be monotone regression, but, to the author's knowledge, there has been no such an application.

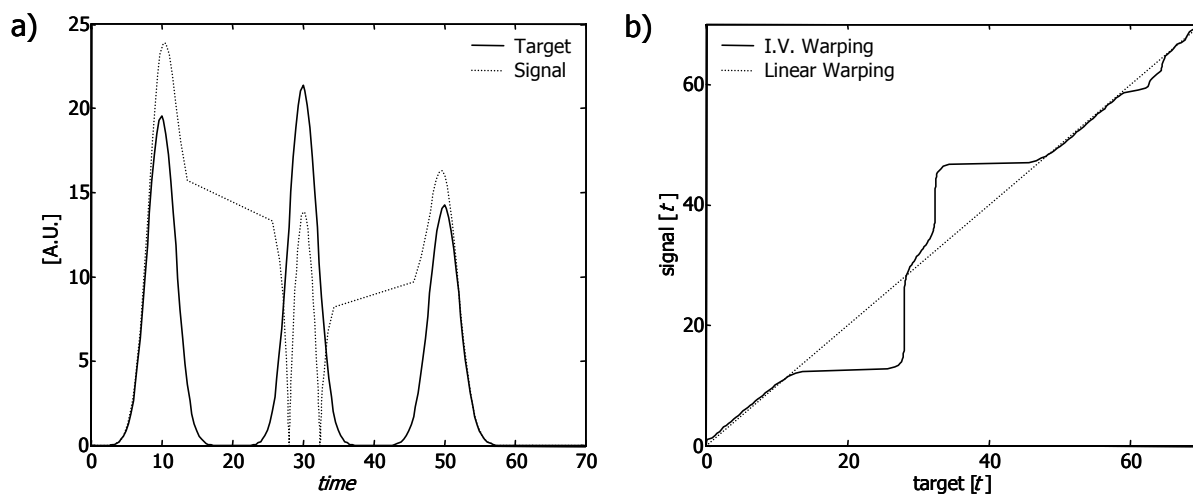


Figure 3.8. Indicator variable warping using the area under the curve. Target and signal are the same as in Figure 3.5a. a) Target and aligned signal; b) warping path. The vertical and horizontal areas in the warping path correspond to the artefacts in the aligned signal.

When there is no indicator variable (or model like (3.9)) that is valid throughout a chemical reaction, distinct variables (or models) can be used at different stages of the process^{98,304}. In this respect, the knowledge of the system that produced the data may help distinguish the different phases^{98,304}. Note that a method that uses different time scales for separate phases of the process is a close relative to landmark registration (cf. section 3.2.2), whereby the end of each stage in the process constitutes a landmark and is assumed to occur at the same time in all batches.

Although extremely flexible, nonlinear warpings based on indicator variables or models such as (3.9) seem inapplicable to most chromatographic or spectral data at least in a naïve form (Figure 3.8). Similarly, quadratic models have been found insufficient for correcting complex shift patterns, with obvious limitations for several types of chemical data. Conversely, for schemes based on splines (Figure 3.5f), the risk is that of overfitting, i.e., admitting too much flexibility in the warping path (e.g., by using too many splines to model the warping path) with the result of yielding distorted curves and misalignments^{101,242}. Artefacts are also common in non-parametric nonlinear alignment algorithms often because excessive flexibility is allowed for the warping path^{124,233,324,PIV}. Various solutions have been proposed to solve this problem either based on different optimisation criteria or additional constraints on \mathcal{F} (cf. sections 3.2.3 and 3.3).

3.2.2 Landmarks

Landmarks, which are also referred to as ‘singular points’²⁸⁴ or ‘structural’ points^{160, 324}, are salient features such as local extrema or discontinuities that are expected to occur at the same place in all signals and thus can be used to synchronise them. The process of aligning signals using such features also goes under the name of landmark registration^{46, 101}. Typical examples of landmarks in chemical data are peaks in chromatography, electrophoresis and NMR^{299, 333}. Peak matching, i.e. the operation by which peaks relative to the same substance in different samples are matched (manually or by automated procedures^{4, 73, 88, 299}), is implicitly a form of landmark registration. The same can be observed about time normalization using reference peaks^{59, 196}, which also sets the base for more elaborate methods^{9, 142, 333, 345}.

Some landmark registration algorithms that have been used in chemistry are listed in Table 3.1a, whereas few alternatives that do not require the identification of these features are reported in Table 3.1b. As can be seen, few procedures exist in both forms and even the distinction between these two classes of pretreatments is quite thin: some hybrid methods exist that, depending on the choice of optimisation criterion, can penalise the absence of landmark’s matching^{134, 135, 324, 325}. The computational effort between the two classes may be quite different and typically to the advantage of landmark-based algorithms^{284, 299}.

By and large, use of landmarks leads to better alignments and faster algorithms under the condition that these features are effectively and correctly identified^{88, 101, 299}. However, the correct identification of landmarks is not necessarily a simple task, and this can severely limit the applicability of these alignment methods^{101, PV-VI}. In particular, it may rely on some external information (e.g., the phases in the utterance of a word²⁶² or in a batch chemical process^{98, 304}) or from additional orders in the data^{108, 234}, but often cannot be automated and manual intervention of a specialist is required^{63, 99, 307}. Numerous proposals to attain robust unsupervised landmark identification exist, e.g., based on wavelets⁴⁶, matched filters¹⁴⁸, identification of extrema, sharp changes or changes in trends^{284, 299}, and statistical methods³³³.

In some instances, like for data arising from the analysis of complex mixtures associated to biological and environmental processes, reliable landmark identification may be unfeasible and other methods have to be used to align the signals^{61,PV}. In particular, missing and heavily overlapping features may prevent correct identification and are major hindrances for landmark based methods (and more in general to most alignment algorithms^{PV})⁸⁸. Algorithmic techniques such as dynamic programming or graph searching procedures have been tested in order to deal with the problem^{284,299}. Additional orders in the data array may help remove all ambiguities and thus obtain perfect alignment in a rather automated fashion provided that the profiles in the additional dimension are sufficiently different for the single landmark²³⁴. The rank alignment algorithm relies exactly on this principle and uses a PARAFAC model based on the GRAM-DTLD algorithm to yield optimal alignments even in case of heavily coeluting peaks²³⁴. A quite obvious limitation of this algorithm and of some of its close relatives¹⁰⁸ is that it works only for second (or higher) order data.

3.2.3 Distance choice

The quality of the results for alignment procedures heavily depends on the loss function Q that one tries to optimise. The Euclidean distance between the two warped curves or Pearson's correlation coefficient are perhaps the two most natural choices, but numerous alternatives have been investigated in order to yield better results^{54,72,101,104,109,149,150,152,153,187,191,208,214,234,241,242,250,284,286,298,299,318,324,325,341,PV}.

More in detail, $Q(\mathcal{F}, x, y)$ is normally defined as the integration (or summation in the discrete case) over the warping path \mathcal{F} of a local distance measure $q(x(f_x(t)), y(f_y(t)))$ between any two points of x and y :

$$Q(\mathcal{F}, x, y) = \int_0^T q(x(f_x(t)), y(f_y(t))) dt. \quad (3.10)$$

Table 3.1a. Some landmark registration algorithms used in chemistry.

Alignment algorithm	Type of Data	Optimisation method	Criterion
<i>Needleman-Wunsch</i> ²¹⁰	Electropherograms ¹⁰⁹	Dynamic Programming ^{17,20,125}	Scored Matches
<i>Peak Alignment using Reduced Set mapping (PARS)</i>	Spectra Chromatograms ^{4,299}	Breadth First Search ²⁴ Dynamic Programming	Harrington's desirability function
<i>Partial Linear Fit (PLF)</i>	Spectra ³¹⁶	Exhaustive enumeration	Euclidean distance
<i>Singular Points augmented DTW</i>	Batch Process Monitoring ²⁸⁴	Dynamic Programming	Local DTW distance
<i>Malmquist-Danielsson</i> ¹⁹⁶	Chromatograms ¹⁹⁶	Exhaustive search and polynomial fitting for fractional shift	Local Correlation
<i>Piece-wise alignment</i> ³³³	Chromatograms ³³³	based on second order information	Correlation Coefficient
<i>Rank Alignment</i> ²³⁴	Chromatograms ^{90,133,234}	Exhaustive enumeration	Residual variance after multilinear modelling of data
<i>Andersson-Hämäläinen</i> ⁹	Chromatograms ⁹	Simplex optimisation ²³⁵	Local Correlation
<i>Kaisha</i> ¹⁴⁸	Batch process monitoring ¹⁴⁸	Exhaustive enumeration	Local congruence
<i>Yang</i> ³⁴⁵	Chromatograms ³⁴⁵	-	-
<i>Crockford et al</i> ⁶⁶	Spectra ⁶⁶	Simplex optimisation	Least squares/congruence
<i>Brown-Stoyanova, Wijes et al</i>	Spectra ^{45,287,337-339}	Gradient search	Least squares
<i>Jeffries</i> ¹³⁷	Spectra ¹³⁷	Simplex optimisation	Penalised Least squares
<i>Johnson et al</i> ¹⁴²	Chromatograms ¹⁴²	Exhaustive enumeration	Match peak position

Table 3.1b. Alignment algorithms used in chemistry that do not require landmark identification

Alignment algorithm	Type of Data	Optimisation method	Criterion
<i>Dynamic Time Warping (DTW)</i>	Batch Process Monitoring ^{72,149,151,241} Chromatograms ^{15,124,246,321,PIV,PII} Spectra ^{233,266}	Dynamic Programming ^{17,20,125,283}	Euclidean distance, Weighted Euclidean distance
<i>Correlation Optimised Warping (COW)</i>	Chromatograms ^{124,136,214,215,308,309,PIV-} Spectra ³⁰⁵	Dynamic Programming ^{17,20,125,283}	Piece-wise Correlation
<i>Piece-wise alignment</i>	Spectra ^{87,175} Chromatograms ²³¹	Genetic Algorithms ^{7,87} Beam Search ¹⁷⁵ Dynamic Programming	Piece-wise Correlation
<i>Continuous Profile Model (CPM)</i> ¹⁸⁷	Chromatograms ^{186,187}	Dynamic Programming and Expectation Maximisation	Maximum Likelihood (Hidden Markov Model)
<i>Parametric Time Warping (PTW)</i> ⁷⁶	Chromatograms ^{76,124}	Gradient Search	Euclidean distance
<i>Gorenstein</i> ^{1,106}	Chromatograms ¹⁷²	Exhaustive enumeration	Least squares
<i>Indicator Variable (IV)</i> ²¹⁹	Batch Processes ²¹²	-	-
<i>Predicted Indicator Variable</i>	Batch Process Monitoring ^{145,304,340}	Partial Least Squares (PLS)	PLS
<i>Horn-Schunk</i> ¹³⁰	Spectra ³²⁹	Gauss-Seidel ¹²³	-

Apart from the (possibly weighted) Euclidean distance, other local measures have been considered like the logarithm of the dot product of the two observed data vector⁴⁴ or some combination of the Euclidean distance between the derivatives of the curves and the Euclidean distance between the actual curves^{232,324,325}.

The relation between the local distance and Q is more complex than it may appear at first sight and great care should be paid to its definition. First of all, the inherent non-uniqueness of the solution to problem (3.3) affects the choice of a suitable loss function. In particular, it can be demonstrated that the value of loss function (3.10) for two equivalent warpings need not be the same¹⁶⁹. If one considers a strictly monotonous invertible function $t = h(r)$ such that $h(0) = 0$ and $h(R) = T$, then the two warping paths $\mathcal{F} \equiv (f_x(t), f_y(t))$ for $t \in [0, T]$ and $\mathcal{G} \equiv (g_x(r), g_y(r))$ for $r \in [0, R]$ are equivalent if $g_x(r) = f_x(h(r))$ and $g_y(r) = f_y(h(r))$. Operating the variable substitution $t = h(r)$ on (3.10), one obtains¹⁶⁹:

$$\begin{aligned} Q(\mathcal{F}, x, y) &= \int_0^R q(x(f_x(h(r))), y(f_y(h(r)))) h'(r) dr = \\ &= \int_0^R q(x(g_x(r)), y(g_y(r))) h'(r) dr \neq Q(\mathcal{G}, x, y) \end{aligned}$$

because $dt = h'(r) dr$.

In order for Q to be invariant over equivalent warpings, it must have the general form

$$Q(\mathcal{F}, x, y) = \int_0^T q(x(f_x(t)), y(f_y(t))) s(f'_x(t), f'_y(t)) dt, \quad (3.11)$$

where $s(f'_x(t), f'_y(t))$ denotes some family of functions of the derivatives of components of the warping path (e.g., their sum or either of the two derivatives alone)¹⁶⁹. Hence, if $s \equiv f'_x(t)$ and operating an identical substitution as before on equation (3.11), one yields:

$$\begin{aligned} Q(\mathcal{F}, x, y) &= \int_0^R q(x(f_x(h(r))), y(f_y(h(r)))) f'_x(h(r)) h'(r) dr \\ &= \int_0^R q(x(g_x(r)), y(g_y(r))) \frac{g'_x(r)}{h'(r)} h'(r) dr = Q(\mathcal{G}, x, y) \end{aligned}$$

where the simple relation $g'_x(r) = \frac{d}{dr} f_x(h(r)) = f'_x(h(r)) h'(r)$ is used¹⁶⁹.

Expressing equation (3.11) in a discrete setting is useful to provide a mathematical explanation for some weighting schemes commonly found in symmetric Dynamic Time Warping and on why such weights are absent in other formulations of the warping problem. Specifically, let $x_n = x(\tau n)$ for $n = 1, \dots, N$ denote the n -th element of the $N \times 1$ vector \mathbf{x} obtained by sampling x at equally spaced points and $y_m = y(\tau m)$ for $m = 1, \dots, M$ the m -th element of vector \mathbf{y} obtained through sampling y at the same rate as x . Then, the warping path takes the form $\mathcal{F} \equiv \{(n(k), m(k)) | k = 1, \dots, K\}$ (see Figure 3.9), and, by defining the function 's' as the sum of $f_x(t)$ and $f_y(t)$, equation (3.11) becomes:

$$Q(\mathcal{F}, x, y) = \tau \sum_{k=1}^K q(x_{n(k)}, y_{m(k)}) w(k), \quad (3.12)$$

where $w(k) = n(k) - n(k-1) + m(k) - m(k-1)$ is one of the most common weights used in Dynamic Time Warping (cf. section 3.3.2)^{149,208,PIV,PVI}, τ denotes the sampling interval and K is the length of the warping path. $n(0)$ and $m(0)$ are defined as equal to zero.

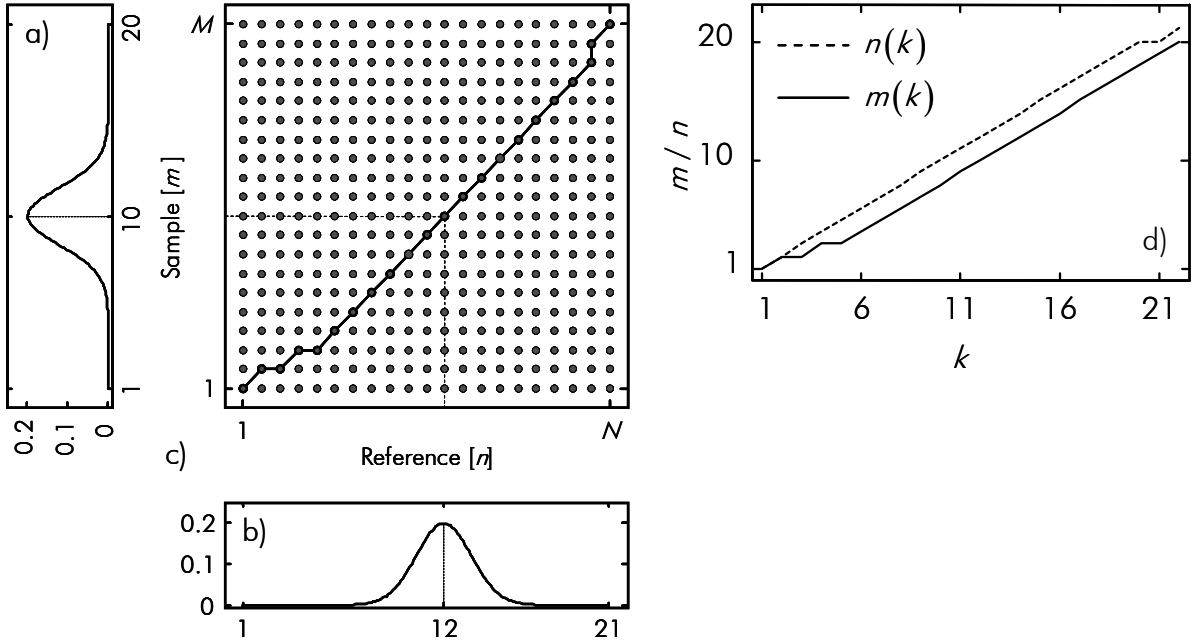


Figure 3.9. Illustration of the discrete warping problem. a) Sample signal, a Gaussian curve centred at 10, sampled $M = 20$ times in the interval $[1, 20]$. b) Reference signal (length $N = 21$). c) Mapping grid, the bold line connecting $(1, 1)$ with $(21, 20)$ is the optimal warping path $\mathcal{F}^* \equiv \{(n(k), m(k)) | k = 1 \dots K\}$; in this example $K = 22$; d) components of the warping path.

Other weighting schemes in the discrete case can be derived from the continuous setting in a similar fashion^{169,208}. In particular, the weighting factor disappears if one expresses the warping function asymmetrically and defines 's' as the derivative of only one of the two components of the warping path. For example, if one imposes $f_x(t) = t$ and s is chosen as $f'_x(t) = 1$, equation (3.11) becomes:

$$Q(\mathcal{F}, x, y) = \int_0^T q(x(t), y(f_y(t))) dt, \quad (3.13)$$

which is still invariant over equivalent warplings. Note that, while this asymmetric formulation allows avoiding explicit weighting, requiring $f_x(t) = t$ infers in the discrete case that $n(k) = k$ and thus that a specific additional constraint be imposed to the warping path (i.e., that $\Delta n(k) = 1$ for $k = 1 \dots N$). Loss function (3.13) is often used in functional data analysis for the warping in the continuous case^{243,324}.

When the reference is not an actual measurement, but corresponds to the template ψ of equation (3.5), it represents a further set of parameters that needs to be estimated and all the recorded signals need be taken into account in the loss function^{242,250,324,325}. Thus, given a data set of I curves $y_i(t)$ for $i = 1, \dots, I$ and $t \in [0, T]$, an example of loss function for an asymmetric warping may be:

$$Q(g_1, \dots, g_I, y_1, \dots, y_I) = \sum_{i=1}^I \int_0^T (a_i y_i(g_i(t)) - \psi(t))^2 dt, \quad (3.14)$$

where $g_i(t)$ and a_i respectively denote the warping path and the amplitude factor for the i -th curve. The minimisation of (3.14) often requires iterative procedures that seek estimates for the warping path, the template and the amplitude parameters^{95,101,134,135,189,243}. This procedure is sometimes referred to as the 'Procrustes method' and, under additional considerations on the distribution of the residuals, can be classified as an Expectation Maximisation method^{71,95}. Loss function (3.14) has been used, for example, to determine the optimal alignment between replicate electropherograms according to a Maximum Likelihood criterion²⁵⁰. A similar loss function has been used by the methods based on a Taylor expansion of the template with respect to t (cf. equation (3.7)), which are conceptually identical to the Procrustes method just outlined^{45,287,337-339}.

The loss functions for the warping problem frequently include a penalty term to avoid extreme warpings and deformations of the sample curve^{134,135,161,241,242,254,286,324}. By and large, the penalty is a (convex) function of the derivatives of the warping path^{286,324}, but can also be implemented as explicit (fixed) weights whenever the warping path deviates from a linear relation^{241,254}. Note that a more direct way of avoiding extreme warpings is to directly constrain \mathcal{F} (cf. section 3.3)^{104,161,170,259,PIV-VI}.

In order to achieve better alignments, the optimised loss function is sometimes defined on preprocessed data (e.g., smoothed, filtered or derivatised^{134,135,142,182,231,PV,PVI}). The main reason for this is to avoid that features such as baseline in chromatographic data or low-frequency noise could result in worse alignments^{4,7,233,PIV-VI}.

3.2.4 Optimisation algorithms for alignment

A wide array of optimization algorithms exists for all the different ways of expressing \mathcal{F} and all the choices for the loss function. Thus, exhaustive enumeration^{142,148,316,341}, dynamic programming^{109,149-151,214,284}, graph searching procedures^{44,175,284,299}, nonlinear optimization methods such as simplex and gradient based algorithms^{9,66,76,137,337-339}, genetic algorithms^{87,170}, neural networks³⁰³ and expectation maximisation⁹⁵ (EM) have all been used to align signals. Not all these methods are suitable for all the possibilities outlined in the previous sections. Hence, for example, nonlinear least squares methods are common for parametric registration and are often preferred for fine alignments because they can achieve a precision for discrete signals that exceeds the instrumental sampling rate^{299,337}. Conversely, graph searching, exhaustive enumeration and dynamic programming are used for non-parametric registration methods limited by instrumental precision²⁹⁹.

Given the different characteristics of the aligned signals (e.g., noisy process variables as opposed to smooth spectral measurements), the choice of which form to use for the warping path and thus which algorithm to employ also depends on the area of application.

In chromatography, landmark-based parametric warpings using either artificial internal standards or natural markers are widely accepted and employed as a standard procedure (cf. Table 3.1a). The corresponding estimation algorithms are often direct or, in more elaborate cases, based on the simplex method and the warping path is assumed linear between successive landmarks^{9,59,63,66,137}. Interesting results have also been obtained with the Procrustes method based on PCA³³⁷⁻³³⁹ and the Maximum Likelihood alignment through an EM algorithm²⁵⁰. Nonlinear non-parametric warpings are less well established in this area of analytical chemistry and in spectroscopy, even though promising results have been obtained through piece-wise linear, non-parametric warping functions sought using graph searching procedures^{88,175,299}, exhaustive enumeration^{142,231,341}, and dynamic programming^{186,187,266,308,309,321,PIV-PVI}. Nonlinear non-parametric time warpings based on splines and using gradient methods have also been used with success, although to a lesser extent¹⁸³.

In process monitoring, nonlinear non-parametric warpings are often preferred. They are typically found by either direct methods based on indicator variables, when available, or using DP-based alignments under relatively mild constraints (cf. section 3.3.2)^{104,145,149,150,182,241,284,304,340}.

Quite some literature exists in terms of comparison of the computational aspects of the different algorithms, both applied to real chemical data and on more general terms^{20,44,88,231,299,341}.

With specific reference to the algorithms based on dynamic programming, which are the main subject of this chapter, a common consideration is that they are slow and expensive compared to other methods^{142,231,299,341}. While true for some methods and implementations, the relative slowness of some algorithms is sometimes dictated by aspects that prescind dynamic programming and rather pertain to the underlying model for \mathcal{F} , the choice of the loss function or even the programming environment.

For example, interpolation makes up for a considerable fraction of the computation time (around 50% in MATLAB) in COW and avoiding it can bring great savings^{142,231,341}. Nonetheless, interpolation is a necessary operation when local time scalings are admitted

because the warping path is not bound to pass through the discrete points in the mapping grid when its slope is not equal to one (cf. section 3.3.3). The question is then which model for \mathcal{F} is more adequate for the observed shifts, and whether the correct warping path is sufficiently well approximated by a simpler model that does not require interpolation (Figure 3.5d-f)^{142,231}.

Likewise, DP is sometimes rejected on the ground that its computational complexity and memory consumption are $O(N^2)$, for N equal to the number of points in the curves being aligned³⁴¹. However, real data hardly ever require that the search area be the entire rectangle delimited by the endpoint constraints (Figures 3.4 and 3.6)^{149,208,228,299,PIV}. For example, recent technology in gas chromatography allows for a more refined control over the retention time, and the maximum observed shift ($\pm A$) tends to be small compared to both peak width and length of the chromatogram^{142,PIV-PVI}. More in general, it is safe to say that when good laboratory practice and strict quality controls are observed, the quadratic complexity of DP algorithms can be brought down to $O(2AN)$ with $A \ll N$ (cf. section 3.3). Under similar conditions, memory consumption can also be reduced to ℓAN bytes, where ℓ is equal to one or two depending on algorithm and implementation (cf. section 3.3). To give a measure of the difference, for a signal 32768 points long and allowing ± 192 points of maximum correction (i.e., $\pm 0.6\%$), a DP-based algorithm would require at most 12MB (24MB) of memory for ℓ equal to one (two) as opposed to 8GB if the whole search grid were to be stored in double precision.

In general, many registration algorithms based on DP, and thus COW and DTW, solve 'shortest path problems' (i.e., they seek the optimal path traversing a directed graph from an initial node to a final one in which each arc is associated to a cost) with a computational complexity that depends on the number of arches that can be part of the optimal path (cf. section 3.3.1)¹²⁵. For example, in COW, the computational complexity of the DP step is proportional to the number of segments of which the warping path is comprised and to how much their endpoints are allowed to move, whereas the cost of the whole procedure is, in large part, due to computing the cost of associated to each arc in the directed graph that represents the problem (cf. section 3.3.3). Similarly, it can be easily demonstrated that the computational complexity of a DP optimisation in a landmark

based method is not proportional to the length of the signal, but to the number of landmarks and to how many matches for each landmark need to be checked²⁹⁹. Therefore, it seems inappropriate to hold DP responsible for the relative slowness of a registration method that recognises landmarks in order to define the loss function but does not make explicit use of them in finding the optimal path and thus has complexity, for the DP part, proportional to the length of the signal²⁹⁹.

Finally, an important factor that needs to be considered is that most algorithmic techniques and programming structures perform differently depending on the computational environment (e.g., compiled or interpreted languages). In general, computational efficiency improves with code vectorisation and by increasing the recourse to highly optimised routines for computing vector and matrix products and other operations like Fast Fourier Transform^{105,235}. This is especially true in the MATLAB environment, where, in spite of recent optimisations, long and nested loops are particularly inefficient and speed of computation benefits from the exploitation of built-in functions².

This type of optimisation has not been explored for the most widespread DP-based alignment methods (namely, DTW and COW) and even the effect of running compiled optimised routines, i.e., the most obvious option to speed up the computations in MATLAB, has not been tested. Considering also that dedicated hardware exists for DP, that this method can be parallelized²⁸⁵, and that several algorithmic techniques can be used to further reduce the search space or vectorise the code^{227,261,PIV}, a great margin for improvements seems to exist.

3.3 Dynamic Programming

Dynamic programming (DP) is a very general optimisation technique that has originated in the field of operations research¹⁷ and is now widespread in many other areas of investigation^{20,125,237,288}.

With specific reference to shift and alignment problems, DP was introduced in the speech recognition field in the late sixties/early seventies^{260,311-313}. While still common in this area and more in general in that of signal processing^{30,146,237,275}, the application of DP-based alignment algorithms has extended to several other fields (e.g., analytical chemistry^{49,104,124,149,187,214,241,246,284,321,PIV-PVI}, genetics^{3,167} and physiology^{29,323}). In particular, the need for automated methods for treating chemical data containing shift and the diffusion of multilinear modelling in chemistry has greatly increased the interest in time warping algorithms and several solutions based on DP have been proposed in recent years.

The first use of DP for the alignment of chromatographic data dates back to 1979 and was mainly aimed at pattern recognition and classification²⁴⁶. The main focus of the paper, though, was on the distance measure provided by the algorithm rather than on the quality of the aligned signals or on applying multilinear models. In a subsequent attempt,

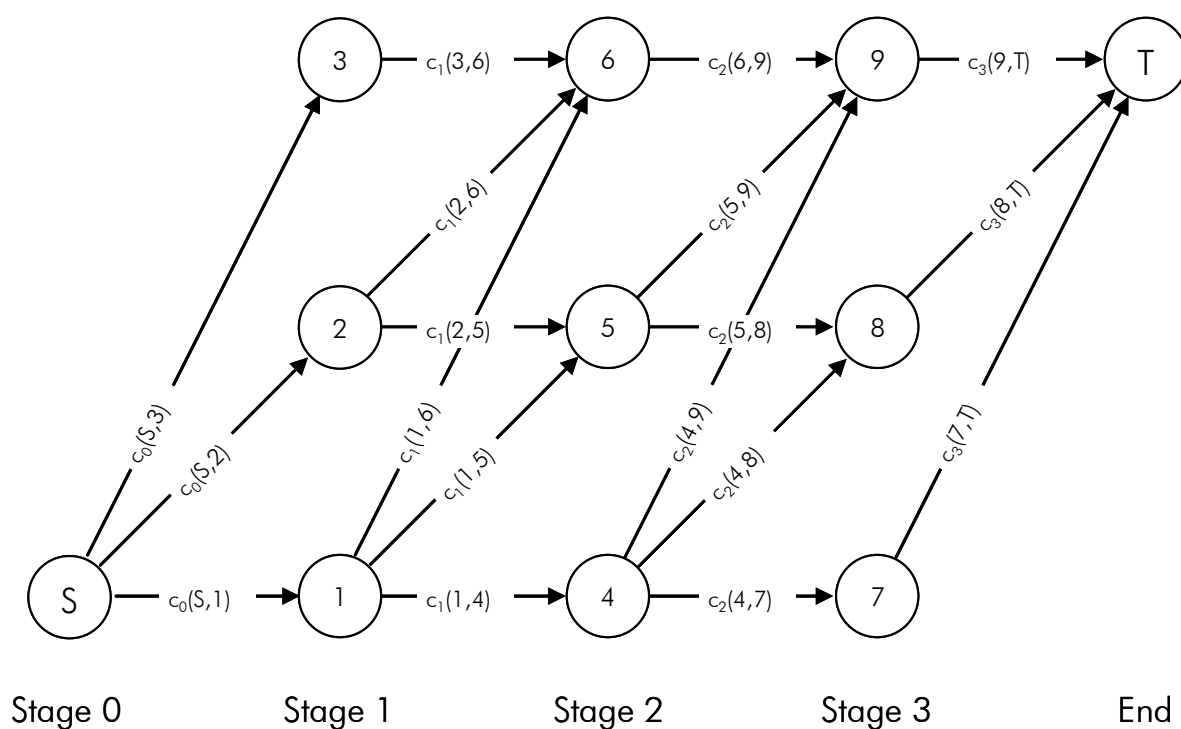


Figure 3.10. Scheme of a classic discrete dynamic programming problem. The circles are the states; the arrows denote the policies and $c_k(s, s')$ the cost for the policy that changes the state from s at the k -th stage to state s' at the $(k+1)$ -th stage. The objective is to minimise the total cost from S to T .

the chromatograms from two different instruments (namely, GC-MS and Gas Chromatography-Fourier Transform Infrared – GC-FTIR) were aligned using an asymmetric DTW algorithm with mild slope constraints (cf. section 3.3.2) and more attention was paid to the quality of the alignment as the procedure was aimed at peak matching³²¹. However, it was only in the past few years that several succesful applications were documented in chromatography for slope-constrained DTW, COW and other DP-based algorithms (see Table 3.1a-b) combined with multilinear modelling. Likewise, the first application of DP for the alignment of batch process monitoring dates 1992⁷², but only some years later were more extensive studies published where alignment was combined with fitting low-rank multilinear models^{149-151,241}.

One of the advantages of DP is that, if the principle of optimality holds (see below), the global optimum is found for the loss function and is thus preferable to methods that cannot guarantee this¹²⁵. The main hindrance to its extensive use in chemistry is that it can be considerably more expensive than other search methods unless specific measures are taken (cf. section 3.2.4)^{44,299}. For example, unconstrained DTW has complexity $O(N^2)$ and is clearly unfeasible for chromatographic and NMR signals, whose length may well be on the order of tens of thousands^{88,299}. The unsuitability of this method for chromatographic data (and, most likely, spectral ones) is well documented also in terms of quality of the alignment^{124,233,PIV}. However, this does not remove the usefulness of DP for other registration algorithms for many chemistry applications^{49,124,136,186,187,241,266,308,309,PIV-PVI}.

3.3.1 Principals^k

Unlike other optimisation methods, there is no standard formulation of the DP problem, but problems that can be solved using DP have some common characteristics that can also be interpreted as requirements for its application¹²⁵.

First of all, the optimisation problem can be decomposed in *stages*, and each stage is associated with a (possibly infinite) number of *states* (Figure 3.10). The set of all possible states for the problem is denoted as \mathcal{S} (e.g., $\mathcal{S} \equiv \{S, 1, 2, \dots, 9, T\}$ in the figure) and the set of states associated to the k -th stage is \mathcal{S}_k (e.g., $\mathcal{S}_2 \equiv \{4, 5, 6\}$). A *policy decision* taken at each stage (an arrow in the figure) transforms the current state to a state associated with the subsequent stage. If this choice is done according to a probability distribution (i.e., if only the probability is known that a certain state will follow from the present one), DP is referred to as *probabilistic*, or *stochastic*¹²⁵, as opposed to *deterministic*. Each policy decision has an immediate contribution $c_{ss'}$ to the loss function, where $s \in \mathcal{S}_k$ and $s' \in \mathcal{S}_{k+1}$ identify the initial and final state resulting from the policy. The objective of DP is to find an *optimal policy* (e.g., the warping path for DTW) for the overall problem.

A fundamental aspect of problems that can be solved through DP is well described by the *Principle of Optimality*: "An optimal policy has the property that whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision"¹⁷. In other words, an optimal policy for the remaining stages given the current state needs to be *independent* of the policy decisions adopted in preceding stages. Therefore, the optimal immediate decision depends only on the current state and not on how one got there. This principle defines the *recursive relationship* that is the basis of DP and the reason for its computational effi-

^k The description given here is restricted to those aspects that are useful to describe registration methods that have been used in chemistry^{P/V-PV}, which, with few exceptions, employ deterministic DP to solve a 'shortest path problem' for discrete data (cf. sections 3.3.2 and 3.3.3). More general treatments can be found on the numerous textbooks on the subject^{20,125,283}.

ciency. Thus, the loss function for the optimal policy at the k -th stage is typically of the form

$$Q_k^*(s) = \min_{s' \in \mathcal{S}_{k+1}} Q_{k+1}^*(s') + c_{ss'} \quad \text{for } k = 1 \dots K-1 \quad (3.15a)$$

or

$$Q_k^*(s) = \max_{s' \in \mathcal{S}_{k+1}} Q_{k+1}^*(s') + c_{ss'} \quad \text{for } k = 1 \dots K-1 \quad (3.15b)$$

where $Q_k^*(s)$ denotes the contribution of the stages from the k -th to the K -th if one starts at state s , applies a policy that leads to state $s' \in \mathcal{S}_{k+1}$ and follows an optimal policy thereafter¹²⁵.

While it is customary to formulate the recursive relationship in a backward fashion (i.e., starting with Q_K^* and moving backwards), the direction in which the stages are considered does not change the overall optimal policy and its cost in deterministic DP. For example, the recursive relationship

$$Q_j^*(s) = \min_{s' \in \mathcal{S}_{j-1}} Q_{j-1}^*(s') + c_{ss'} \quad \text{for } j = 2, \dots, K \quad (3.16)$$

produces the same overall optimal policy as equation (3.15a)¹²⁵. Note that in this case $Q_j^*(s)$ denotes the contribution of all stages from the first to the j -th if one follows an optimal policy up to state $s' \in \mathcal{S}_{j-1}$ and then actuates a policy that brings to state s .

DP yields the optimal policy decision at each stage for each of the possible states and not only the overall optimal policy. Thus, the DP solution also provides the optimal warping path from any point in the allowed search grid, even suboptimal ones. While this can be useful for sensitivity analysis (i.e., to evaluate how, and if, the results change depending on the model parameters), it also implies an added workload compared to other methods, like graph search algorithmic techniques, that are not forced to examine all states^{20,125,206}. Since the difference in performance grows with the size of \mathcal{S} , such techniques may be more suitable than DP for some of the alignment problems described in this chapter (and especially DTW – cf. section 3.3.2) in which the number of states may well be in the order of millions⁴⁴.

The applications in chemistry of dynamic programming have been limited to its deterministic form^{72,109,149-151,182,214,237,246,275,321,PIV}. Probabilistic DP, while a standard in speech recognition²³⁷, is considerably less common in chemistry and has recently been proposed for the alignment of LC-MS data^{186,187}.

3.3.2 Dynamic Time Warping

Dynamic Time Warping (DTW) is the oldest alignment method based on dynamic programming^{260,311-313}. However, apart from two early publications^{246,321}, its application in analytical chemistry is considerably more recent^{104,149,246,321,PIV,PVI}. Only the discrete version of this algorithm is outlined in this section; more details on the continuous case are available in the cited literature^{149,208,259}.

Basic algorithm

Given two vectors \mathbf{x} (the reference) and \mathbf{y} (the sample to be aligned) of lengths N and M respectively obtained from sampling at the same regular rate the two curves $x(\nu)$ and $y(\nu)$, the discrete DTW algorithm seeks a warping path of the form $\mathcal{F} \equiv \{(m(k), n(k)) | k = 1, \dots, K\}$ by solving the following minimisation problem^{132,149,208,259,PIV}:

$$\mathcal{F}^* = \underset{\mathcal{F}}{\operatorname{argmin}} \frac{\sum_{k=1}^K q(x_{n(k)}, y_{m(k)}) w(k)}{\sum_{k=1}^K w(k)}, \quad (3.17)$$

$$\text{s.t. } 0 \leq \Delta m(k) \leq a, 0 \leq \Delta n(k) \leq b,$$

$$m(1) = n(1) = 1, n(K) = N, m(K) = M$$

where $q(x_n, y_m)$ denotes the distance (e.g., the Euclidean one) between x_n and y_m and $w(k)$ a weighting function (cf. section 3.2.3). This discrete warping problem is depicted in Figure 3.9. The first two constraints impose (weak) monotonicity on the two components of the warping path and avoid sudden jumps of more than a or b points along the sample's or the reference's time axis²⁰⁸. The last three constraints fix the endpoints of the warping path and so that the first and last elements in \mathbf{x} and \mathbf{y} correspond. Note that K

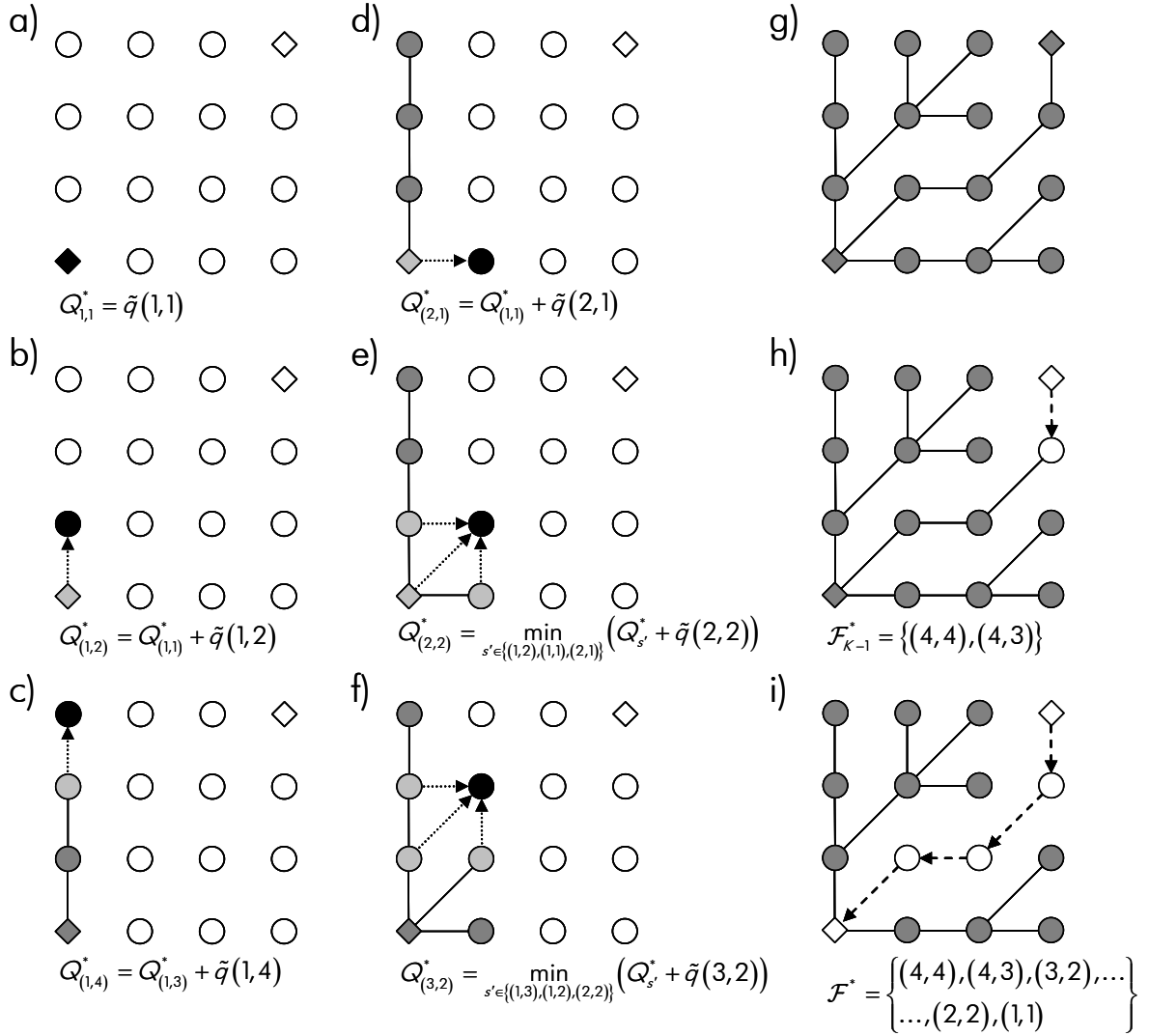


Figure 3.11. Graphical rendition of the (forward) DTW algorithm for $M = N = 4$ and subject to basic constraints $0 \leq \Delta n \leq 1$, $0 \leq \Delta m \leq 1$, $m(1) = n(1) = 1$, $n(K) = N$ and $m(K) = M$. a) to f) illustrate the intermediate stages of the algorithm. The state being evaluated is coloured in black, grey symbols denote states that have already been evaluated and empty ones are for states that have not been evaluated yet. Lines identify the optimal policy, dotted arrows stem from allowed predecessors (in light grey). g) All states have been evaluated and are connected to state (1,1). h-i) Starting from state (4,4) the optimal predecessors are backtraced until (1,1) is reached and the whole \mathcal{F}^* has been retrieved.

depends on the warping path and, unless additional constraints are imposed, it is larger than both N and M . The weights and the normalization factor in (3.17) are essential when one has to compare several loss function values in order to assign a signal to a certain

class because they remove the effect of the length of the warping path and of the signals involved^{149,208,237,275}.

In order to use dynamic programming to solve (3.17), it is necessary that the sum of the weights (i.e., the denominator of the loss function) is constant over any warping path^{208,259}. While standard choices of weights have this property (e.g., for $w(k) = \Delta m(k) + \Delta n(k)$, the sum over any \mathcal{F} is equal to $M + N$), alternative schemes may require ad hoc procedures for the normalisation which result in bias towards longer or shorter \mathcal{F} 's²⁰⁸. For example, if no weights are used (or, equivalently, if $w(k) = 1$ for all transitions in the warping path), the normalisation factor is equal to K and depends on the warping path. Thus, again, DP cannot be used to solve the problem, unless the loss function is modified disregarding the denominator or by fixing it to a constant (e.g., M or N)^{149,208}. In both cases, the optimisation will be biased towards shorter paths, which entail the summations of fewer terms^{149,208,PIV}. It is also worth mentioning that, when DTW is used to remove shift prior to multilinear modelling, there is no actual need to compare two warping distances that have been obtained using signals of different lengths and, consequently, normalisation is commonly skipped²⁴¹. One interesting exception may occur when the endpoint constraints are relaxed and there is a need to compare the warping distances obtained at different endpoints for the same sample and reference¹⁴⁹.

By comparing the numerator of (3.17) with equation (3.15a), one can immediately see that the DP stages in DTW are the discrete time points along the warping path and the set of states contains all the discrete points (n, m) in the search area delimited by the endpoint constraints (i.e., $\mathcal{S} \equiv \{(1, 1), (1, 2), \dots, (N, M)\}$). The decision policies are the transitions $(\Delta n, \Delta m)$ allowed by the constraints and the cost associated to a policy starting from state $s = (n, m)$ to reach state $(n', m') \equiv (n + \Delta n, m + \Delta m)$ is $\tilde{q}(n', m') \equiv q(x_{n'}, y_{m'})w(k)$. The local distance $q(x_n, y_m)$ is typically the Euclidean distance between the measurements at points x_n and y_m , but weighted distances have been used as well for multivariate signals^{04,149,241,PIV}.

The warping path \mathcal{F}^* is then the overall optimal policy sought, which can be represented as the graph that connects state $(1, 1)$ to state (M, N) with the minimum cost (Figure 3.9c). The (forward) recursive relationship for the DTW algorithm is:

$$Q_{(n,m)}^* = \begin{cases} \tilde{q}(1,1) & \text{for } (n,m) = (1,1) \\ \min_{\rho \in \mathcal{P}_{(n,m)}} [Q_{(n-\Delta n, m-\Delta m)}^* + \tilde{q}(n,m)] & \text{for } (n,m) \neq (1,1) \end{cases} \quad (3.18)$$

where $Q_{(n,m)}^*$ denotes the cost of the optimal policy that leads to state (n,m) from state $(1,1)$ and $\mathcal{P}_{(n,m)}$ is the set of allowed policies that lead to state (n,m) from its immediate predecessors^{125,283,PIV}. Note that, with this notation, it is necessary to include the \tilde{q} term in the minimisation because it depends on the policy through the weighting term.

Equation (3.18) shows why problem (3.17) cannot be correctly solved using DP if normalisation is required and the denominator varies depending on the warping path. Since the optimal \mathcal{F} is unknown at the time when the optimal loss function at state (n,m) is evaluated, so is the normalisation factor. Consequently, it is not possible to identify its optimal predecessor (or the corresponding policy) because the choice would depend also on the remaining path, clearly violating the optimality principle.

Figure 3.11 provides an example of how the DTW algorithm works when $M = N = 4$ and \mathcal{F} is subject to the basic constraints $0 \leq \Delta n, \Delta m \leq 1$, $n(1) = m(1) = 1$, $n(K) = N$ and $m(K) = M$. The set of allowed policies that lead to state $s \neq (1,1)$ is in this example:

$$\mathcal{P}_s \equiv \{(1,1), (1,0), (0,1)\}. \quad (3.19)$$

Note that not all policies are applicable at each state: only vertical transitions are allowed for $n = 1$ and only horizontal ones for $m = 1$. Thus, the (forward) recursive relationship to find the optimal value of the loss function at point (n,m) in the search grid is

$$Q_{(n,m)}^* = \begin{cases} \tilde{q}(1,1) & \text{for } (n,m) = (1,1) \\ \sum_{i=1}^m \tilde{q}(n,i) & \text{for } n = 1 \text{ and } 1 < m \leq M \\ \sum_{j=1}^n \tilde{q}(j,m) & \text{for } m = 1 \text{ and } 1 < n \leq N \\ \min_{s \in \left\{ \begin{smallmatrix} (n-1,m), \\ (n,m-1), \\ (n-1,m-1) \end{smallmatrix} \right\}} [Q_s^* + \tilde{q}(n,m)] & \text{otherwise} \end{cases}$$

In order for the forward algorithm to work correctly, it is necessary that the optimal values of the cost function at all the allowed predecessors of the current state have been calculated. Thus, the algorithm must proceed column-wise (or row-wise); and within each column (or row) the index must be increasing (Figure 3.11a-f). The optimal overall policy \mathcal{F}^* can be backtraced from state (N, M) since its optimal predecessor is known (Figure 3.11h-i). Thus, until the last state is reached, \mathcal{F}^* remains undefined and, in general, it is not possible to discard any intermediate state.

Note that the optimal overall policy need not be unique and all possible optimal paths should be returned by the procedure^{125,241}. This event is likely to occur in the case of long stretches of identical values both in the sample and in the reference (viz. if $\tilde{q}(n, m)$ is identical for numerous consecutive values of both n and m). In practice, this situation seems unlikely to become a problem, because the chances that two (or more) very different paths lead to the exactly the same loss function in double precision appear to be rather small. The numerical stability of this alignment method in chemistry applications, though, has not been studied yet and any conclusion in this sense appears highly speculative.

The great computational efficiency of DTW follows from the fact that all states must be visited only once. However, it is apparent that the complexity of this algorithm is $O(MN)$ when the signals are univariate, and $O(LMN)$ if L values are measured at each time point. Moreover, even though only a code identifying the optimal policy among the few (three for $a = b = 1$) admitted by the continuity constraints need to be stored for each state, such a value has to be stored for all the discrete points in the search area in order to retrieve \mathcal{F}^* . Thus, the memory consumption is also in the order of MN . Since in spectral and chromatographic data M and N can be in the order of 10^4 and L in the order of 10^2 , the basic DTW algorithm is hardly feasible without additional measures to limit the search area (i.e., the number of visited states). The fact that only limited corrections are often sufficient to align both chromatographic and spectral data (cf. section 3.2.1) infers that it is possible to restrict the search area without consequences on the retrieval of the correct path.

The basic DTW algorithm may lead to severe artefacts that discourage its use for spectral and chromatographic data even when the dimensions of the problem would not be an obstacle^{124,233,PIV}. Again, more severe constraints must be imposed on \mathcal{F} than the basic continuity ones to yield acceptable results^{PIV}.

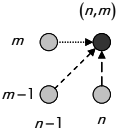
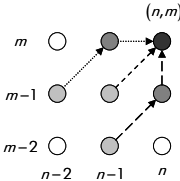
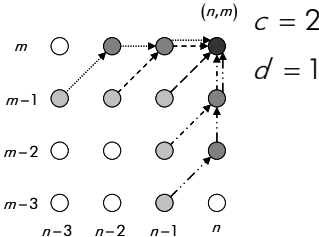
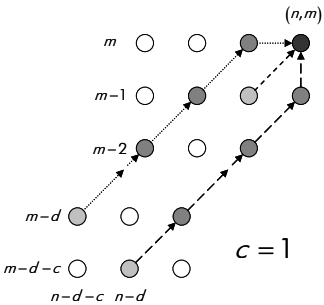
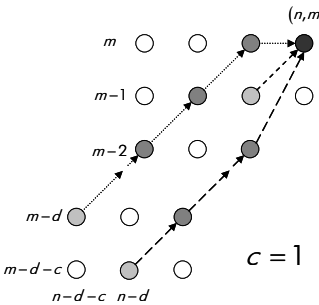
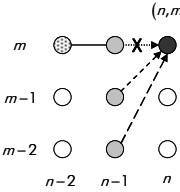
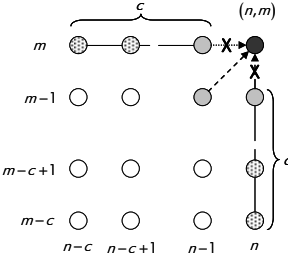
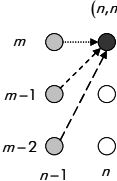
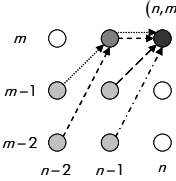
Constraints

Three types of constraints can be applied to DTW: local, global and endpoint constraints. Local constraints define the local behaviour of the warping path and correspond to a generalisation of the continuity constraints of problem (3.17), global constraints affect the area in which the warping path is sought and endpoint constraints fix the initial and final point in the warping path²⁰⁸.

Most local constraints can be described as sets of more or less complex policies ('rules'^{PIV}) that can be followed to arrive at (or, equivalently, to depart from) a given state in the search grid²⁰⁸. Such sets of rules are denoted by the letter \mathcal{T} and are here termed 'lookup tables'^{PIV}. More precisely, let $p_i \equiv (\Delta n_i, \Delta m_i)$ denote an elementary transition (i.e., a single arc) connecting consecutive points in the warping path, then a rule $t_h \in \mathcal{T}$ is an allowed sequence of ℓ_h elementary transitions and is written as $t_h \equiv p_1 \dots p_{\ell_h}$. For example, according to this definition, the basic local continuity constraints $0 \leq \Delta m, \Delta n \leq 1$ correspond to the lookup table: $\mathcal{T} \equiv \{(1,0), (1,1), (0,1)\}$ (cf. Table 3.2a).

Every fraction of the warping path \mathcal{F} must comply with one of the rules in the lookup table \mathcal{T} to which it is subject and a valid warping path must be writable as a sequence of rules of the lookup table^{208,259}. For example, starting from state (1,1) the optimal warping path of the example of Figure 3.11 can be expressed as: $t_2 t_1 t_2 t_3 \equiv (1,1)(1,0)(1,1)(0,1)$.

Table 3.2. Graphs of some of the constraints that have been used for DTW in chemistry. Different line styles denote the distinct rules: t_1 , ---- t_2 , --- t_3 , -.- t_4 and -.- t_5 . In f) and g), the crosses indicate that a transition is forbidden if the optimal path up to the predecessor is formed by 1 horizontal transition or c consecutive horizontal/vertical transitions, respectively.

Type	Graphs	
Basic	a) 	
Slope Constraints	Mild	b)  $c = d = 1$
		c)  $c = 2$ $d = 1$
	Rigid	d)  $c = 1$
		e)  $c = 1$
Itakura/Kassidas Constraints		f) 
		g) 
Asymmetric		h) 
		i) 

The most common family of local constraints requires that at most c consecutive horizontal or vertical transitions (i.e., $(1,0)$ and $(0,1)$ respectively) can occur and that these must be followed by a minimum of d diagonal transitions of the type $(1,1)$ ^{149,208,259}. Note that the second condition is an active restriction that excludes the presence of right angles (i.e., sequences like $(0,1)(1,0)$ or $(1,0)(0,1)$) in the warping path^{169,259}.

The net result of these constraints is that the slope of the warping path must lie in the interval $\left[d(c+d)^{-1}, d^{-1}(c+d)\right]$. Hence, they are referred to as ‘slope constraints’^{149,208,259,PIV}. For example, the condition $c = d = 1$ corresponds to the lookup table $\mathcal{T} \equiv \{(1,0)(1,1), (1,1), (0,1)(1,1)\}$ and imposes that the slope of the warping path must lie in

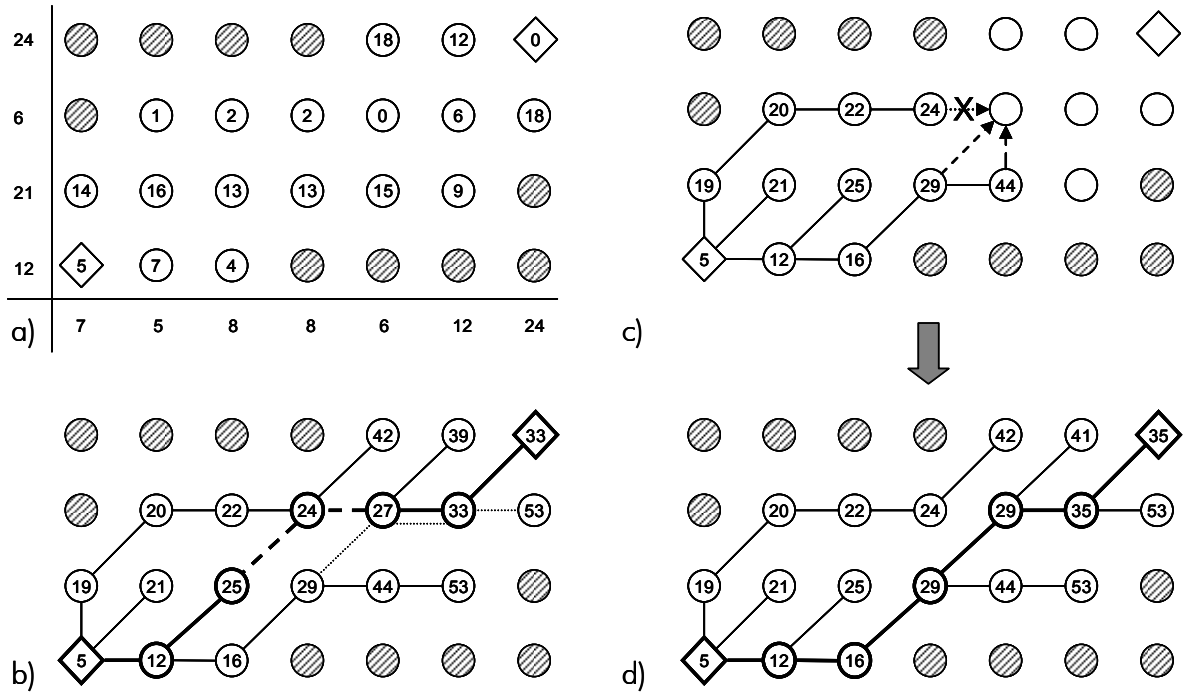


Figure 3.12. Itakura/Kassidas constraints are not equivalent to slope constraints and can yield a suboptimal warping path. Here it is shown for $c = 2$ and $d = 1$. a) Mapping grid; the two data vectors are reported at the bottom ($N = 7$) and on the left ($M = 4$) of the graph. The numbers in the circles are the (local) Euclidean distances and shadowed circles cannot be reached given the constraints. b) Warping path (bold) with standard slope constraints; the values in the circles are the optimal distance up to the corresponding state. — — and indicate the optimal local policies leading to states (5,3) and (7,3), respectively. c) Due to the way Itakura/Kassidas constraints are enforced, a policy that belongs to the optimal warping is rejected and a transition that might result in a right angle is permitted. The arrows denote the available policies in \mathcal{T} (cf. Table 4). d) A suboptimal path (bold line) is thus determined.

the range $[0.5, 2]$ (Table 3.1b).

Slope constraints have proven to be essential for the successful application of DTW to the alignment of chromatographic as well as batch process data prior to analysis with multilinear modelling^{149, PIV}. In particular, mild slope constraints in which $d' = 1$ and c is in the order of a few units (typically 1 to 3) (cf. Table 3.2c) have been found more appropriate to treat batch process data, whereas the treatment of chromatographic (and spectral) data benefits from more rigid constraints in which $d' \gg c$ (Table 3.2d)^{149, PIV}.

Extending slope constraints and DTW algorithms to the general case of arbitrary c and d' may yield rather complex lookup tables and recursive relationships in which the number of predecessors to look at for each stage grows larger and larger. Consequently, another formulation with seemingly equivalent constraints has been devised that is easier to implement and is likely to be faster because it only looks at the three immediate predecessors $(n-1, m)$, $(n, m-1)$ and $(n-1, m-1)$ of the current state (n, m) . Specifically, state $(n-1, m)$ is not considered an allowed predecessor of (n, m) if it has been reached through c horizontal transitions, and state $(n, m-1)$ is not an allowed predecessor of (n, m) if it has been reached through c vertical transitions¹⁴⁹ (Table 3.2g).

These constraints are an extension to the symmetric warping problem of those proposed by Itakura (Table 3.2f)¹³² and, unlike other slope constraints, cannot be represented as sequences of elementary transitions (i.e., they are not associated to any lookup table)²⁰⁸. Unfortunately, like Itakura's, these constraints do not necessarily yield the same optimal warping as the slope constraints they are expected to replace (Figure 3.12). First, because the transitions that are locally rejected may be parts of an optimal warping path that does not violate the conditions imposed by standard slope constraints, and second because they implicitly allow right angles in the warping path²⁰⁸.

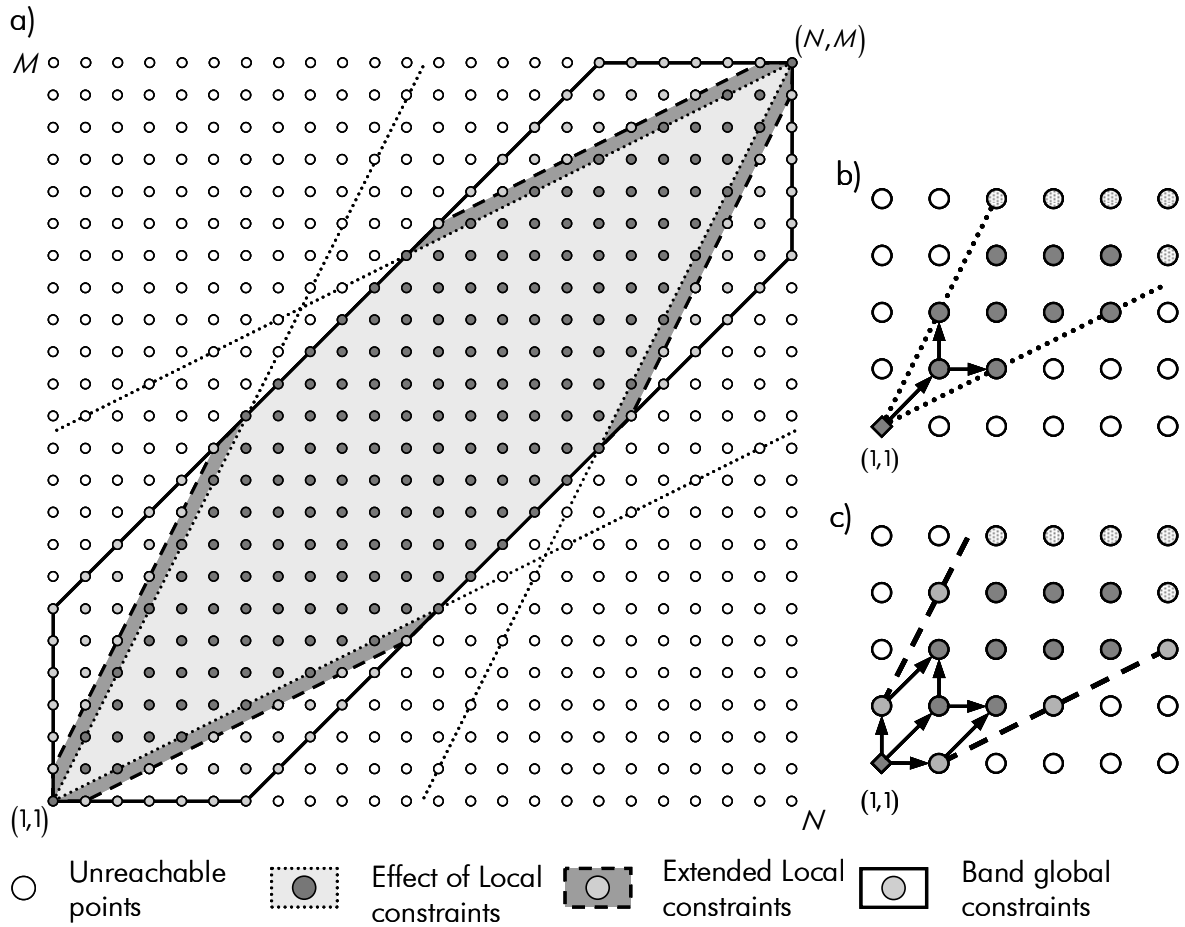


Figure 3.13. a) Effect of local and global constraints on the search area; the dotted lines have either maximum or minimum slope. b) Standard boundaries for search area with lookup table $\mathcal{T} \equiv \{(0,1)(1,1), (1,1), (1,0)(1,1)\}$. c) Extended boundaries taking into account the presence of horizontal and vertical transitions some of the rules of \mathcal{T} .

A family of asymmetric constraints that is particularly popular in the field of speech recognition and has occasionally been employed for chemical data imposes that the only admissible transitions have the form $(1, \Delta m)$ for $0 \leq \Delta m \leq c$ ²⁰⁸. The graph for $c = 2$ can be seen in Table 3.2h and the associated lookup table is $\mathcal{T} \equiv \{(1,0), (1,1), (1,2)\}$. Imposing such constraints implies that the reference time axis is the one along which the loss function is calculated and that the warping path is of the form $\mathcal{F} \equiv \{(n, m(n)) | n = 1, \dots, N\}$. Clearly, the sample time axis could have the same role, should transitions $(\Delta n, 1)$ for $0 \leq \Delta n \leq c$ be allowed²⁰⁸. Like in the symmetric case, slope constraints are often imposed on the warping path to avoid extreme behaviours. In this

case, this is done by requiring that c horizontal transitions $(1,0)$ must be followed by a diagonal one in which $1 \leq \Delta m \leq c$ (cf. Table 3.2i)^{132,149,208,259}.

There is, of course, the possibility of combining several types of constraints in the same lookup table (Table 3.2e). The choice depends on the data at hand and, once again, on the model for the warping path that one is using (cf. section 3.2.1). In this respect, it is important to understand how the warping path and the constraints to which it is subject affect the signal after the alignment^{233,PIV}. Since this effect depends on the synchronisation step, it is treated in detail in the corresponding section on page 99.

Because of slope and endpoint constraints, not all points within the search grid can be reached^{149,208}. By and large, for endpoints $(1,1)$ and (N,M) , the legal search area is the lozenge comprised between four lines passing from $(1,1)$ and (N,M) and having either the maximum or the minimum allowed slope (Figure 3.13a). Hence:

$$\begin{aligned} \max(\alpha_{\min}(n-1)+1, \alpha_{\max}(n-N)+M) &\leq m \\ &\leq \min(\alpha_{\max}(n-1)+1, \alpha_{\min}(n-N)+M) \quad \text{for } n=1, \dots, N \end{aligned} \quad (3.20)$$

where α_{\min} and α_{\max} are the minimum and maximum slope allowed by the local constraints in the (n,m) plane. However, the search area defined in this fashion would be stricter than necessary and would prevent a number of valid paths from being investigated if any of the rules in \mathcal{T} contained horizontal or vertical transitions. For example, a path that starts with a horizontal transition followed by a diagonal one would not violate any rule in the lookup table $\mathcal{T} \equiv \{(1,0)(1,1), (0,1)(1,1), (1,1)\}$ but would not be allowed using (3.20) because the intermediate point $(2,1)$ lies below the line $m = 0.5(n+1)$ (Figure 3.13b). In order to avoid this, the four lines delimiting the search area must pass at c_h and c_v points from the corresponding endpoint, where c_h and c_v denote respectively the maximum number of consecutive horizontal and vertical transitions allowed by any rule in the lookup table. Thus, equation (3.20) becomes:

$$\begin{aligned} \max(\alpha_{\min}(n+c_h-1)+1, \alpha_{\max}(n-N)+M-c_v) &\leq m \\ &\leq \min(\alpha_{\max}(n-1)+c_v+1, \alpha_{\min}(n-N+c_h)+M) \quad \text{for } n=1, \dots, N \end{aligned} \quad (3.21)$$

For example, using lookup table $\mathcal{T} \equiv \{(1,0)(1,1), (0,1)(1,1), (1,1)\}$, the boundaries of the allowed search area should pass through points $(2,1)$ and $(N-1, M)$ with slope 0.5, and points $(1,2)$ and $(N, M-1)$ with slope 2 (Figure 3.13c).

The search area can be more directly delimited using global constraints^{149,259}. For example, if one knows that the maximum shift observed in a data set is A points, it would be sensible to restrict the search area so that m does not differ from n by more than the same amount^{149,259}. Similarly, when M and N are different, it may be useful to require that the deviation from the linear path be limited to A points (i.e., $|m - MN^{-1}n| \leq A$ for $n = 1 \dots N$)^{227,PIV} (Figure 3.13a). Standard band constraints are often formulated so that the width of the band remains constant for all values of M and N ; however, for some data it may be preferable to allow this band to change. For example, the retention time shift in chromatographic data typically increases with the retention time and this could easily be implemented in the band constraints. Perhaps the most important effect of band constraints is to reduce the computational complexity from $O(MN)$ to $O(AN)$. Since A can be as small as 1 or 2% of M (or N), the advantage in this sense can be remarkable.

Another global constraint that has been found useful is to require that the warping path must pass through specific points in the mapping grid^{149,284}. This has been done, for example, for process monitoring when comparing autocorrelation patterns of two stochastic faults and can easily find application when the position of feature that is known to occur (e.g., an internal standard in chromatography) can be determined with great precision and is expected to appear at the same time in all signals. This also shows how DTW can be applied in the context of landmark registration. Most alignment algorithms based on landmark matching used in chemistry correct the signal under the assumption that the warping path is linear between any two landmarks^{148,333}. However, this is not necessarily the case in practice and DTW may be useful to provide a non-parametric nonlinear warping path that aligns the signal between landmarks²⁸⁴.

Note that, when the warping path is known to pass through a certain point, the warping problem is effectively split in two smaller ones. Whether or not they can also be solved separately depends on the presence and nature of local constraints. This point can be further clarified considering that, if the warping path is subject to slope constraints with

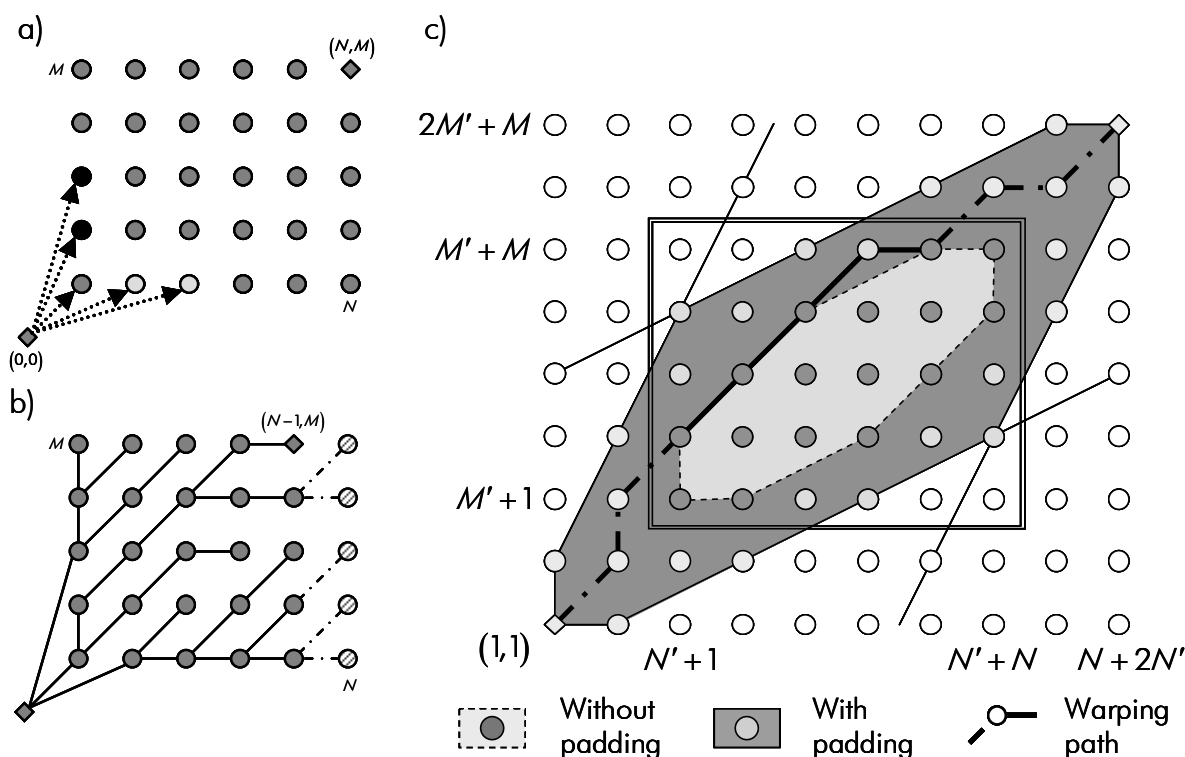


Figure 3.14 Endpoint constraints. a) adding initial state $(0,0)$ with $Q_{(0,0)}^* = 0$ to relax initial endpoint; b) all points on the top and right border are possible final endpoints and solution to a smaller warping problem when denominator does not depend on the warping path; c) attaching to the reference and signal sequences of constant values expands the search area. The rectangle in the centre delimits the standard mapping grid.

$c = d = 1$ and the first part terminates with a vertical or a horizontal transition, the second must start with a diagonal one. Finally, it is worth mentioning that DTW has been used, in combination with an additional level based on dynamic programming, to align process data when some landmarks are missing²⁸⁴. This way of operating the DTW resembles the algorithms for solving the “connected word recognition” problem^{209,213,258}.

With respect to endpoints, the standard choice is to impose that $(1,1)$ and (N,M) are the initial and final point^{149,239}. The combination of these constraints with the local constraints just described implies that the search area for the warping path is the lozenge depicted in Figure 3.13a and prevents relatively large corrections at the beginning and at the end of the signal. However, this limitation is often impractical for chemistry problems and the endpoints must be relaxed. For example, when aligning a spectrum, it makes little

sense to expect that the maximum shift is smaller (and in fact nearly absent when the local constraints are particularly rigid) at the beginning than in the rest of the signal. Likewise, the retention time shift is likely to be larger at the end than in the middle part of a chromatogram and the initial point of two chromatograms may not correspond because the analytical conditions have changed^{15,PIV-PVI}. Interestingly, the two endpoints are handled differently depending on whether DP is implemented in a forward or a backward manner. Only the application to the forward algorithm is explained here since it is more common for DTW and the backward one can be obtained straightforwardly.

Endpoint $(1,1)$ can be relaxed by including an artificial initial state $s_0 \equiv (0,0)$ whose value of the loss function is fixed to zero and by allowing only transitions of the type $(1,\Delta m)$ and $(\Delta n,1)$ with $1 \leq \Delta m \leq \delta_M$ and $1 \leq \Delta n \leq \delta_N$ to depart from it (Figure 3.14a). In this way, the first point in the reference is allowed to correspond to any of the first δ_M points in the sample, and the first point in the sample is allowed to correspond to the first δ_N in the reference. Moreover, using state s_0 is consistent with the generally imposed condition that $Q_{(1,1)}^* = 2q(x_1, y_1)$ for a choice of weighting function $w(k) = \Delta m(k) + \Delta n(k)$ ¹⁴⁹.

It is worth mentioning that, if one uses the weighting function $w(k) = \Delta n$, the optimal cumulated distance for the allowed endpoints on the left border on the mapping grid (in black in Figure 3.14a) are equal to the local distances between such points and the first point in the reference. The same observation was made by Kassidas¹⁴⁹ without making reference to state s_0 . Using this additional state, though, allows a straightforward treatment of the relaxation of the initial endpoint such that it is valid also for the points on the bottom border (in light grey in Figure 3.14a). This option seems more consistent than permitting $(1,\Delta m)$ transitions to relax the endpoint on the left border while allowing δ_N consecutive horizontal transitions to handle the relaxation at the bottom of the mapping grid¹⁴⁹. In fact, relaxing the slope constraints at the bottom and left borders and allowing up to δ_N and δ_M horizontal or vertical transitions from point $(1,1)$, respectively, represents an alternative to admitting $(1,\Delta m)$ and $(\Delta n,1)$ transitions from $(0,0)$ ¹⁴⁹. Unfortunately, both options are associated with artefacts that largely depend on the subsequent synchronisation step (see next section) and which one to choose depends mostly on the data.

With respect to the final point in the warping path, one should consider that the optimal value of the loss function and a warping path leading to state s_0 are obtained for any point that belongs to the search area. Therefore, the warping problem is solved for the same initial point at any of the boundary points on the right and top boundary of the mapping grid, if the warping path is allowed to reach them. In other words, the optimal value of the loss function for state $(N - n', M)$ is the optimal value for the warping problem between a sample of length M and a reference that has been truncated at length $N - n'$ (Figure 3.14b). Similarly, the value of Q^* at state $(N, M - m')$ is the solution to a warping problem in which the sample being aligned has been truncated to length $M - m'$. Thus, unlike the general case, in which the normalisation factor can be disregarded, normalisation is necessary to eliminate the effect of the length of the warping path and decide which is the best endpoint.

Finally, an empirical but effective way of relaxing the endpoint constraints is to append more or less long stretches of data points at the beginning or at the end of the sample and/or the reference (Figure 3.14c). This solution turned out to be quite useful in practice even though it increases the computational cost because it does not require any additional modification of the algorithm^{PV, PV}. In particular, if the last (or the first) point in the sample and/or the signal is replicated in the padded sections, the results will be almost equivalent¹ to relaxing the slope constraints at the border of the mapping grid and allowing a certain number of consecutive horizontal/vertical transitions at the end of the warping path.

Synchronisation

When alignment algorithms are used as a preprocessing step prior to the fitting of multilinear models, it is necessary that the resulting signals have the same length so that they can be stacked in a matrix or in a multi-way array. This is not the case with the data resulting from symmetric DTW, because the length of the warping path K depends on the

¹ Complete equivalence depends on the choice of the weighting scheme.

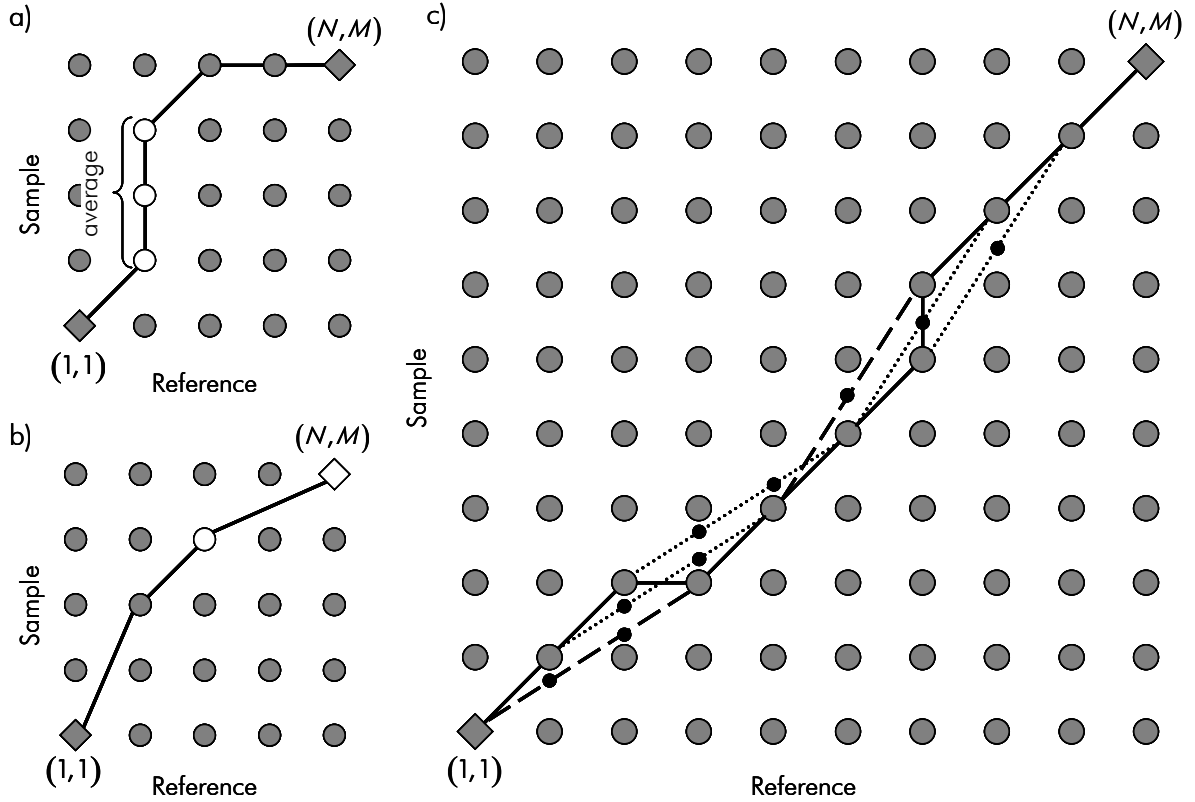


Figure 3.15 a) Averaging data points spanned by vertical transitions (open circles) yields a synchronised sample having the same length as the reference. b) If local constraints allow skipping data points along the reference time axis (e.g. between open circles) averaging is not sufficient for synchronisation and interpolation is required. c) Interpolation can be used to deal with all types of lookup tables. The solid line is the warping path, the dashed lines denote the optimal synchronisation path and the dotted lines are all the possible synchronisation paths. The black dots identify the locations along the sample of the points corresponding to the reference points spanned with segments of slope $\frac{3}{4}$ or $\frac{4}{3}$.

sample being aligned. Synchronisation is the procedure that is used to yield aligned samples of length identical to that of the reference^{149,233,241,PIV,PVI} and is by definition asymmetric in that it treats sample and reference differently.

For example, synchronisation is necessary when the same point in the reference corresponds to several points in the sample (i.e., when vertical transitions occur in the warping path). In this case, a straightforward and common way of synchronising a sample to a reference is to take the average of the points in the sample interested by the consecutive vertical transitions (Figure 3.15a). Unfortunately, this operation is also responsible for

most of the artefacts observed when basic or mildly constrained DTW is applied to chromatographic and spectral data^{233,PIV}. These range from merging smaller peaks that are absent in the reference into larger ones, to reducing the height of peaks that are taller in the sample than in the reference and creating plateaus at the top of peaks that are smaller in the sample than in the reference^{233,PIV}.

Another problem of averaging is that it does not work for all types of transitions. For example, \mathcal{T} may admit transitions that skip one or more points in the reference (e.g., (2,1) – Figure 3.15b), which would result in a shortened synchronised sample even after averaging (four points instead of five in Figure 3.15b). A more general option that also works in this case is to treat the points spanned by the optimal rule that constitute a fraction of the warping path as a segment and to use interpolation^{PIV,PVI}. For example, if the optimal rule spans 4 distinct points on the sample and 3 on the reference, one can interpolate the original 4 points into 3, thus removing the additional one (Figure 3.15c)^{PIV,PVI}. This solution has been used on chromatographic data and is preferable particularly because the smoothness of the original signals is maintained^{PIV,PVI}.

Interpolation and averaging operate in two different ways and lead to two distinct ways of aligning a signal. In essence, interpolation corresponds with local time scaling, whereas averaging (as well as asymmetric constraints of the type $(1, \Delta m)$ with $0 \leq \Delta m \leq c$) corresponds to insertion (i.e., the horizontal segments in the warping path) and deletion of time points. Figures 3.5d and 3.5e show a clear example of the difference that the two different approaches have on the quality of the final result.

In order to yield good results it is crucial that the warping path after the synchronisation be a good estimation of the “true” warping path. For example, the shifts observed for peaks 2 and 3 in Figure 3.5a do not correspond to a proportional change in width. Thus, it seems unlikely that the shift is determined by a local (different) timescale. This is confirmed in Figure 3.5e, in which interpolation allows the alignment of peak tops but results in an unrealistic widening of peak 3. It is certainly difficult to establish when deletion is preferable to scaling as it depends on the data. However, it is important to point out that Euclidean distance or other local distance measures that consider only single observations along the curve cannot give an indication on which is the best choice. For example, dele-

tion would allow removing entire peaks from the sample if they were missing in the reference since this would lead to a lower value of the loss function compared to scaling. In this respect, interpolation and rigid slope constraints appear as a working palliative rather than a cure and other local distance measures (e.g., Pearson's correlation coefficient, cross-correlation³⁴¹ or the residuals from a local modelling¹⁰⁶) may be more effective.

One of the aspects of synchronising using interpolation that requires further investigation is that the warping path is not found by minimising a loss function based on the interpolated sample (i.e., after synchronisation) as it is done for example in COW (cf. section 3.3.3). Hence, there is no guarantee that the aligned sample after synchronisation is the one that yields the minimum distance from the reference. For example, in the case of standard slope constraints, the fact that c consecutive vertical (or horizontal) transitions are followed by at least d diagonal ones implies that there are at least d possible sections in the warping path that contain only c consecutive vertical/horizontal transitions (Figure 3.15c) and can be used for interpolation. All of these must be checked in order to determine the one that yields the best alignment. In this sense, it is apparent that using only the optimal rule of the lookup table to define the points that are involved in the interpolation^{PV, PVI} may lead to suboptimal solutions. Since there may be some conflicts between the possible endpoints of adjacent segments (Figure 3.15c), an additional DP step could be used to find the optimal solution, also allowing to find the optimal choice between local time scalings or deletions/insertions. In such two-level algorithm, the first DP step, which operates on the original data points, would essentially restrict the search area for the second DP step which could also achieve a precision that exceeds that of the original signal. In this sense, there is some similarity with the method proposed by Salvador to speed up DTW²⁶¹.

3.3.3 Correlation Optimised Warping

Correlation Optimised Warping (COW) is a warping algorithm based on DP that finds a piece-wise linear warping path. The segments that compose the warping path span a fixed number of data points along one of the two time axes and are allowed to assume a limited number of distinct slopes. The optimisation criterion is the sum of the Pearson's

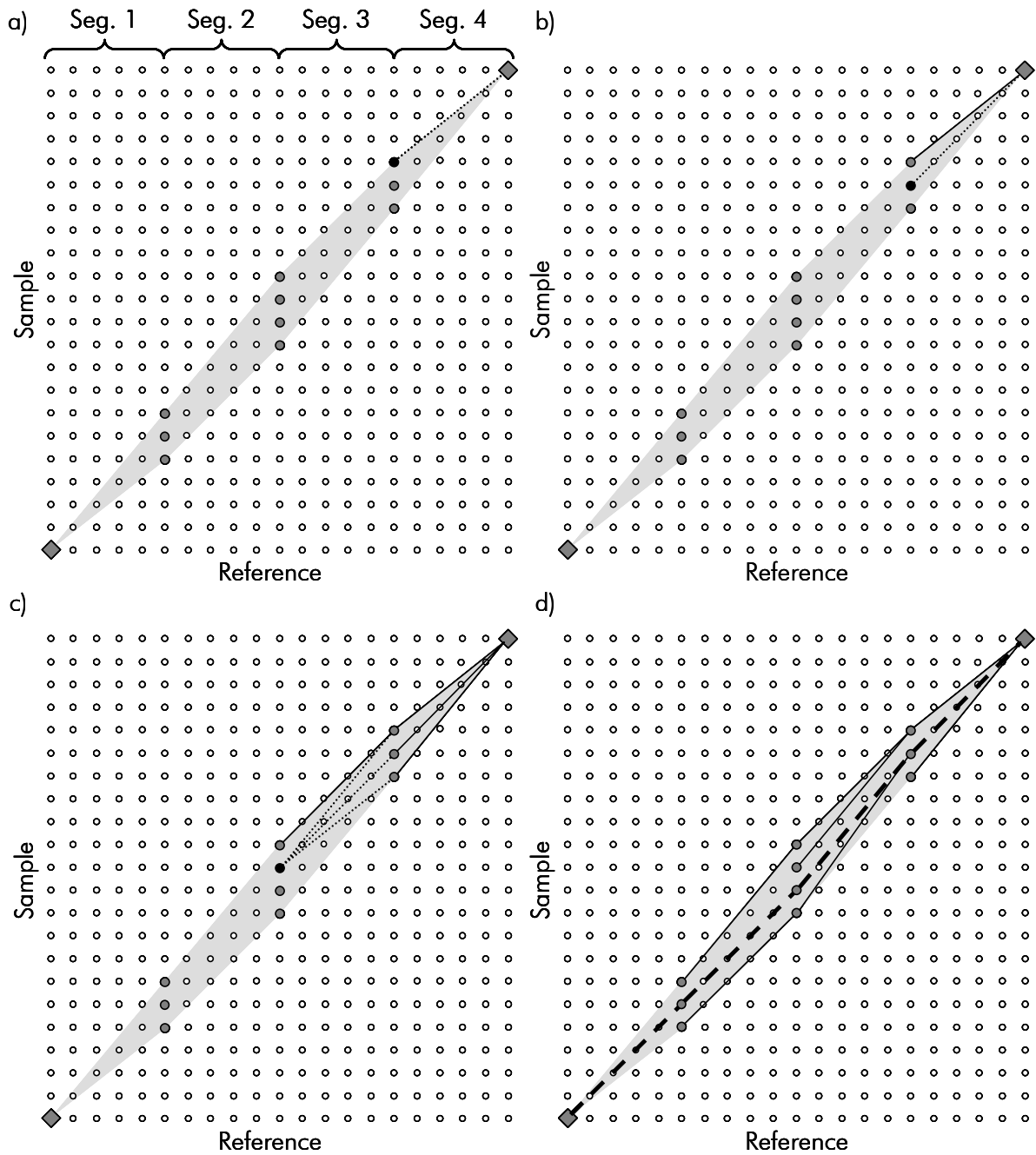


Figure 3.16. Graphical representation of the backward DP algorithm for COW. a-c) Dotted lines are the evaluated segments that reach the black point in the mapping grid. Solid lines are the segments that maximise the correlation for the relative sections of sample and reference. d) Once all the possible endpoints have been evaluated, the optimal warping path (dashed line) can be retrieved.

correlation coefficients between the interpolated sections of sample and reference spanned by the same segments of the warping path. This method was first presented in

1998²¹⁴ and has been used in chromatography both alone (for pattern recognition) and in combination with multilinear models^{49,61,124,136,233,308,309,PIV-V}.

COW operates on discrete signals by aligning a sample \mathbf{y} to a reference \mathbf{x} of length M and N respectively. Both reference and sample are split into K segments and the segment boundaries in either the reference or the sample have fixed positions. While these can be defined based on prior knowledge (e.g., they could be fixed at some landmark feature), in the original COW algorithm a more generic criterion is preferred to avoid feature recognition (namely, it is required that all segments have fixed length on the sample time axis)^{214,233,PIV,PV}. More in general, there are two different ways of implementing COW depending on whether the segment boundaries are fixed on the reference or on the sample time axis. In some literature and in the original publication^{136,214,215,233}, the latter option was used, whereas in this thesis and in some recent publications the former has been preferred^{61,PIV-PV}. There seems to be no particular reason to choose one or the other although the search areas for the warping path are different in the two cases. However, there has been little or no investigation in this direction.

In the basic COW algorithm, the first and last point of the reference and sample are forced to match, and the remaining $K - 1$ boundaries in the sample are the subject of the optimisation. However, the endpoint constraints can be relaxed similarly to the DTW algorithm (cf. page 97)^{49,214,PIV}. The length of the segments in the reference is fixed to ℓ_x time points. Conversely, the length of the segments in the sample is allowed to change and may assume all integer values in a fixed interval $\mathcal{I} = [\ell_y - t, \ell_y + t]$, where ℓ_y is the average length of the segments in the sample (i.e., $\ell_y = \lfloor M/K \rfloor$) and t is referred to as the slack parameter^{214,PIV}. t may be equal for all segments or change depending on the need and non-integer values can be picked in \mathcal{I} ^{49,214,PIV}.

The loss function is the sum over the segments of the Pearson's correlation coefficient ρ between corresponding segments in the sample and the reference. The sample segment is (linearly) interpolated to ℓ_x points if its length is different. The loss function is expressed as:

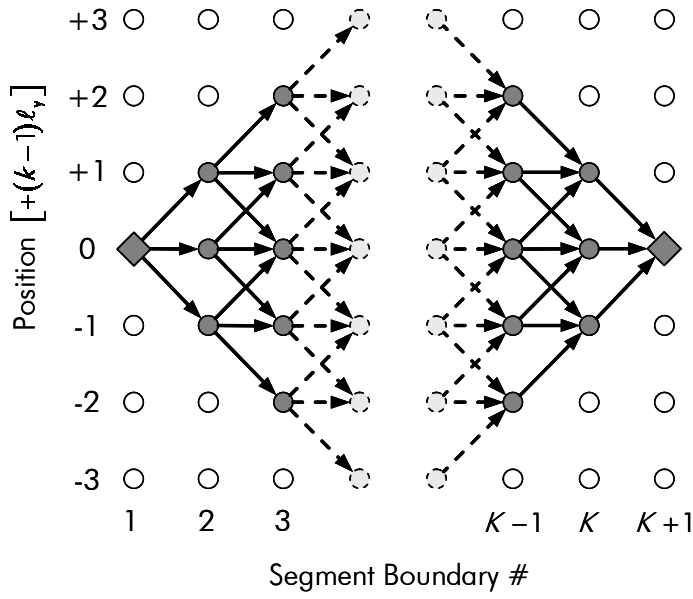


Figure 3.17. Graph of the COW as a shortest path problem. The slack is limited to 1; therefore there are three arrows departing from each point. The ‘cost’ of each policy is the correlation coefficient between the interpolated sample segment and the corresponding segment in the reference. The path is not allowed to pass through the empty circles.

$$Q^* = \max_{\mathbf{b}} \sum_{k=1}^K \rho(\mathbf{x}_k, \bar{\mathbf{y}}_{b_k:b_{k+1}}), \quad (3.22)$$

where \mathbf{b} is the vector of $K + 1$ boundaries for the segment in the sample whose k -th element is the position of the first point in the k -th segment, and $\bar{\mathbf{y}}_{b_k:b_{k+1}}$ denotes the vector obtained by interpolating the k -th segment in the sample to length ℓ_x . The elements of \mathbf{b} are strictly monotonically increasing with $b_1 = 0$ and $b_{K+1} = M$.

By comparing equation (3.22) with (3.15b), it is apparent that the problem of maximising Q can be solved using DP. The stages in the DP algorithm are then the segments, the states are the possible segment boundaries and the policy decisions are the segment lengths. Figure 3.16 visualises on a mapping grid the process of finding the optimal warping path according to COW when M and N are equal to 22 and 21 respectively and both ℓ_x and ℓ_y are equal to 5. The slack parameter t is set to 1. As can be seen, the segments need not pass through the points (i.e., integer values) in the mapping grid as is the case with DTW and the sample points at which the correlation coefficient is calculated are interpolated at the integer values of n in the interval spanned by the warping path segment on the reference time axis (Figure 3.15).

As (3.17), problem (3.22) can be considered a shortest path problem. Thus, other algorithms may be more efficient than DP if the number of states grows extremely

large^{20,125}. The graph corresponding to problem (3.22) can be seen in Figure 3.17 for $t=1$. Each circle corresponds to the position of a boundary and the arrows are associated with one of the possible segment lengths: ℓ_y and $\ell_y \pm 1$. This graph shows the storage requirements for COW, which are in the order of $O(tK^2)$, i.e., much smaller than $O(N^2)$. However, while memory consumption can hardly be considered a problem for COW, the computational load is heavily affected by the interpolation step. The time consumption can be reduced using band constraints similar to those described for DTW (cf. section 3.3.2)²¹⁴.

A final note regards some implementations, in which adjacent segments in both sample and reference do not share the boundary they have in common^{PIV}; instead, the last point in the k -th segment is next to the first point in the $(k+1)$ -th. This condition forces the warping path to have transitions of slope 1 between connecting the last point in one segment and the first point in the next^{PIV} and may induce some visible discontinuity in the warping path. However, even though imposing this additional condition is undesirable and not necessary, hardly any visible artefact or discontinuity has been observed when ℓ_x and ℓ_y are much larger than the slack (i.e., for rigid constraints)^{61,PIV-PIV}.

4. CONCLUSIONS AND PERSPECTIVES

In the course of this thesis two main subjects have been investigated: 1) the fitting of the PARAFAC model to multi-way arrays and 2) the correction of shift in chemical data by means of warping algorithms based on dynamic programming as a preprocessing step prior to the fitting of multilinear models.

Perhaps the most basic conclusion is the confirmation that the mathematical and computational aspects are fundamental in chemometric data analysis as to understand why certain methods do (or do not) yield the desired results on chemical data and how to improve the existing methods. In particular, it has become apparent that some algorithmic choices (e.g., SWATLD for PARAFAC and COW for warping) infer modifications of the model used to analyse the data that should be taken into account when evaluating the results or when comparing several methods. Therefore, a stricter collaboration between chemometricians, chemical and numerical analysts and experts in information science seems urgent for both PARAFAC fitting and warping algorithms as to improve the existing algorithms and avoid the spreading of (possibly harmful) misconceptions. One of the most ambitious objectives of this work has been to give a general framework for the two investigated subjects that, by making reference to both chemical and computational aspects,

allows both chemists/chemometricians and mathematicians/computer scientists to draw useful information from it.

A more detailed account on the specific results and some of the perspectives on PARAFAC and on warping methods is given in ensuing sections.

4.1 PARAFAC

With specific reference to the fitting of PARAFAC models, the following results are worth mentioning:

- PARAFAC-ALS performed well in tests based on real as well as simulated data sets and appears more stable with respect to over-factoring than it has recently been suggested^{P/}. However, other methods such as the Levenberg-Marquardt algorithm have been shown to be more efficient when the problem is characterised by high collinearity between factors^{P/,P//}, again in case of over-factoring^{P/} and when the fraction of missing values (if any) is relatively large or their pattern is particularly problematic^{P//}. Moreover, for higher order arrays and when the number of estimated parameters is sufficiently small, it has been shown how PARAFAC-ALS (without recurring to compression) is more expensive than Hessian based methods so long as the model rank is not particularly high^{P///}.
- The Levenberg-Marquardt method (PARAFAC-LM) appears to be more stable than PARAFAC-ALS with respect to collinearity and over-factoring^{P/}. The higher cost per iteration can be mitigated using compression^{P/}, but, even without it, PARAFAC-LM has been shown to be preferable to ALS for higher order arrays (for N larger than 4-5) and sufficiently small model ranks based on purely computational considerations^{P///}.
- While the least squares fitting criterion has well known statistical properties, other loss functions (for example yielding solutions with a higher value of core consistency^{P/}) may lead to faster algorithms and better predictions^{P/}. In particular, further research is necessary on SWATLD in order to establish the exact mathemati-

cal properties of its solution and the source of its relative stability^{P/}. On the contrary, other non-least squares algorithms (e.g., the iterative PARAFAC-PDE and ASD and the direct method GRAM/DTLD) appeared unsatisfactory.

- Compression based on Tucker models helps to reduce the computational cost of the fitting procedure so long as the original array is sufficiently large and can be beneficial in case of over-factoring for both ALS and the Levenberg-Marquardt algorithm^{P/}. Its application in combination with an expensive algorithm (e.g., Gauss-Newton) seems particularly useful so long as the rank of the model, which determines the size of the compressed array, is relatively small^{P/,P///}.

Interestingly, compression is compatible with equality constraints operating on the columns of the loading matrices, which are sometimes necessary to deal with rank overlaps and the corresponding lack of uniqueness. In this sense, compression in combination with, for example, PARAFAC-LM may provide a useful alternative to ALS for the so-called restricted Tucker models or for models that belong to the PARALIND family. Conversely, the problem of combining compression with other, more common, constraints like non-negativity and unimodality remains open.

Finally, it has been shown how the properties of the Khatri-Rao and of the Kronecker products can be used to extend compression in order to deal with a weighted least squares fitting criterion and maximum likelihood (with known error variance/covariance matrix) estimation. Whether these expressions are also associated to useful algorithms remains to be seen, especially given the difficulty of obtaining reliable estimates of variances in chemical data.

- The data one deals with in chemometrics often contain missing values and PARAFAC has been found to be more stable in this respect than bilinear models^{P//}. In particular, acceptable results have been obtained with up to 70% missing values for problems of $30 \times 30 \times 30$ and it was possible to correctly predict the concentration of several analytes in specimens containing up to five components^{P//}. Moreover, it has been shown that the presence of artefacts in the spectral

loadings does not necessarily correspond to worse predictions^{P/}. Thus, although less pleasant to the eye, the presence of such artefacts does not appear sufficient to reject a model or to justify the use of additional constraints.

The reason for the stability of PARAFAC with respect to missing values can intuitively be found in the number of elements in a data array, which is much larger than the number of parameters that needs to be estimated for most chemical problems. Combined with the rigid structure of PARAFAC, this infers that a large fraction of array elements can be safely set to missing without much influence on the estimates of the model parameters so long as their pattern does not "interfere" with the model structure^{P/}.

- The pattern of missing values has been found to have a much greater influence on the quality of the results and the speed of convergence of the fitting algorithms than their fraction. For example, the most common pattern of missing values in EEM fluorescence data has been shown to be responsible for particularly slow convergence and for the presence of artefacts in the loading vectors^{P/}. Furthermore, such a pattern has been shown to interfere with multilinearity allowing fictitious interactions between factors and enhancing the effect of slight deviations from strict N -linearity^{P/}. Since the multilinear nature of the components may require that some of the values be set to missing (or assigned zero weight in the loss function), the correct trade off has to be found between setting certain values to missing, to zero or any other "expected value" in order to yield the correct results. In general, this means that the pattern of missing values could be carefully designed in order to reduce the magnitude of the artefacts and the impact on the speed of convergence^{P/}.
- It has been shown how strict is the connection between the column-wise Khatri-Rao product and PARAFAC. The exploitation of the properties of the Khatri-Rao product leads to concise expressions of matrices (e.g., the Hessian) and vectors (like the gradient) associated with PARAFAC and to more efficient fitting algorithms^{P//}.

- The Jacobian matrix \mathbf{J} relative to the PARAFAC model is specially structured and very sparse^{PIII}. Its cross product $\mathbf{J}^T\mathbf{J}$ and its products with a vector $\mathbf{J}^T\mathbf{v}$ and $\mathbf{J}\mathbf{v}$ can be formed using only products of full matrices, with great savings in terms of time and memory consumption^{PIII}. The use of so-called ‘sub-expressions’ has been formalised in terms of the chain rule for vector functions and has been shown to be able to further reduce the computational load for the products $\mathbf{J}^T\mathbf{v}$ and $\mathbf{J}\mathbf{v}$. The corresponding advantage has been shown to grow with the ratio between the largest dimension in the data array and the rank of the fitted model. Moreover, it has been demonstrated how formulae identical to those derived using the chain-rule can be used in PARAFAC-ALS yielding a similar reduction in the computational cost^{PIII}.
- An efficient procedure for the solution of an exact line-search problem (in the MATLAB[®] environment) for PARAFAC has been devised based on the work by Franc⁹³. The algorithm has been extended to the general N -way case and its cost has been shown to increase exponentially with the order of the array. In this respect, when this thesis work had already been completed, it was found that, by exploiting the common sub-terms in the Khatri-Rao products involved in the direct procedure, its cost can be brought down to $(N-2)(N+1)$ Khatri-Rao products of two terms (equivalent to $F \sum_{m=1}^{N-2} [2(N-m)+3] \prod_{n=m+1}^N I_n$ operations) plus N loss function evaluations. The corresponding algorithm though has not been implemented yet and no information is currently available on its time expense.

4.2 Dynamic Programming and Shift

Two DP-based warping methods, namely DTW and COW, have been studied in detail as preprocessing steps prior to the fitting of multilinear models.

- It has been shown that COW and rigidly constrained DTW are compatible with the subsequent fitting of multilinear models, with examples in the context of ex-

ploratory data analysis on environmental^{PV} and food data^{PV} and for calibration problems^{PVI}.

- In the standard DTW algorithm, the warping path \mathcal{F} is expressed symmetrically and an asymmetric synchronisation step is necessary to obtain aligned signals having the same length. This step has been shown to be responsible for several artefacts that make basic DTW (i.e., without slope constraints) unsuitable for chromatographic data^{PV}. Mild slope constraints have been found unable to avoid such artefacts and thus also not appropriate to treat chromatographic data^{PV}. Artefacts also appear when mild asymmetric constraints are imposed in order to avoid the asymmetric synchronisation step. The nature of the artefacts and the reasons for their appearance in the aligned data indicate that both unconstrained and mildly constrained DTW (either symmetric or asymmetric) are also unfit for spectral data. These conclusions are consistent with the observations of other authors^{124,233}.
- Rigid slope constraints have been found necessary in order to properly align chromatographic (and spectral) data using either COW or DTW^{PV-PVI}. Moreover, it has been found that under rigid slope constraints both algorithms seek piece-wise linear warping paths and yield similar results^{PV}. The concept that the length of the segments should not be smaller than the width of the features one desires to align (e.g., peaks in chromatography), which was originally expressed only for COW²¹⁴, has been extended to DTW^{PV,PVI}.
- A simple method has been proposed to optimise the warping parameters whenever replicates of the same reference sample are available^{PV}. The results also indicate that, for specific types of data, the optimal length of the segments can largely exceed the width of the features that are to be aligned. The simplest explanation is that COW and rigidly constrained DTW produce piece-wise linear estimates of the "true" warping path and local slope ranges allowed by the constraints, together with the length of the segments, determine how close these estimates are to the correct paths. Thus, for GC data and signals relative to compounds with similar physical/chemical properties, the shift pattern is unlikely to be

particularly complex and long segments and small slope ranges for the single segments yield good estimates of the true warping paths^{PV,PV}. Other analytical methods (e.g., LC) may be characterised by more complex shift patterns that require more flexibility and thus shorter segments and larger slope ranges.

- The effect of global (band) and endpoint constraints on the warping path has been investigated in relation to the type of data that are affected by shift^{PV}. In particular, it has been determined that band constraints alone are insufficient to guarantee the absence of artefacts for chromatographic data, but determine a considerable reduction in the computational cost.

With respect to endpoint constraints, padding the signal with a leading and a trailing section repeating the first and last element in the signal (in DTW) or containing just noise (in COW) has been shown to be a simple empirical solution to the problem of relaxing the endpoints of the warping path^{PV,PV}. Although computationally more expensive, padding has the advantage of not requiring any major modification of the basic DTW and COW algorithms.

- The warping paths obtained using DTW and COW have been thoroughly characterised and a connection between them has been established^{PV}. It has been shown that the COW algorithm operates using solely time scaling and that, in fact, there are at least two distinct implementations of COW described in the literature in which the search area for the warping path changes^{214,PV}.

The characterisation of DTW in terms of local time scaling and time translation has been found to be more elusive, especially given the additional options offered by the asymmetric synchronisation step. In practice, using averaging only time translations are allowed and the alignment is achieved by "deletion" (i.e., when several data points in the sample signal are averaged into one) and "insertion" (i.e., the replication of the signal recorded at a specific time point). Alternatively, one can introduce interpolation as a more general means to perform asymmetric synchronisation, which results in allowing time scalings^{PV}.

An important conclusion is that neither COW nor DTW need to be restricted to using only time scalings or time translations and that one can straightforwardly implement both options and verify which warping path yields the best value of the loss function. Such an alignment algorithm has not been implemented yet, although it has already been proposed some years ago¹⁶⁹.

- Both DTW and COW are not landmark based and thus, almost by definition, cannot deal correctly with factors whose features overlap, but are differently shifted in each analysed signals (e.g., coeluting peaks in chromatography). The issue becomes more critical if some of these features are missing in either the sample or the reference. In this case, neither DTW nor COW has been found capable of properly removing shift, which infers that basic multilinear models like PARAFAC may have to be abandoned depending on the characteristics of the specific data set in favour of other models that can account for this behaviour like PARAFAC2^{PV}, MCR or shifted factors models.

In theory, at least for multi-way data, one could also devise a warping method that combines rank alignment with dynamic programming. However, the fact that the local rank is typically unknown is likely to be a hard problem to solve. Several attempts have been made to automatically find the rank of an N -way array (as required by rank alignment), but the results do not seem conclusive.

- With specific reference to the alignment of chromatographic data, neither COW nor DTW take into account the change in area under the peak, which is expected to be constant for equal concentrations of the substance in a sample. For example, a large peak may be matched with a more narrow one and its width changed accordingly. However, its height has to be changed as well in order not to underestimate the concentration and not to affect the factors of the fitted multilinear model. This has not represented a problem for the work conducted in this thesis because of the rigid constraints imposed to the warping paths, but it may have to be addressed when more flexible warping paths are required^{PV}. A solution that has been only marginally investigated is to maintain the area under the curve

in the warped signals constant for the parts corresponding to the segments that form the warping path.

- Dynamic Programming (DP) is an optimisation method that can be used in registration algorithms to find a warping path according to some fitting criterion and represents only one of several aspects that determine the usefulness of an alignment method. In this respect, the characteristics of the warping path (i.e., whether it is linear, piece-wise linear or nonlinear), whether or not landmarks are used to define it, and the criterion according to which the warping path is sought (e.g., the Euclidean distance or the Pearson's correlation coefficient) have been found as important as the choice of the optimisation algorithm.

Especially in the chemometric literature, there seems to be a little confusion between dynamic programming and the alignment methods in which this technique is used and the better results obtained with other algorithms are sometimes the result of aspects that prescind DP. Per contra, this distinction played an important role in the studies conducted in the course of this thesis on DTW and COW and seems even more relevant to improve the existing alignment methods.

- As a part of this work, a series of MATLAB routines to apply DTW and COW to chemical data have been produced and made available to the chemometrics community. Much work is left to do with respect to the computational efficiency of these routines, especially because this study was aimed mostly at determining the feasibility in terms of quality of the alignment rather than its cost. In this sense, any consideration about the effect of DP or its relative inefficiency based on MATLAB routines would be somewhat biased and not conclusive (especially considering that DP cannot be easily vectorised and would benefit from the use of compiled modules in this specific environment).

In terms of computational efficiency, it is noteworthy that many registration algorithms (and thus COW and DTW) based on DP solve shortest path problems and that specially designed algorithms exist to solve these types of problems¹²⁵. Their use in the context of alignment methods seems at least worth investigating.

- The results also indicate that the Euclidean distance is unsatisfactory as a local distance choice because it focuses solely on the intensity (or rather the difference in intensity) in the two signals interested by warping and do not account for their shape. Although rigid slope constraints mitigate this effect^{PV,PV'}, other choices must be studied that define the local distance (or more appropriately the cost of passing through a certain state in the optimal path) using the information contained in several time points like the cross-correlation using the Fourier transform. Alternatively, one could envision an algorithm using as a local distance measure the value of the residuals obtained after local modelling of shift^{106,196,338}. In this respect, DP could represent an improvement over methods that are not guaranteed to yield a minimum of the loss function (e.g., genetic algorithms). Another interesting advantage of this possibility is that local modelling may lead to better final results as it could deal with fractional shifts and moderate changes in shape, which represent additional sources of variations that need to be accounted for by multilinear models^{PV}.

Reference List

1. Empower 5.0, Waters Corporation (2002) www.waters.com
2. MATLAB 6.5, The Mathworks Inc. (2002) www.themathworks.com
3. J. Aach, G. M. Church, *Bioinformatics* **17**, 495 (2001).
4. K. M. Åberg, R. J. O. Torgrip, S. P. Jacobsson, *J. Chemom.* **18**, 465 (2004).
5. A. H. Andersen, W. S. Rayens, *Neuroimage* **22**, 728 (2004).
6. C. M. Andersen, R. Bro, *J. Chemom.* **17**, 200 (2003).
7. F. O. Andersson, R. Kaiser, S. P. Jacobsson, *J. Pharm. Biomed. Anal.* **34**, 531 (2004).
8. G. G. Andersson, B. K. Dable, K. S. Booksh, *Chemometr. Intell. Lab. Syst.* **49**, 195 (1999).
9. R. Andersson, M. D. Hamalainen, *Chemometr. Intell. Lab. Syst.* **22**, 49 (1994).
10. C. J. Appellof, E. R. Davidson, *Anal. Chem.* **53**, 2053 (1981).
11. C. J. Appellof, E. R. Davidson, *Anal. Chim. Acta* **146**, 9 (1983).
12. J. A. Arancibia, G. M. Escandar, *Talanta* **60**, 1113 (2003).
13. J. A. Arancibia, A. C. Olivieri, G. M. Escandar, *Anal. Bioanal. Chem.* **374**, 451 (2002).
14. T. J. Bahowick, R. E. Synovec, *Anal. Chem.* **67**, 631 (1995).
15. M. Bassompierre, G. Tomasi, L. Munck, R. Bro, S. B. Engelsen, *Chemosphere* Accepted for publication, (2006).

16. D. Baunsgaard, Analysis of color impurities in sugar processing using fluorescence spectroscopy and chemometrics., Ph.D. dissertation, The Royal Veterinary and Agricultural University, Frederiksberg, Denmark (2000). Available at <http://www.models.kvl.dk/research/theses/>.
17. R. Bellman, Dynamic Programming (Princeton University Press, Princeton, New Jersey, USA, 1957).
18. J. L. Beltran, J. Guiteras, R. Ferrer, *Anal. Chem.* **70**, 1949 (1998).
19. J. L. Beltran, J. Guiteras, R. Ferrer, *J. Chromatogr. A* **802**, 263 (1998).
20. D. P. Bertsekas, Dynamic Programming and Optimal Control (Athena Scientific, Nashua, NH, USA, 2nd edition, 2001).
21. C. Bessant, R. G. Brereton, S. Dunkerley, *Analyst* **124**, 1733 (1999).
22. S. Bijlsma, *Estimating Rate Constants of Chemical Reactions using Spectroscopy*, Ph.D. dissertation, University of Amsterdam (2000). Available at <http://three-mode.leidenuniv.nl/index.html>.
23. Å. Björck, *Numerical Methods for Least Squares Problems* (Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, first edition, 1996).
24. P. E. Black, "Breadth-first search" from *Dictionary of Algorithms and Data Structures* (National Institute of Standards and Technology, 2005). Available at <http://www.nist.gov/dads/HTML/breadthfirst.html>.
25. K. Booksh, J. M. Henshaw, L. W. Burgess, B. R. Kowalski, *J. Chemom.* **9**, 263 (1995).
26. K. S. Booksh, B. R. Kowalski, *Anal. Chem.* **66**, A782 (1994).
27. K. S. Booksh, A. R. Muroski, M. L. Myrick, *Anal. Chem.* **68**, 3539 (1996).
28. K. S. Booksh, C. M. Stellman, W. C. Bell, M. L. Myrick, *Appl. Spectrosc.* **50**, 139 (1996).
29. N. V. Boulgouris, K. N. Plataniotis, D. Hatzinakos, "Gait recognition using dynamic time warping", proceedings of the IEEE 6th Workshop on Multimedia Signal Processing (2004), pp. 263-266
30. A. R. Brenner, K. Eck, G. Engelhard, T. G. Noll, "Phase aberration correction using dynamic time warping" Proceedings IEEE Ultrasonics Symposium 1995 (1995), pp. 1361-1365
31. R. Bro, *Multi-way Analysis in the Food Industry. Models, Algorithms, and Applications*, Ph.D. dissertation, University of Amsterdam (1998). Available at <http://www.models.kvl.dk/research/theses/>.
32. R. Bro, *Chemometr. Intell. Lab. Syst.* **38**, 149 (1997).
33. R. Bro, C. A. Andersson, *Chemometr. Intell. Lab. Syst.* **42**, 105 (1998).

34. R. Bro, C. A. Andersson, H. A. L. Kiers, *J. Chemom.* **13**, 295 (1999).
35. R. Bro, S. DeJong, *J. Chemom.* **11**, 393 (1997).
36. R. Bro, R. A. Harshman, N. D. Sidiropoulos, "New rank-deficient models for multi-way data", Submitted to *J. Chemom.* (2006).
37. R. Bro, H. Heimdal, *Chemometr. Intell. Lab. Syst.* **34**, 85 (1996).
38. R. Bro, M. Jakobsen, *J. Chemom.* **16**, 294 (2002).
39. R. Bro, H. A. L. Kiers, *J. Chemom.* **17**, 274 (2003).
40. R. Bro, N. D. Sidiropoulos, *J. Chemom.* **12**, 223 (1998).
41. R. Bro, N. D. Sidiropoulos, A. K. Smilde, *J. Chemom.* **16**, 387 (2002).
42. R. Bro, A. K. Smilde, *J. Chemom.* **17**, 16 (2003).
43. R. Bro, A. K. Smilde, S. de Jong, *Chemometrics Intell. Lab. Syst.* **58**, 3 (2001).
44. M. K. Brown, L. Rabiner, *IEEE Trans. ASSP* **30**, 535 (1982).
45. T. R. Brown, R. Stoyanova, *J. Magn. Reson. Ser. B* **112**, 32 (1996).
46. C. Bugli, P. Lambert, G. Bigot, "Curve registration using fractional polynomials with application to electroencephalograms analysis", Technical Report 0524 (Université catholique de Louvain, Bruxelles, Belgium, 2004).
47. D. S. Burdick, X. M. Tu, L. B. McGown, D. W. Millican, *J. Chemom.* **4**, 15 (1990).
48. D. Bylund, *Chemometric Tools for Enhanced Performance in Liquid Chromatography-Mass Spectrometry*, Ph.D. Dissertation Acta Universitatis Upsaliensis (2001). Available at <http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-1091>.
49. D. Bylund, R. Danielsson, G. Malmquist, K. E. Markides, *J. Chromatogr. A* **961**, 237 (2002).
50. Y. Z. Cao, Z. P. Chen, C. Y. Mo, H. L. Wu, R. Q. Yu, *Analyst* **125**, 2303 (2000).
51. J. D. Carroll, J. J. Chang, *Psychometrika* **35**, 283 (1970).
52. J. D. Carroll, G. de Soete, S. Pruzansky, in *Multiway data analysis*, R. Coppi, S. Bolasco, Eds. (Elsevier, Amsterdam, The Netherlands, 1989), pp. 463-472.
53. J. D. Carroll, S. Pruzansky, J. B. Kruskal, *Psychometrika* **45**, 3 (1980).
54. S. Casarotto, A. M. Bianchi, S. Cerutti, G. A. Chiarenza, *IEEE Eng. Med. Biol. Mag.* **24**, 68 (2005).
55. R. B. Cattell, *Psychometrika* **9**, 267 (1944).
56. Z. P. Chen, Y. Li, R. Q. Yu, *J. Chemom.* **15**, 149 (2001).
57. Z. P. Chen, H. L. Wu, J. H. Jiang, Y. Li, R. Q. Yu, *Chemometr. Intell. Lab. Syst.* **52**, 75 (2000).

58. Z. P. Chen, R. Q. Yu, *Anal. Chim. Acta* **487**, 171 (2003).
59. D. D. Chilcote, C. D. Scott, *Anal. Chem.* **45**, 721 (1973).
60. J. Christensen, E. M. Becker, C. S. Frederiksen, *Chemometrics Intell. Lab. Syst.* **75**, 201 (2004).
61. J. H. Christensen, *Chemometrics as a tool to analyse complex chemical mixtures*, Ph.D. Dissertation Roskilde University, Roskilde, Denmark (2005).
62. J. H. Christensen, A. B. Hansen, G. Tomasi, J. Mortensen, O. Andersen, *Environ. Sci. Technol.* **38**, 2912 (2004).
63. J. H. Christensen, J. Mortensen, A. B. Hansen, O. Andersen, *J. Chromatogr. A* **1162**, 113 (2005).
64. W. F. Christensen, Y. Amemiya, *Math. Geol.* **33**, 801 (2001).
65. E. Comas *et al.*, *J. Chromatogr. A* **1035**, 195 (2004).
66. D. J. Crockford, H. C. Keun, L. M. Smith, E. Holmes, J. K. Nicholson, *Anal. Chem.* **77**, 4556 (2005).
67. A. de Juan, R. Tauler, *J. Chemom.* **15**, 749 (2001).
68. A. de Juan, R. Tauler, *Anal. Chim. Acta* **500**, 195 (2003).
69. L. De Lathauwer, B. De Moor, J. Vandewalle, *SIAM J. Matrix Anal. Appl.* **21**, 1253 (2000).
70. J. Deleeuw, F. W. Young, Y. Takane, *Psychometrika* **41**, 471 (1976).
71. A. P. Dempster, N. M. Laird, D. B. Rubin, *J. R. Stat. Soc. Ser. B Stat. Methodol.* **39**, 1 (1977).
72. S. B. Dolins, J. D. Reese, *IEEE Trans. Ind. Appl.* **28**, 261 (1992).
73. A. L. Duran, J. Yang, L. J. Wang, L. W. Sumner, *Bioinformatics* **19**, 2283 (2003).
74. S. R. Durell, C. H. Lee, R. T. Ross, E. L. Gross, *Arch. Biochem. Biophys.* **278**, 148 (1990).
75. B. Efron, R. J. Tibshirani, *An Introduction to the Bootstrap* (Chapman & Hall, New York, NY, USA, 1993).
76. P. H. C. Eilers, *Anal. Chem.* **76**, 404 (2004).
77. S. B. Engelsens, R. Bro, *J. Magn. Reson.* **163**, 192 (2003).
78. G. M. Escandar, D. G. Gomez, A. E. Mansilla, A. M. de la Pena, H. C. Goicoechea, *Anal. Chim. Acta* **506**, 161 (2004).
79. A. Espinosa-Mansilla, A. M. de la Pena, T. C. Goicoechea, A. C. Olivieri, *Appl. Spectrosc.* **58**, 83 (2004).
80. K. Faber, A. Lorber, B. R. Kowalski, *J. Chemom.* **11**, 419 (1997).

81. N. M. Faber, *Anal. Bioanal. Chem.* **372**, 683 (2002).
82. N. M. Faber, R. Bro, P. K. Hopke, *Chemometr. Intell. Lab. Syst.* **65**, 119 (2003).
83. N. M. Faber, L. M. C. Buydens, G. Kateman, *J. Chemom.* **8**, 147 (1994).
84. N. M. Faber, L. M. C. Buydens, G. Kateman, *J. Chemom.* **8**, 181 (1994).
85. N. M. Faber, L. M. C. Buydens, G. Kateman, *J. Chemom.* **8**, 273 (1994).
86. A. S. Field, D. Graupe, *Brain Topogr.* **3**, 407 (1991).
87. J. Forshed, I. Schuppe-Koistinen, S. P. Jacobsson, *Anal. Chim. Acta* **487**, 189 (2003).
88. J. Forshed *et al.*, *J. Pharm. Biomed. Anal.* **38**, 824 (2005).
89. C. G. Fraga, *J. Chromatogr. A* **1019**, 31 (2003).
90. C. G. Fraga, B. J. Prazen, R. E. Synovec, *Anal. Chem.* **73**, 5833 (2001).
91. C. G. Fraga, B. J. Prazen, R. E. Synovec, *Anal. Chem.* **72**, 4154 (2000).
92. C. G. Fraga, B. J. Prazen, R. E. Synovec, *HRC, J. High Resolut. Chromatogr.* **23**, 215 (2000).
93. A. Franc, *Etude algebrique des multitableaux: apports de l'algebrique tensorielle*, Ph.D. dissertation, Université Montpellier II (1992). Available at <http://three-mode.leidenuniv.nl/index.html>.
94. P. E. Frandsen, K. Jonasson, H. B. Nielsen, O. Tingleff, *Unconstrained Optimization* (Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 3rd edition, 2004).
95. S. J. Gaffney, *Probabilistic Curve-Aligned Clustering and Prediction with Regression Mixture Models*, Ph.D. Dissertation, University of California, Irvine (2004). Available at <http://www.ics.uci.edu/~sgaffney/>.
96. I. Garcia, *Modelos de Calibración de Tres Vías y Análisis de Residuos Según la Decisión 2002/657/CE de la Unión Europea*, Ph.D. Dissertation Universidad de Burgos, Spain (2005).
97. I. Garcia, L. Sarabia, M. C. Ortiz, J. M. Aldama, *Anal. Chim. Acta* **515**, 55 (2004).
98. S. García-Muñoz, T. Kourti, J. F. MacGregor, A. G. Mateos, G. Murphy, *Ind. Eng. Chem. Res.* **42**, 3592 (2003).
99. T. Gasser, A. Kneip, A. Binding, A. Prader, L. Molinari, *Ann. Hum. Biol.* **18**, 187 (1991).
100. P. J. Gemperline, *Anal. Chem.* **71**, 5398 (1999).
101. D. Gervini, T. Gasser, *J. R. Stat. Soc. Ser. B Stat. Methodol.* **66**, 959 (2004).

102. P. E. Gill, W. Murray, M. H. Wright, *Practical Optimization* (Academic Press, London, 1986).
103. C. A. Glasbey, K. V. Mardia, *J. Appl. Stat.* **25**, 155 (1998).
104. K. Gollmer, C. Posten, *Control Eng. Practice* **4**, 1287 (1996).
105. G. H. Golub, C. F. Van Loan, *Matrix Computations* (The Johns Hopkins University Press, Baltimore & London, 3 edition, 1996).
106. Gorenstein, M. V., "Method and devices for chromatographic pattern analysis employing chromatographic variability characterization", USA Pat. No 5969228 (1999)
107. B. Grung, R. Manne, *Chemometrics Intell. Lab. Syst.* **42**, 125 (1998).
108. B. Grung, O. M. Kvalheim, *Anal. Chim. Acta* **304**, 57 (1995).
109. C. Guillo, D. Barlow, D. Perrett, M. Hanna-Brown, *J. Chromatogr. A* **1027**, 203 (2004).
110. S. P. Gurden, J. A. Westerhuis, S. Bijlsma, A. K. Smilde, *J. Chemom.* **15**, 101 (2001).
111. M. V. Guterres, P. O. L. Volpe, M. M. C. Ferreira, *Appl. Spectrosc.* **58**, 54 (2004).
112. A. Gutmanas, P. Jarvoll, V. Y. Orekhov, M. Billeter, *J. Biomol. NMR* **24**, 191 (2002).
113. R. A. Harshman, in *Research methods for multimode data analysis*, H. G. Law, C. W. Snyder, J. A. Hattie, R. P. McDonald, Eds. (Praeger, New York, NY, USA, 1984), chap. Appendix A, pp. 566-591.
114. R. A. Harshman, *UCLA Working Papers in Phonetics* **16**, 1 (1970).
115. R. A. Harshman, *UCLA Working Papers in Phonetics* **22**, 30 (1972).
116. R. A. Harshman, *UCLA Working Papers in Phonetics* **22**, 111 (1972).
117. R. A. Harshman, W. S. DeSarbo, in *Research methods for multimode data analysis*, H. G. Law, C. W. Snyder, J. A. Hattie, R. P. McDonald, Eds. (Praeger, New York, NY, USA, 1984), chap. Appendix C, pp. 602-642.
118. R. A. Harshman, S. J. Hong, M. E. Lundy, *J. Chemom.* **17**, 363 (2003).
119. R. A. Harshman, M. E. Lundy, *Comput. Stat. Data Anal.* **18**, 39 (1994).
120. R. A. Harshman, M. E. Lundy, in *Research methods for multimode data analysis*, H. G. Law, C. W. Snyder, J. A. Hattie, R. P. McDonald, Eds. (Praeger, New York, NY, USA, 1984), chap. 6, pp. 216-284.
121. R. A. Harshman, M. E. Lundy, in *Research methods for multimode data analysis*, H. G. Law, C. W. Snyder, J. A. Hattie, R. P. McDonald, Eds. (Praeger, New York, NY, USA, 1984), chap. 5, pp. 122-215.

122. C. Hayashi, F. Hayashi, *Behaviormetrika* **49** (1982).
123. H. Heimdal, R. Bro, L. M. Larsen, L. Poll, *J. Agric. Food Chem.* **45**, 2399 (1997).
124. M. M. W. B. Hendriks, L. Cruz-Juarez, D. D. Bont, R. Hall, *Anal. Chim. Acta* **545**, 53 (2005).
125. F. S. Hillier, G. J. Lieberman, *Introduction to operations research* (McGrawHill, New York, NY, USA, 7th edition, 2001).
126. P. Hindmarch, K. Kavianpour, R. G. Brereton, *Analyst* **122**, 871 (1997).
127. S. J. Hong, R. A. Harshman, *J. Chemom.* **17**, 379 (2003).
128. S. J. Hong, R. A. Harshman, *J. Chemom.* **17**, 389 (2003).
129. P. K. Hopke, P. Paatero, H. Jia, R. T. Ross, R. A. Harshman, *Chemometr. Intell. Lab. Syst.* **43**, 25 (1998).
130. B. K. P. Horn, B. G. Schunck, *Artif. Intell.* **17**, 185 (1981).
131. I. Ibraghimov, *Numer. Linear Algebr. Appl.* **9**, 551 (2002).
132. F. Itakura, *IEEE Trans. ASSP* **A23**, 67 (1975).
133. M. Jalali-Heravi, M. Vosough, *Anal. Chim. Acta* **537**, 89 (2005).
134. G. M. James, "Moments Based Functional Synchronization", downloaded at <http://www-rcf.usc.edu/~gareth/research/camp.pdf> (2005).
135. G. M. James, "Curve Alignment by Moments", downloaded at <http://www-rcf.usc.edu/~gareth/research/curvesync.pdf> (2005).
136. E. Jarde, F. Vilmin, L. Mansuy, P. Faure, *J. Anal. Appl. Pyrolysis* **71**, 553 (2004).
137. N. Jeffries, *Bioinformatics* **21**, 3066 (2005).
138. J. H. Jiang, H. L. Wu, Y. Li, R. Q. Yu, *J. Chemom.* **13**, 557 (1999).
139. J. H. Jiang, H. L. Wu, Y. Li, R. Q. Yu, *J. Chemom.* **14**, 15 (2000).
140. R. D. Jiji, G. G. Andersson, K. S. Booksh, *J. Chemom.* **14**, 171 (2000).
141. R. D. Jiji, K. S. Booksh, *Anal. Chem.* **72**, 718 (2000).
142. K. J. Johnson, B. W. Wright, K. H. Jarman, R. E. Synovec, *J. Chromatogr. A* **996**, 141 (2003).
143. I. T. Jolliffe, *Principal Components Analysis* (Springer-Verlag, New York, NY, USA, 2nd edition, 2002).
144. P. Jonsson *et al.*, *Anal. Chem.* **76**, 1738 (2004).
145. P. Jørgensen, J. G. Pedersen, E. P. Jensen, K. H. Esbensen, *J. Chemom.* **18**, 81 (2004).
146. B. H. Juang, C. Tsuhan, *Signal Processing Magazine, IEEE* **15**, 24 (1998).

147. K. Kaczmarek, B. Walczak, S. de Jong, B. G. Vandenginste, *J. Chem. Inf. Comput. Sci.* **43**, 978 (2003).
148. N. Kaistha, C. F. Moore, *Ind. Eng. Chem. Res.* **40**, 252 (2001).
149. A. Kassidas, *Fault detection and diagnosis in dynamic multivariable chemical processes using speech recognition methods*, Ph.D. Mc Master University, Hamilton, Ontario, Canada (1997).
150. A. Kassidas, J. F. MacGregor, P. A. Taylor, *Aiche J.* **44**, 864 (1998).
151. A. Kassidas, P. A. Taylor, J. F. MacGregor, *J. Process Control* **8**, 381 (1998).
152. E. J. Keogh, M. Pazzani, "Derivative Dynamic Time Warping", paper presented at the First SIAM International Conference on Data Mining (Chicago, IL, USA, 2001) Available at <http://www.cs.ucr.edu/~eamonn>.
153. E. J. Keogh, M. J. Pazzani, *Principles of Data Mining and Knowledge Discovery* **1704**, 1 (1999).
154. H. A. L. Kiers, *J. Chemom.* **14**, 105 (2000).
155. H. A. L. Kiers, *J. Chemom.* **12**, 155 (1998).
156. H. A. L. Kiers, *Psychometrika* **62**, 251 (1997).
157. H. A. L. Kiers, A. K. Smilde, *J. Chemom.* **12**, 125 (1998).
158. H. A. L. Kiers, A. K. Smilde, *J. Chemom.* **9**, 179 (1995).
159. H. A. L. Kiers, J. M. F. Ten Berge, R. Bro, *J. Chemom.* **13**, 275 (1999).
160. A. Kneip, T. Gasser, *Annals of Statistics* **20**, 1266 (1992).
161. A. Kneip, X. Li, K. B. McGibbon, J. O. Ramsay, *The Canadian Journal of Statistics* **28**, (2000).
162. T. G. Kolda, *Limited-Memory Matrix Methods with Applications*, University of Maryland (1997). Available at <http://csmr.ca.sandia.gov/~tgkolda/pubs/>.
163. W. P. Krijnen, *The analysis of three-way arrays by constrained PARAFAC methods*, University of Groningen, The Netherlands (1993). Available at <http://three-mode.leidenuniv.nl/index.html>.
164. P. M. Kroonenberg, *Stat. Appl.* **4**, 619 (1992).
165. J. B. Kruskal, *Linear Alg. Appl.* **18**, 95 (1977).
166. J. B. Kruskal, in *Multiway Data Analysis*, R. Coppi, S. Bolasco, Eds. (Elsevier, Amsterdam, 1989), pp. 8-18.
167. J. B. Kruskal, in *Time warps, string edits and macromolecules: the theory and practice of sequence comparison*, D. Sankoff, J. B. Kruskal, Eds. (Center for the Study of Language and Information Publications, Leland Stanford University, CA, USA, 1999), chap. 1, pp. 1-44.

168. J. B. Kruskal, R. A. Harshman, M. E. Lundy, in *Multiway data analysis*, R. Coppi, S. Bolasco, Eds. (Elsevier Science Publishers, Amsterdam, 1989), chap. 2, pp. 115-122.
169. J. B. Kruskal, M. Liberman, in *Time warps, string edits and macromolecules: the theory and practice of sequence comparison*, D. Sankoff, J. B. Kruskal, Eds. (Center for the Study of Language and Information Publications, Leland Stanford University, CA, USA, 1999), chap. 4, pp. 125-161.
170. S. Kwong, C. W. Chau, W. A. Halang, *IEEE Trans. Ind. Electron.* **43**, 559 (1996).
171. J. R. Lakovicz, *Principles of fluorescence spectroscopy* (Kluwer Academic/Plenum Publishers, New York, NY, US, 2nd edition, 1999).
172. A. J. Lau, B. H. Seo, S. O. Woo, H. L. Koh, *J. Chromatogr. A* **1057**, 141 (2004).
173. W. H. Lawton, E. A. Sylvestre, *Technometrics* **13**, 617 (1971).
174. W. H. Lawton, E. A. Sylvestre, M. S. Maggio, *Technometrics* **14**, 513 (1972).
175. G. C. Lee, D. L. Woodruff, *Anal. Chim. Acta* **513**, 413 (2004).
176. J. K. Lee, R. T. Ross, S. Thampi, S. Leurgans, *J. Phys. Chem.* **96**, 9158 (1992).
177. S. Leurgans, R. T. Ross, *Stat. Sci.* **7**, 289 (1992).
178. S. E. Leurgans, "Multilinear Models: Array Formulas", Technical report 473 (Department of Statistics, The Ohio State University, Columbus, OH, USA, 1991).
179. S. E. Leurgans, R. T. Ross, R. B. Abel, *SIAM J. Matrix Anal. Appl.* **14**, 1064 (1993).
180. M. A. B. Levi, L. S. Scarminio, R. J. Poppi, M. G. Trevisan, *Talanta* **62**, 299 (2004).
181. Y. Li, J. H. Jiang, H. L. Wu, Z. P. Chen, R. Q. Yu, *Chemometr. Intell. Lab. Syst.* **52**, 33 (2000).
182. Y. Li, C. L. Wen, Z. Xie, X. H. Xu, "Synchronization of batch trajectory based on multi-scale dynamic time warping" International Conference on Machine Learning and Cybernetics (2003), pp. 2403-2408
183. S. M. Lin, R. P. Haney, M. J. Campa, M. C. Fitzgerald, K. G. Paul, *Cancer Informatics* **1**, 32 (2005).
184. M. Linder, R. Sundberg, *J. Chemom.* **16**, 12 (2002).
185. M. Linder, R. Sundberg, *Chemometrics Intell. Lab. Syst.* **42**, 159 (1998).
186. J. Listgarten, A. Emili, *Mol. Cell. Proteomics* **4**, 419 (2005).
187. J. Listgarten, N. M. Radford, S. T. Roweis, A. Emili, in *Advances in Neural Information Processing Systems*, L. K. Saul, Y. Weiss, L. Bottou, Eds. (MIT Press, Cambridge, MA, USA, 2005), pp. 817-824.

188. X. L. Liu, H. G. Müller, *Journal of the American Statistical Association* **99**, 687 (2004).
189. X. L. Liu, H. G. Müller, *Bioinformatics* **19**, 1937 (2003).
190. X. Q. Liu, N. D. Sidiropoulos, *IEEE Trans. Signal Process.* **49**, 2074 (2001).
191. J. C. Lucero, L. L. Koenig, *J. Acoust. Soc. Am.* **108**, 1408 (2000).
192. M. E. Lundy, R. A. Harshman, J. B. Kruskal, in *Multiway data analysis*, R. Coppi, S. Bolasco, Eds. (Elsevier Science Publishers, Amsterdam, The Netherlands, 1989), pp. 123-130.
193. K. Madsen, H. B. Nielsen, O. Tingleff, *Methods for non-linear least squares problems* (Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, second edition, 2004).
194. K. Madsen, H. B. Nielsen, O. Tingleff, *Optimization with constraints* (Informatics and Mathematical Modelling, Technical University of Denmark, Lingby, Denmark, second edition, 2004).
195. J. R. Magnus, H. Neudecker, *Matrix differential calculus with applications in statistics and econometrics Rev. ed* (John Wiley & Sons Ltd, Chichester, UK, 1999).
196. G. Malmquist, R. Danielsson, *J. Chromatogr. A* **687**, 71 (1994).
197. H. Martens, M. Hoy, B. M. Wise, R. Bro, P. B. Brockhoff, *J. Chemom.* **17**, 153 (2003).
198. H. Martens, M. Martens, *Food. Qual. Prefer.* **11**, 5 (2000).
199. H. Martens, T. Næs, *Multivariate Calibration* (John Wiley & Sons, Chichester, UK, 1996).
200. D. L. Massart *et al.*, *Handbook of Chemometrics and Qualimetrics* (Elsevier, Amsterdam, 1997).
201. L. B. McGown, D. W. Millican, M. S. Cooter, D. S. Burdick, X. M. Tu, *Abstracts of Papers of the American Chemical Society* **198**, B7 (1989).
202. G. J. McLachlan, T. Krishnan, *The EM Algorithm and Extensions* (John Wiley & Sons, New York, NY, USA, 1997).
203. B. C. Mitchell, D. S. Burdick, *J. Chemom.* **8**, 155 (1994).
204. B. C. Mitchell, D. S. Burdick, *Chemometr. Intell. Lab. Syst.* **20**, 149 (1993).
205. F. Miwakeichi *et al.*, *Neuroimage* **22**, 1035 (2004).
206. R. Moore, M. Russell, M. Tomlinson, "Locally constrained dynamic programming in automatic speech recognition", proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 1982 (1982), pp. 1270-1273
207. A. R. Muroski, K. S. Booksh, M. L. Myrick, *Anal. Chem.* **68**, 3534 (1996).

208. C. Myers, L. R. Rabiner, A. E. Rosenberg, *IEEE Trans. ASSP* **28**, 623 (1980).
209. C. S. Myers, L. R. Rabiner, *IEEE Trans. ASSP* **29**, 284 (1981).
210. S. B. Needleman, C. D. Wunsch, *J. Mol. Biol.* **48**, 443 (1970).
211. P. R. C. Nelson, P. A. Taylor, J. F. MacGregor, *Chemometrics Intell. Lab. Syst.* **35**, 45 (1996).
212. D. Neogi, C. E. Schlags, *Ind. Eng. Chem. Res.* **37**, 3971 (1998).
213. H. Ney, *IEEE Trans. ASSP* **32**, 263 (1984).
214. N. P. V. Nielsen, J. M. Carstensen, J. Smedsgaard, *J. Chromatogr. A* **805**, 17 (1998).
215. N. P. V. Nielsen, J. Smedsgaard, J. C. Frisvad, *Anal. Chem.* **71**, 727 (1999).
216. R. P. H. Nikolajsen, *Development of a method for measurement of stress hormones*, Ph.D. dissertation, The Royal Veterinary and Agricultural University (2002).
217. R. P. H. Nikolajsen, K. S. Booksh, A. M. Hansen, R. Bro, *Anal. Chim. Acta* **475**, 137 (2003).
218. R. P. H. Nikolajsen, A. M. Hansen, R. Bro, *Luminescence* **16**, 91 (2001).
219. P. Nomikos, J. F. MacGregor, *Technometrics* **37**, 41 (1995).
220. D. V. O'connor, D. Phillips, *Time-correlated single photon counting* (Academic Press, London, 1 edition, 1984).
221. J. Ohman, P. Geladi, S. Wold, *J. Chemom.* **4**, 79 (1990).
222. V. Y. Orekhov, I. Ibraghimov, M. Billeter, *J. Biomol. NMR* **27**, 165 (2003).
223. V. Y. Orekhov, I. V. Ibraghimov, M. Billeter, *J. Biomol. NMR* **20**, 49 (2001).
224. P. Paatero, *Chemometr. Intell. Lab. Syst.* **38**, 223 (1997).
225. P. Paatero, *J. Comput. Graph. Stat.* **8**, 854 (1999).
226. P. Paatero, *J. Chemom.* **14**, 285 (2000).
227. K. Paliwal, A. Agarwal, S. Sinha, "A modification over Sakoe and Chiba's dynamic time warping algorithm for isolated word recognition", proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 1982 (1982), pp. 1259-1261
228. K. K. Paliwal, A. Agarwal, S. S. Sinha, *Signal Process.* **4**, 329 (1982).
229. H. T. Pedersen, *Low-field Nuclear Magnetic Resonance and Chemometrics Applied in Food Science*, The Royal Veterinary and Agricultural University (2001). Available at <http://www.models.kvl.dk/research/theses>.
230. H. T. Pedersen, R. Bro, S. B. Engelsen, *J. Magn. Reson.* **157**, 141 (2002).

231. K. M. Pierce, J. L. Hope, K. J. Johnson, B. W. Wright, R. E. Synovec, *J. Chromatogr. A* **1096**, 101 (2005).
232. A. Pikrakis, S. Theodoridis, D. Kamarotos, *IEEE Trans. Speech Audio Process.* **11**, 175 (2003).
233. V. Pravdova, B. Walczak, D. L. Massart, *Anal. Chim. Acta* **456**, 77 (2002).
234. B. J. Prazen, R. E. Synovec, B. R. Kowalski, *Anal. Chem.* **70**, 218 (1998).
235. W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C* (Cambridge University Press, Cambridge, UK, 2nd edition, 2002).
236. L. R. Rabiner, *Proc. IEEE* **77**, 257 (1989).
237. L. R. Rabiner, B. Juang, *Fundamentals of speech recognition* (Prentice Hall, Englewood Cliffs, NJ, USA, 1993).
238. L. R. Rabiner, B. H. Juang, *Technometrics* **33**, 251 (1991).
239. L. R. Rabiner, A. E. Rosenberg, S. E. Levinson, *IEEE Trans. ASSP* **26**, 575 (1978).
240. H. J. Ramaker, E. N. M. van Sprang, S. P. Gurden, J. A. Westerhuis, A. K. Smilde, *J. Process Control* **12**, 569 (2002).
241. H. J. Ramaker, E. N. M. van Sprang, J. A. Westerhuis, A. Smilde, *Anal. Chim. Acta* **498**, 133 (2003).
242. J. O. Ramsay, X. Li, *J. R. Stat. Soc. Ser. B Stat. Methodol.* **60**, 351 (1998).
243. J. O. Ramsay, B. W. Silverman, *Functional Data Analysis* (Springer, New York, NY, USA, 2nd edition, 2005).
244. C. R. Rao, S. Mitra, *Generalized inverse of matrices and its applications*. (John Wiley & Sons, 1971).
245. W. S. Rayens, B. C. Mitchell, *Chemometr. Intell. Lab. Syst.* **38**, 173 (1997).
246. E. Reiner, L. E. Abbey, T. F. Moran, P. Papamichalis, R. W. Shafer, *Biomed. Mass Spectrom.* **6**, 491 (1979).
247. M. M. Reis, S. P. Gurden, A. K. Smilde, M. M. C. Ferreira, *Anal. Chim. Acta* **422**, 21 (2000).
248. Å. Rinnan, *Application of PARAFAC on spectral data*, Ph.D. dissertation, The Royal Veterinary and Agricultural University (2004). Available at <http://www.models.kvl.dk/research/theses>.
249. J. Riu, R. Bro, *Chemometrics Intell. Lab. Syst.* **65**, 35 (2003).
250. B. B. Rønn, *J. R. Stat. Soc. Ser. B Stat. Methodol.* **63**, 243 (2001).
251. TPALS, Ross, R. T. (1995) www.biosci.ohio-state.edu/~rtr/docs.html
252. R. T. Ross *et al.*, *Biochim. Biophys. Acta* **1056**, 317 (1991).

253. R. T. Ross, S. Leurgans, *Methods Enzymol.* **246**, 679 (1995).
254. M. Russell, R. Moore, M. Tomlinson, "Some techniques for incorporating local timescale variability information into a dynamic time-warping algorithm for automatic speech recognition", proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 1983 (1983), pp. 1037-1040
255. M. D. Russell, M. Gouterman, *Spectroc. Acta Pt. A-Molec. Biomolec. Spectr.* **44**, 857 (1988).
256. M. D. Russell, M. Gouterman, *Spectroc. Acta Pt. A-Molec. Biomolec. Spectr.* **44**, 863 (1988).
257. M. D. Russell, M. Gouterman, J. A. Vanzee, *Spectroc. Acta Pt. A-Molec. Biomolec. Spectr.* **44**, 873 (1988).
258. H. Sakoe, *IEEE Trans. ASSP* **27**, 588 (1979).
259. H. Sakoe, S. Chiba, *IEEE Trans. ASSP* **26**, 43 (1978).
260. H. Sakoe, S. Chiba, "A dynamic-programming approach to continuous speech recognition", proceedings of the International Congress of Acoustics (Budapest, Hungary, 1971), Paper 20 C13.
261. S. Salvador, P. Chan, "FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space" (2004) Available at www.cs.fit.edu/~pkc/papers/.
262. M. Sambur, L. Rabiner, *Acoustics, Speech, and Signal Processing [See Also IEEE Transactions on Signal Processing]*, *IEEE Transactions on* **24**, 550 (1976).
263. E. Sanchez, B. R. Kowalski, *J. Chemom.* **29** (1990).
264. E. Sanchez, B. R. Kowalski, *Anal. Chem.* **58**, 496 (1986).
265. J. Saurina, S. HernandezCassou, R. Tauler, *Anal. Chem.* **67**, 3722 (1995).
266. A. C. Sauve, T. P. Speed, "Normalization, baseline correction and alignment of high-throughput mass spectrometry data" Workshop on Genomic Signal Processing and Statistics (Baltimore, MD, USA, 2004) Available at <http://www.stat.berkeley.edu/users/terry/Group/publications.html>.
267. J. R. Schott, *Matrix Analysis for Statistics* (John Wiley & Sons, Chichester, 1997).
268. E. Selli, C. Zaccaria, F. Sena, G. Tomasi, G. Bidoglio, *Water Res.* **38**, 2269 (2004).
269. M. M. Sena, J. C. B. Fernandes, L. Rover, R. J. Poppi, L. T. Kubota, *Anal. Chim. Acta* **409**, 159 (2000).
270. R. Siciliano, A. Mooijaart, *Comput. Stat. Data Anal.* **24**, 337 (1997).
271. N. D. Sidiropoulos, R. Bro, *J. Chemom.* **14**, 229 (2000).

272. N. D. Sidiropoulos, R. Bro, G. B. Giannakis, *IEEE Trans. Signal Process.* **48**, 2377 (2000).
273. N. D. Sidiropoulos, G. B. Giannakis, R. Bro, *IEEE Trans. Signal Process.* **48**, 810 (2000).
274. B. W. Silverman, *J. R. Stat. Soc. Ser. B Stat. Methodol.* **57**, 673 (1995).
275. H. F. Silverman, D. P. Morgan, *IEEE ASSP Mag.* **7**, 6 (1990).
276. A. E. Sinha, C. G. Fraga, B. J. Prazen, R. E. Synovec, *J. Chromatogr. A* **1027**, 269 (2004).
277. D. A. Skoog, D. M. West, J. F. Holler, *Fundamentals of Analytical Chemistry* (Saunders College Publishing, Philadelphia, PA, USA, 6th edition, 1992).
278. A. K. Smilde, R. Bro, P. Geladi, *Multi-way analysis Applications in the chemical sciences* (John Wiley & Sons Ltd, Chichester, England, 2004).
279. A. K. Smilde, H. C. J. Hoefsloot, H. A. L. Kiers, S. Bijlsma, H. F. M. Boelens, *J. Chemom.* **15**, 405 (2001).
280. A. K. Smilde, R. Tauler, J. M. Henshaw, L. W. Burgess, B. R. Kowalski, *Anal. Chem.* **66**, 3345 (1994).
281. A. K. Smilde, R. Tauler, J. Saurina, R. Bro, *Anal. Chim. Acta* **398**, 237 (1999).
282. A. K. Smilde, Y. D. Wang, B. R. Kowalski, *J. Chemom.* **8**, 21 (1994).
283. M. Sniedovich, *Dynamic Programming* (Marcel Dekker Ltd, New York, NY, USA, 1991).
284. R. Srinivasan, M. S. Qian, *Ind. Eng. Chem. Res.* **44**, 4697 (2005).
285. G. N. Stainhaouer, G. Carayannis, *IEEE Trans. ASSP* **38**, 705 (1990).
286. Y. Stettiner, D. Malah, D. Chazan, "Dynamic time warping with path control and non-local cost", proceedings of the 12th IAPR International Conference on Pattern Recognition (1994), pp. 174-177
287. R. Stoyanova, A. W. Nicholls, J. K. Nicholson, J. C. Lindon, T. R. Brown, *J. Magn. Reson.* **170**, 329 (2004).
288. R. S. Sutton, A. G. Barto, *Reinforcement Learning: an introduction* (MIT Press, Cambridge, MA, USA, 1998).
289. Y. X. Tan, J. H. Jiang, H. L. Wu, H. Cui, R. Q. Yu, *Anal. Chim. Acta* **412**, 195 (2000).
290. R. Tauler, *Chemometr. Intell. Lab. Syst.* **30**, 133 (1995).
291. R. Tauler, A. Smilde, B. Kowalski, *J. Chemom.* **9**, 31 (1995).
292. J. M. F. Ten Berge, *J. Chemom.* **18**, 17 (2004).

293. J. M. F. Ten Berge, *Psychometrika* **65**, 525 (2000).
294. J. M. F. Ten Berge, H. A. L. Kiers, *Linear Alg. Appl.* **294**, 169 (1999).
295. J. M. F. Ten Berge, N. D. Sidiropoulos, *Psychometrika* **67**, 399 (2002).
296. J. M. F. Ten Berge, A. K. Smilde, *J. Chemom.* **16**, 609 (2002).
297. L. G. Thygesen, A. Rinnan, S. Barsberg, J. K. S. Moller, *Chemometr. Intell. Lab. Syst.* **71**, 97 (2004).
298. Y. Tohkura, *IEEE Trans. ASSP* **35**, 1414 (1987).
299. R. J. O. Torgrip, M. Aberg, B. Karlberg, S. P. Jacobsson, *J. Chemom.* **17**, 573 (2003).
300. X. M. Tu, D. S. Burdick, *Stat. Sin.* **2**, 577 (1992).
301. S. Uchida, H. Sakoe, *IEICE Trans. Inf. Syst.* **E82D**, 693 (1999).
302. S. Uchida, H. Sakoe, *IEICE Trans. Inf. Syst.* **E83D**, 109 (2000).
303. F. A. Unal, N. Tepedelenlioglu, "Dynamic time warping using an artificial neural network", proceedings of the International Joint Conference on Neural Networks (1992), pp. 715-721
304. C. Ündey, S. Ertunç, A. Çınar, *Ind. Eng. Chem. Res.* **42**, 4645 (2003).
305. F. van den Berg, G. Tomasi, N. Viereck, in *Magnetic Resonance in Food Science The multivariate challenge*, S. B. Engelsen, P. S. Belton, H. J. Jakobsen, Eds. (Royal Chemistry Society, Cambridge, UK, 2005), pp. 131-138.
306. H. van der Vorst. Iterative Krylov methods for large linear systems. Ablowitz, M. J., Davis, S. H., Hinch, E. J., Iserles, A., Ockendon, J., and Olver, P. J. (13). 2003. Cambridge, U.K., Cambridge University Press. Cambridge Monographs on Applied and Computational Mathematics.
Ref Type: Serial (Book, Monograph)
307. V. G. van Mispelaar, A. C. Tas, A. K. Smilde, P. J. Schoenmakers, A. C. van Asten, *J. Chromatogr. A* **1019**, 15 (2003).
308. A. M. van Nederkassel, M. Daszykowski, D. L. Massart, Y. Vander Heyden, *J. Chromatogr. A* **1096**, 177 (2005).
309. A. M. van Nederkassel, V. Vijverman, D. L. Massart, Y. Vander Heyden, *J. Chromatogr. A* **1085**, 230 (2005).
310. L. Vega-Montoto, P. D. Wentzell, *J. Chemom.* **17**, 237 (2003).
311. V. M. Velichko, N. G. Zagoruyko, *Int. J. Man-Machine Studies* **2**, 223 (1970).
312. T. K. Vintsyuk, *Kibernetika* **7**, 133 (1971).
313. T. K. Vintsyuk, *Kibernetika* **4**, 81 (1968).

314. G. Vivó-Truyols, J. R. Torres-Lapasió, A. M. van Nederkassel, Y. Vander Heyden, D. L. Massart, *J. Chromatogr. A* **1096**, 133 (2005).
315. G. Vivó-Truyols, J. R. Torres-Lapasió, A. M. van Nederkassel, Y. Vander Heyden, D. L. Massart, *J. Chromatogr. A* **1096**, 146 (2005).
316. J. T. W. E. Vogels, A. C. Tas, J. Venekamp, J. VanderGreef, *J. Chemom.* **10**, 425 (1996).
317. F. Vogt, K. Booksh, *Appl. Spectrosc.* **58**, 624 (2004).
318. H. J. L. M. Vullings, M. H. G. Verhaegen, H. B. Verbruggen, "Automated ECG segmentation with dynamic time warping", proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (1998), pp. 163-166
319. B. Walczak, D. L. Massart, *Chemometrics Intell. Lab. Syst.* **58**, 15 (2001).
320. B. Walczak, D. L. Massart, *Chemometrics Intell. Lab. Syst.* **58**, 29 (2001).
321. C. P. Wang, T. L. Isenhour, *Anal. Chem.* **59**, 649 (1987).
322. J. H. Wang, P. K. Hopke, *Chemometr. Intell. Lab. Syst.* **55**, 13 (2001).
323. K. Wang, H. Begleiter, B. Porjesz, *Clin. Neurophysiol.* **112**, 1917 (2001).
324. K. Wang, T. Gasser, *Ann. Statist.* **25**, 1251 (1997).
325. K. Wang, T. Gasser, *Ann. Statist.* **27**, 439 (1999).
326. Y. D. Wang, O. S. Borgen, B. R. Kowalski, *J. Chemom.* **7**, 439 (1993).
327. Y. D. Wang, O. S. Borgen, B. R. Kowalski, M. Gu, F. Turecek, *J. Chemom.* **7**, 117 (1993).
328. P. D. Wentzell, S. S. Nair, R. D. Guy, *Anal. Chem.* **73**, 1408 (2001).
329. F. Westad, H. Martens, *Chemometr. Intell. Lab. Syst.* **45**, 361 (1999).
330. J. A. Westerhuis, T. Kourti, J. F. MacGregor, *J. Chemom.* **13**, 397 (1999).
331. G. White, R. Neely, *IEEE Trans. ASSP* **24**, 183 (1976).
332. K. Wiberg, S. P. Jacobsson, *Anal. Chim. Acta* **514**, 203 (2004).
333. A. Willse *et al.*, *Anal. Chem.* **77**, 2348 (2005).
334. B. E. Wilson, B. R. Kowalski, *Anal. Chem.* **61**, 2277 (1989).
335. W. Windig, B. Antalek, *Chemometrics Intell. Lab. Syst.* **37**, 241 (1997).
336. B. M. Wise, N. B. Gallagher, E. B. Martin, *J. Chemom.* **15**, 285 (2001).
337. H. Witjes *et al.*, *J. Magn. Reson.* **144**, 35 (2000).
338. H. Witjes, M. Pepers, W. J. Melssen, L. M. C. Buydens, *Anal. Chim. Acta* **432**, 113 (2001).

- 339. H. Witjes, M. van den Brink, W. J. Melssen, L. M. C. Buydens, *Chemometr. Intell. Lab. Syst.* **52**, 105 (2000).
- 340. S. Wold, N. Kettaneh, H. Friden, A. Holmberg, *Chemometrics Intell. Lab. Syst.* **44**, 331 (1998).
- 341. J. W. H. Wong, C. Durante, H. M. Cartwright, *Anal. Chem.* **77**, 5655 (2005).
- 342. H. L. Wu, M. Shibukawa, K. Oguma, *J. Chemom.* **12**, 1 (1998).
- 343. H. L. Wu, M. Shibukawa, K. Oguma, *Anal. Sci.* **13**, 53 (1997).
- 344. F. Wulfert, W. T. Kok, A. K. Smilde, *Anal. Chem.* **70**, 1761 (1998).
- 345. J. Yang *et al.*, *J. Chromatogr. A* **1084**, 214 (2005).
- 346. F. Yates, *Emp. J. Exp. Agric.* **1**, 129 (1933).
- 347. B. J. H. Zijlstra, H. A. L. Kiers, *J. Chemom.* **16**, 596 (2002).

Papers

Paper I

A comparison of algorithms for fitting the PARAFAC
model

G. Tomasi, R. Bro

Computational Statistics and Data Analysis, 50, 2006, 1700-
1734.



A comparison of algorithms for fitting the PARAFAC model

Giorgio Tomasi*, Rasmus Bro

*Food Science, Royal Veterinary and Agricultural University, Rolighedsvej 30,
DK-1958 Frederiksberg C, Denmark*

Received 24 November 2004; accepted 26 November 2004

Available online 22 December 2004

Abstract

A multitude of algorithms have been developed to fit a trilinear PARAFAC model to a three-way array. Limits and advantages of some of the available methods (i.e. GRAM-DTLD, PARAFAC-ALS, ASD, SWATLD, PMF3 and dGN) are compared. The algorithms are explained in general terms together with two approaches to accelerate them: line search and compression. In order to compare the different methods, 720 sets of artificial data were generated with varying level and type of noise, collinearity of the factors and rank. Two PARAFAC models were fitted on each data set: the first having the correct number of factors F and the second with $F + 1$ components (the objective being to assess the sensitivity of the different approaches to the over-factoring problem, i.e. when the number of extracted components exceeds the rank of the array). The algorithms have also been tested on two real data sets of fluorescence measurements, again by extracting both the right and an exceeding number of factors. The evaluations are based on: number of iterations necessary to reach convergence, time consumption, quality of the solution and amount of resources required for the calculations (primarily memory).

© 2005 Elsevier B.V. All rights reserved.

Keywords: PARAFAC; ALS; SWATLD; PMF3; Levenberg–Marquadt

* Corresponding author. Tel.: +45 35 28 32 96; fax: +45 55 28 32 45.

E-mail addresses: gt@kvl.dk (G. Tomasi), rb@kvl.dk (R. Bro).

1. Introduction

The PARAFAC (PARallel FACtor analysis) model was introduced in 1970 by Harshman (1970) and simultaneously by Carroll and Chang (1970) under the name CANDECOMP. For a three-way data array $\underline{\mathbf{X}}$, the PARAFAC model is defined as

$$x_{ijk} = \sum_{f=1}^F a_{if} b_{jf} c_{kf} + r_{ijk}, \quad i = 1 \dots I, \quad j = 1 \dots J, \quad k = 1 \dots K, \quad (1)$$

where x_{ijk} is the measured value, a_{if} , b_{jf} , and c_{kf} represent the parameters to estimate, r_{ijk} are the residuals and F is the number of factors extracted.

In general terms, fitting model (1) boils down to minimising (usually in a least squares sense) the fitting error made. This means finding the parameters $a_{11}, a_{12}, \dots, c_{KF}$ that minimise the loss function:

$$L(a_{11}, a_{12}, \dots, c_{KF}) = \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \left(x_{ijk} - \sum_{f=1}^F a_{if} b_{jf} c_{kf} \right)^2. \quad (2)$$

Several algorithms for solving such problems are described in the literature (Faber et al., 2003; Hayashi and Hayashi, 1982; Paatero, 1997; Tomasi and Bro, 2004). Some of those tested in this work (DTLD, ASD and SWATLD) do not minimise (2) and in fact their loss functions are not strictly well-defined (Faber et al., 2003).

While the PARAFAC model remains the same, the different methods of fitting it to a three-way array $\underline{\mathbf{X}}$ can be better explained introducing distinct notations. The parameters can be gathered in three loading matrices \mathbf{A} (also referred to as scores matrix), \mathbf{B} and \mathbf{C} defined as

$$\mathbf{A} = \{a_{if} | i = 1 \dots I, f = 1 \dots F\} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \dots \quad \mathbf{a}_F], \quad (3a)$$

$$\mathbf{B} = \{b_{jf} | j = 1 \dots J, f = 1 \dots F\} = [\mathbf{b}_1 \quad \mathbf{b}_2 \quad \dots \quad \mathbf{b}_F] \quad (3b)$$

and

$$\mathbf{C} = \{c_{kf} | k = 1 \dots K, f = 1 \dots F\} = [\mathbf{c}_1 \quad \mathbf{c}_2 \quad \dots \quad \mathbf{c}_F], \quad (3c)$$

where \mathbf{a}_f , \mathbf{b}_f and \mathbf{c}_f denote the columns of \mathbf{A} , \mathbf{B} and \mathbf{C} respectively.

Using the matricised format for multi-way arrays (Bro, 1998) and with the introduction of the column-wise Khatri–Rao product \odot (Rao and Mitra, 1971) (see Appendix C), the model can be written as

$$\mathbf{X}^{(I \times JK)} = \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T + \mathbf{R}^{(I \times JK)}, \quad (4)$$

where “ T ” means the transpose and $(I \times JK)$ refers to the way the multi-way array is matricised (K slabs of size $I \times J$ are put one beside the other forming a matrix of dimension $I \times JK$).

Another possibility is to express the model in a slab-wise form as:

$$\mathbf{X}_{:,k} = \mathbf{A} \mathbf{D}_k \mathbf{B}^T + \mathbf{R}_{:,k}, \quad (5)$$

where \mathbf{D}_k is a diagonal matrix containing the k -th row of \mathbf{C} , $\mathbf{X}_{:,k}$ is the $I \times J$ matrix representing the k -th frontal slab and $\mathbf{R}_{:,k}$ is the corresponding matrix of the residuals. Eqs. (4) and (5) have to be trivially modified in case other matricisations (e.g. $J \times IK$) or slabs (e.g. the horizontal slab $\mathbf{X}_{i,:}$ of size $K \times J$) are required (Bro, 1998).

The algorithms fitting the PARAFAC model can be classified in three main groups: alternating algorithms, which update only a subset of the parameters at each step; derivative-based methods, seeking an update for all the parameters simultaneously by successive approximations; and direct (non-iterative) procedures. To the first group belong PARAFAC-ALS (Alternating Least Squares (Harshman, 1970)), ASD (Alternating Slice-wise Diagonalisation (Jiang et al., 2000)) and SWATLD (Self Weighted Alternating TriLinear Decomposition (Chen et al., 2000)). The second set is represented by PMF3 (Positive Matrix Factorisation for 3-way arrays (Paatero, 1997)) and dGN (damped Gauss–Newton, also known as Levenberg–Marquadt (Paatero, 1997; Tomasi and Bro, 2004)). As for the third grouping of algorithms, the most known implementations are the Generalised Rank Annihilation Method (GRAM (Sanchez and Kowalski, 1986)) and the Direct TriLinear Decomposition method (DTLD), both based on a generalized eigenvalue problem. All these algorithms will only be described in general terms as more details are available in the original papers.

Fitting the PARAFAC model presents numerous problems. One aspect affecting speed of convergence and retrieval of the underlying solutions is the condition number of the loading matrices, which reflects both collinearity and relative magnitude of the factors (Hopke et al., 1998; Kiers, 1998). Compression, which is outlined in Section 2.5, has been reported to be beneficial in this respect and has the added advantage of reducing the computational expense (Bro and Andersson, 1998; Kiers, 1998). Another problem is the so-called two factor degeneracy (2FD), i.e. the presence in the solution of two factors that are almost perfectly collinear but have opposite signs and almost cancel out each others' contribution. The essential problem with 2FDs is that the degenerate factors can grow arbitrarily large while the loss function continuously (and very slowly) decreases. 2FDs may appear in the final solution as a consequence of e.g. a wrong estimation of the rank of the array, or some aspects specific to the data set at hand, (Kruskal et al., 1989; Paatero, 2000). A connection has been established between some of the possible causes of 2FDs in the final solution and the so-called swamps (Mitchell and Burdick, 1994; Rayens and Mitchell, 1997), which are sequences of iterations where the interim solutions contain features similar to 2FDs of increasing severity and the loss function decreases very slowly. When an (iterative) algorithm encounters a swamp, it either emerges from it (i.e. after an unpredictably large number of iterations, the loss function starts decreasing more rapidly again and the 2FD slowly disappears) or it reaches an earlier stop because one of the convergence criteria is suddenly met. Great attention has been given in the development of new methods capable of dealing effectively with swamps and more in general with the more frequent high collinearity case. Two examples are regularisation (Paatero, 1997; Rayens and Mitchell, 1997) and line search (Bro, 1998; Harshman, 1970) (described in Section 2.4).

2. Algorithms

2.1. DTLD

Ho et al. (1978, 1980, 1981) developed an algorithm called RAFA (Rank Annihilation Factor Analysis) for estimating the concentration of a chemical analyte in an unknown sample-matrix solely using the measurements of the unknown sample and of a pure standard. This property was coined the second-order advantage, as it is obtained by using the second-order or two-way structure of the individual sample measurements instead of vectorising the corresponding matrix. The second-order advantage is in essence equivalent to the uniqueness of the trilinear structure. The idea behind RAFA was based on reducing the rank of the calibration sample by subtracting the contribution from the analyte of interest. That is, if the signal from the analyte of interest is subtracted from the sample data, then the rank of this matrix will decrease by one as the contribution of the analyte of interest is one in case of ordinary bilinear data such as chromatographic or fluorescence data. This was intuitively appealing, but the method itself was somewhat unsatisfactory and slow. Later, Lorber (1985) showed that it was possible to automate this search for rank-reduction and extended the method. This new automated method was called Generalized Rank Annihilation Method (GRAM) and works with two samples. Using only two samples, there is a direct solution to the PARAFAC model based on solving a generalized eigenvalue problem. Despite the fact that this solution is not a least squares solution, it has been found to work well for data that are well approximated by a PARAFAC model. From a GRAM solution for any two samples, the loading matrices in the two variable modes are obtained (as well as the two scores for the sample mode). For more than two samples, the two loading matrices obtained from any two samples can be used for calculating the score-values for all samples in a simple regression step. This is the principle behind DTLD (Sanchez and Kowalski, 1990), whereby there are alternative strategies for selecting the two samples used in the GRAM step. Thus, DTLD is a direct method using a generalized eigenvalue problem of fixed known complexity. The required storage is represented in essence by the sole data array \mathbf{X} , the three loading matrices and three matrices (two with the same size of the loading matrices and one having only two columns) used to compress the array and to select the two samples used for GRAM step.

2.2. Alternating Least Squares

Alternating least squares is a relatively old method (its principles were introduced in 1933 by Yates (1933)) and it is based on the idea of reducing the optimisation problem to smaller sub-problems that are solved iteratively. The parameters to be determined are separated in different groups and, by fixing all of the groups but one, a new loss function depending only on the set left free to vary is minimised. The solution of this (linear) least squares problem is known and relatively simple to calculate. The subsequent stages of the algorithm consist of applying the same principle on the other groups of parameters. The algorithm iterates, alternating from one set to the next, until the variation of the loss function or of the parameters is less than a predefined convergence criterion. Since all the steps are

optimisations in the least squares sense, the loss function is bound not to increase at any step and tends asymptotically to a minimum.

2.2.1. PARAFAC-ALS (ALS)

In the PARAFAC case (Carroll and Chang, 1970; Harshman, 1970) and having a three-way array \mathbf{X} , there are three sets of parameters (\mathbf{A} , \mathbf{B} , and \mathbf{C}) and the loss function (2), using (4), can be written as

$$L(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \left\| \mathbf{X}^{(I \times JK)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T \right\|_F^2, \quad (6)$$

where $\|\bullet\|_F$ denotes the Frobenius norm. While calculating all the three loading matrices that minimise L at the same time is a rather complicated non-linear problem (and it is treated as such in other algorithms), if initial approximations of \mathbf{B} and \mathbf{C} (respectively $\mathbf{B}^{(0)}$ and $\mathbf{C}^{(0)}$) are available, the interim optimal \mathbf{A} can be easily calculated as

$$\mathbf{A}^{(1)} = \mathbf{X}^{(I \times JK)} \left(\left(\mathbf{C}^{(0)} \odot \mathbf{B}^{(0)} \right)^+ \right)^T, \quad (7a)$$

where ‘+’ denotes the Moore–Penrose inverse. $\mathbf{B}^{(1)}$ and $\mathbf{C}^{(1)}$ are determined in a similar fashion. That is

$$\mathbf{B}^{(1)} = \mathbf{X}^{(J \times IK)} \left(\left(\mathbf{C}^{(0)} \odot \mathbf{B}^{(1)} \right)^+ \right)^T \quad (7b)$$

and

$$\mathbf{C}^{(1)} = \mathbf{X}^{(K \times IJ)} \left(\left(\mathbf{B}^{(1)} \odot \mathbf{A}^{(1)} \right)^+ \right)^T. \quad (7c)$$

The three steps in Eq. (7) are repeated until a predefined convergence criterion is met.

Eq. (7) shows the general approach of PARAFAC-ALS. In the actual implementation of the algorithm (Jiang et al., 1999), certain properties of the Khatri–Rao product and line search (Section 2.4) are employed in order to accelerate the calculations (Bro, 1998).

This algorithm has several advantages: it is easy to implement, guaranteed to converge and simple to extend to higher order arrays. The shortcomings are mainly in the occasional slowness of the convergence process in presence of swamps (Mitchell and Burdick, 1994; Paatero, 2000; Rayens and Mitchell, 1997) or high collinearity (Kiers, 1998). Furthermore, the loss function decreases almost linearly with the iterations while other methods can provide, at least in principle, superlinear or even quadratic convergence rate. It is also worth mentioning, even if it is not relevant for the purposes of this work, that several types of constraints can be imposed to the loading vectors in a relatively straightforward way (Bro, 1998).

The memory consumption for PARAFAC-ALS is limited; in essence: two arrays (one for \mathbf{X} and one for the model/residuals) of size $I \times J \times K$, a matrix of size $\max(IJ, JK, IK) \times F$ for the Khatri–Rao products and the three loading matrices.

2.2.2. SWATLD and ASD

Recently, several algorithms (ASD (Jiang et al., 2000), SWATLD (Chen et al., 2000) ATLD (Wu et al., 1998), PALS (Chen et al., 2001), ACOMAR (Li et al., 2000), ACOVER (Jiang et al., 1999)) based on ideas similar to PARAFAC-ALS have been proposed. A comparison between these methods and standard ALS already exists in the literature (Faber et al., 2003). Therefore, based also on some preliminary tests, only SWATLD and ASD have been chosen to represent this group of new algorithms.

As mentioned by its authors, SWATLD does not attempt to find the minimum of (6), instead it alternates between minimising three different (non-least squares) loss functions, one per each of the loading matrices. Such loss functions are structured in the following way (shown here with respect to \mathbf{C}):

$$L_{\mathbf{C}}(\mathbf{C}) = \sum_{k=1}^K \left\| \left(\mathbf{A}^+ \mathbf{X}_k - \mathbf{D}_k \mathbf{B}^T \right)^T \mathbf{D}_{\mathbf{B}}^{-1} \right\|_{\mathbf{F}}^2 + \sum_{k=1}^K \left\| \left(\mathbf{X}_k (\mathbf{B}^+)^T - \mathbf{A} \mathbf{D}_k \right) \mathbf{D}_{\mathbf{A}}^{-1} \right\|_{\mathbf{F}}^2, \quad (8)$$

where $\mathbf{D}_{\mathbf{A}}$ and $\mathbf{D}_{\mathbf{B}}$ are $F \times F$ diagonal matrices holding the norm of the loading vectors of respectively \mathbf{A} and \mathbf{B} , $\mathbf{X}_{:,k}$ is the k -th frontal slab of \mathbf{X} and \mathbf{D}_k is an $F \times F$ diagonal matrix with the elements of the k -th row of \mathbf{C} on the diagonal. Thus, given \mathbf{A} and \mathbf{B} , a new estimate for \mathbf{C} can be found as (cf. Appendix B):

$$\mathbf{C} = 0.5 \cdot \mathbf{X}^{(K \times IJ)} \left[\left(\mathbf{B}^{+T} \odot \mathbf{A} \mathbf{D}_{\mathbf{A}}^{-2} \right) + \left(\mathbf{B} \mathbf{D}_{\mathbf{B}}^{-2} \odot \mathbf{A}^{+T} \right) \right]. \quad (9)$$

The update evidently depends on the values of $\mathbf{D}_{\mathbf{A}}$ and $\mathbf{D}_{\mathbf{B}}$ and thus on the scaling convention applied to the interim loading matrices. Hence, the loading matrices were scaled so that $\mathbf{D}_{\mathbf{A}} = \mathbf{D}_{\mathbf{B}}$ (cf. Appendix A). Equivalent equations are developed mutatis mutandis for the updates of \mathbf{A} and \mathbf{B} . It is apparent that the solution is not found in the least squares sense as $0.5 \cdot \left[\left(\mathbf{B}^{+T} \odot \mathbf{A} \mathbf{D}_{\mathbf{A}}^{-2} \right) + \left(\mathbf{B} \mathbf{D}_{\mathbf{B}}^{-2} \odot \mathbf{A}^{+T} \right) \right] \neq (\mathbf{B} \odot \mathbf{A})^{+T}$. In fact, SWATLD typically yields solutions that are less affected by deviations from exact trilinearity than the least squares ones (see the experimental part), which seems to bear some advantages in terms of convergence speed and resistance to over-factoring. However, since its optimisation criterion is not well-defined, this algorithm must be used with care.

One important remark about SWATLD regards the three matrices \mathbf{A} , \mathbf{B} and \mathbf{C} , which must have full column rank in order for the algorithm to resolve uniquely the components of interest. In other words, the maximum number of components that can be extracted is upper-bounded by any of the dimensions of \mathbf{X} . Such conditions for uniqueness are stricter than the theoretical ones (Sidiropoulos and Bro, 2000).

With regard to memory consumption, SWATLD is approximately as demanding as ALS.

ASD is based on the hypothesis of \mathbf{A} and \mathbf{B} having full column rank (hence also stricter than the theoretical conditions for uniqueness (Sidiropoulos and Bro, 2000)) and thus on the existence of two matrices \mathbf{P} and \mathbf{Q} for which it holds $\mathbf{P}^T \mathbf{A} = \mathbf{I}_F$ and $\mathbf{B}^T \mathbf{Q} = \mathbf{I}_F$, where \mathbf{I} denotes the identity matrix of appropriate size.

Eq. (5) can then be written as

$$\mathbf{P}^T \mathbf{X}_{:,k} \mathbf{Q} = \mathbf{D}_k + \mathbf{P}^T \mathbf{R}_{:,k} \mathbf{Q} \equiv \mathbf{D}_k + \tilde{\mathbf{R}}_{:,k}. \quad (10)$$

Loss function (6) is minimised by ASD using an alternating algorithm of the form

$$L_{\text{ASD}}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{P}, \mathbf{Q}) = \sum_{k=1}^K \left\| \mathbf{P}^T \mathbf{X}_{..k} \mathbf{Q} - \mathbf{D}_k \right\|_F^2 + \lambda \left(\left\| \mathbf{P}^T \mathbf{A} - \mathbf{I}_F \right\|_F^2 + \left\| \mathbf{B}^T \mathbf{Q} - \mathbf{I}_F \right\|_F^2 \right), \quad (11)$$

where λ is a predefined constant (Faber et al., 2003; Jiang et al., 2000).

It is apparent that L_{ASD} is a function of $\tilde{\mathbf{R}}_{..k}$, not of the residuals, and that it contains a penalty term that does not necessarily decrease to a negligible magnitude. Consequently, as it is the case for SWATLD, the final solution is not providing a least squares fit and must be assessed with care.

A final observation is that ASD employs a compression procedure based on singular value decomposition that allows a significant reduction in the number of operations per single iteration (Faber et al., 2003; Jiang et al., 2000). This compression step further reduces the memory consumption of the algorithm, which requires, beside the original array and the loading matrices, an array $\tilde{\mathbf{R}}$ of size $F \times F \times K$, four matrices of size $F \times F$ and 2 matrices holding the bases for the compression and having size $I \times F$ and $J \times F$.

2.3. Gauss–Newton method

As mentioned in Section 2.2.1, minimising (6) is a relatively difficult non-linear least squares problem: several authors (Hayashi and Hayashi, 1982; Paatero, 1997; Tomasi and Bro, 2004) have proposed the use of algorithms usually employed for such optimisation problems. All these algorithms are based on the Gauss–Newton method (Björck, 1996). However, in order to deal with some of the peculiarities of the PARAFAC model it is necessary to introduce some modifications to the basic algorithm.

Defining the model values as

$$y_{ijk}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \sum_{f=1}^F a_{if} b_{jf} c_{kf}, \quad (12)$$

loss function (6) can be written as

$$\begin{aligned} L(\mathbf{A}, \mathbf{B}, \mathbf{C}) &= \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K (x_{ijk} - y_{ijk}(\mathbf{A}, \mathbf{B}, \mathbf{C}))^2 \\ &= \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K r_{ijk}^2(\mathbf{A}, \mathbf{B}, \mathbf{C}), \end{aligned} \quad (13)$$

where $r_{ijk}(\mathbf{A}, \mathbf{B}, \mathbf{C})$ are the residuals. By stringing out \mathbf{X} and \mathbf{Y} in two vectors $\mathbf{x} = \text{vec } \mathbf{X}^{(I \times JK)}$ and $\mathbf{y} = \text{vec } (\mathbf{A}(\mathbf{C} \odot \mathbf{B})^T)$ of length $M = IJK$ (the vec operator is defined as in Magnus and Neudecker (1999)) and considering a vector $\mathbf{p} = \text{vec } [\mathbf{A}^T | \mathbf{B}^T | \mathbf{C}^T]$ of length $N = (I + J + K)F$ with all the sought parameters, loss function (13) can be

expressed as

$$L(\mathbf{p}) = \sum_{m=1}^M (x_m - y_m(\mathbf{p}))^2 = \sum_{m=1}^M r_m^2(\mathbf{p}) = \mathbf{r}(\mathbf{p})^T \mathbf{r}(\mathbf{p}), \quad (14)$$

where $m = IJ(k-1) + I(j-1) + i$ and $\mathbf{r} = [r_1 \ \dots \ r_M]^T$.

The Gauss–Newton method works under the assumption that the residuals in the neighbourhood of a point \mathbf{p}^0 can be approximated by a Taylor expansion truncated after the linear term:

$$\begin{aligned} r_m(\mathbf{p}) &= r_m(\mathbf{p}^0) + \sum_{n=1}^N \frac{\partial r_m}{\partial p_n} (p_n - p_n^0) + O\left(\|\mathbf{p} - \mathbf{p}^0\|_2^2\right) \\ &\cong r_m(\mathbf{p}^0) - \sum_{n=1}^N \frac{\partial y_m}{\partial p_n} (p_n - p_n^0) \equiv \tilde{r}_m(\mathbf{p}), \quad m = 1, \dots, M. \end{aligned} \quad (15)$$

Thus, by defining an $M \times N$ matrix $\mathbf{J}(\mathbf{p}^0)$, the Jacobian, having as elements j_{mn}

$$j_{mn} = \frac{\partial r_m(\mathbf{p}^0)}{\partial p_n} = - \frac{\partial y_m(\mathbf{p}^0)}{\partial p_n}$$

and if the linear approximation (15) suffices, (14) can be expressed as a function of $\Delta\mathbf{p} = \mathbf{p} - \mathbf{p}^0$:

$$\tilde{L}(\Delta\mathbf{p}) = \tilde{\mathbf{r}}(\Delta\mathbf{p})^T \tilde{\mathbf{r}}(\Delta\mathbf{p}) = \left\| \mathbf{r}(\mathbf{p}^0) + \mathbf{J}(\mathbf{p}^0) \Delta\mathbf{p} \right\|_2^2. \quad (16)$$

A new approximation of the parameter-vector is then calculated as $\mathbf{p}^{(s+1)} = \mathbf{p}^{(s)} + \Delta\mathbf{p}^{(s)}$, where the update $\Delta\mathbf{p}^{(s)}$ for the s -th iteration is computed as a solution to the linear problem

$$\min_{\Delta\mathbf{p}} \left\| \mathbf{r}(\mathbf{p}^{(s)}) + \mathbf{J}(\mathbf{p}^{(s)}) \Delta\mathbf{p}^{(s)} \right\|_2^2. \quad (17)$$

This can be done by solving for $\Delta\mathbf{p}^{(s)}$ the system of normal equations:

$$(\mathbf{J}^T \mathbf{J}) \Delta\mathbf{p}^{(s)} = -\mathbf{J}^T \mathbf{r} \equiv \mathbf{g}, \quad (18)$$

where \mathbf{g} is the gradient of $\tilde{L}(\Delta\mathbf{p})$. At this point, convergence is checked and if the loss function has not decreased in relative terms for less than a predefined convergence criterion or the number of iterations has not reached the maximum, the algorithm computes a new Jacobian and a new update $\Delta\mathbf{p}$ and keeps on iterating. Note that, as the minimum of a function is a stationary point, the algorithm should stop if $\|\mathbf{g}\|_\infty$ is smaller than a sufficiently small number (Madsen et al., 2004).

This iterative method for fitting the PARAFAC model (i.e. the Gauss–Newton algorithm) corresponds to the one proposed by Hayashi and Hayashi (1982).

$\tilde{\mathbf{H}} = \mathbf{J}^T \mathbf{J}$ is in fact an approximation of the Hessian matrix \mathbf{H} , which holds the second derivatives of the loss function (Madsen et al., 2004). The smaller the residuals r_m are, the better is the approximation. This implies that if the elements of \mathbf{r} are small enough, the update $\Delta \mathbf{p}^{(s)}$ is close to the one calculated with the Newton method (requiring \mathbf{H}), which guarantees, under the same conditions, quadratic convergence (Madsen et al., 2004). On the other hand, if the behaviour of L is far from linear (e.g. QS the number of modes of the array increases) or the elements of r_m are relatively large, the approximation is not appropriate and the Newton method should be preferred (Madsen et al., 2004). Line search procedures and trust region methods (Björck, 1996; Madsen et al., 2004) can be used to deal with this problem without requiring the full calculation of the Hessian, as outlined in the ensuing sections.

The very sparse structure of \mathbf{J} in the PARAFAC case (Paatero, 1997) induces some observations regarding whether or not it is advantageous to exploit it in solving (17) without using the system of normal equations. By partitioning the Jacobian as $[\mathbf{J}_A | \mathbf{J}_{BC}]$ where \mathbf{J}_A and \mathbf{J}_{BC} refer to the parameters in \mathbf{A} and in \mathbf{B} and \mathbf{C} respectively and by a simple row permutation, it is apparent that problem (17) has a block angular form (i.e. \mathbf{J}_A is block diagonal, each block being $JK \times F$). A substantial reduction in the number of operations could be attained by using a QR algorithm specifically designed for this type of problems (Björck, 1996). Even greater savings may be achieved considering that the I blocks in \mathbf{J}_A are identical. It is important to notice, though, that such savings are significant if one of the dimensions of \mathbf{X} is much larger than the others (Björck, 1996; Paatero, 1997). The fill relative to \mathbf{J}_{BC} as the QR decomposition progresses, makes the use of sparse QR function impractical in most of the other cases: after \mathbf{J}_A part has been treated, \mathbf{J}_{BC} is full, sparse storage is consequently less effective and bears no advantage in terms of number of operations. For this reason, the artificial data sets employed in the experimental part are not designed to test this aspect and the implementation of a block angular QR algorithm specific for PARAFAC is left for future developments. Hence, problem (17) is solved by means of the system of normal equations (19) using the Matlab (www.themathworks.com) built-in (full) Cholesky decomposition and back-substitution. Since \mathbf{J} is typically a very thin and tall matrix, this solution is particularly advantageous. Furthermore, if no weights are required, the Jacobian \mathbf{J} need not be formed and both $\mathbf{J}^T \mathbf{J}$ and $\mathbf{J}^T \mathbf{r}$ can be computed directly without recurring to sparse matrices (Liu and Sidiropoulos, 2001), whose creation and update are rather slow.

2.3.1. dGN

The algorithm as outlined in the previous section is not sufficient to yield a globally convergent algorithm for fitting the PARAFAC model, which is intrinsically indeterminate with respect to factor scaling (cf. Appendix A). This phenomenon leads to rank deficiency of the Jacobian, which has always at least $2F$ zero singular values, and consequently to singularity of $\tilde{\mathbf{H}}$, a problem that can be successfully handled by the damped Gauss Newton (dGN) algorithm devised by Levenberg and Marquadt (Björck, 1996; Levenberg, 1944; Marquadt, 1963). In dGN, the update $\Delta \mathbf{p}^{(s)}$ is calculated from the modified normal equations:

$$\left(\tilde{\mathbf{H}} + \lambda^{(s)} \mathbf{I}_N \right) \Delta \mathbf{p}^{(s)} = -\mathbf{g}, \quad (19)$$

which is equivalent to solving problem (17) under the constraint:

$$\|\Delta \mathbf{p}^{(s)}\|_2^2 \leq \delta \left(\lambda^{(s)} \right). \quad (20)$$

That is to say, the minimum is sought within a region (the trust region) of radius $\delta \left(\lambda^{(s)} \right)$ (Madsen et al., 2004). If the damping parameter $\lambda^{(s)}$ is large enough compared to the singular values of $\tilde{\mathbf{H}}$, the matrix $(\tilde{\mathbf{H}} + \lambda \mathbf{I}_N)$ is non-singular and system (19) can be efficiently solved (Björck, 1996; Madsen et al., 2004).

The updating strategy for $\lambda^{(s)}$ is crucial for the efficiency of the algorithm. The scheme used in the implemented dGN algorithm is thoroughly described in Madsen et al. (2004) and is based on the ratio between the actual variation of the loss function (ΔL) and the linear decrease $\Delta \tilde{L} = \tilde{L}(\mathbf{0}) - \tilde{L}(\Delta \mathbf{p})$. E.g., if this ratio is less than 0 (which, if $\Delta \mathbf{p}^{(s)}$ is a descent direction, implies $\Delta L < 0$), the step is rejected and $\lambda^{(s)}$ is increased. The idea behind such updating scheme is that if the ratio is small, the linear approximation does not hold and thus the region must be shrunk. Vice versa, if the ratio is very large the linear model holds and the region can be enlarged. This strategy allows dealing effectively with non-linearities and, as can be seen from Eq. (20), the damping parameter can determine the shortening or the elongation of the step by acting on the trust region size. Thus, it produces results similar to those expected by a line search procedure, which consequently was not implemented in the dGN algorithm.

The intrinsic scaling indeterminacy poses another problem that may affect the numerical stability of the algorithm: if the conventional scaling used for the PARAFAC model is applied (i.e. $\|\mathbf{b}_f\| = \|\mathbf{c}_f\| = 1$), the values of the norms of the columns of \mathbf{J} may differ by several orders of magnitude (typically $a_{if}b_{jf} \cong a_{if}c_{kf} \gg b_{jf}c_{kf}$) depending on the data. Such problem is reflected in the practical condition number of \mathbf{J} (i.e. computed disregarding the last $2F$ zero singular values (Tomasi and Bro, 2004)) and thus in the accuracy of the solution of the system of normal equations and can be avoided by employing a different scaling convention (see Appendix A), whereby the different loading vectors for the same factor are given the same norm (i.e. $\|\mathbf{a}_f\| = \|\mathbf{b}_f\| = \|\mathbf{c}_f\|$). The rescaling is performed at each step as it represents a small additional overhead compared to other steps of the algorithm.

With specific reference to memory consumption, dGN is the most expensive method together with PMF3 (cf. the next section). It requires, besides the original array $\underline{\mathbf{X}}$, three vectors of size $M \times 1$, $N \times 1$ and $N \times 1$ for respectively \mathbf{r} , \mathbf{g} and \mathbf{p} , and two $NF \times NF$ matrices, one for $\tilde{\mathbf{H}}$ and one for the Cholesky factor of $(\tilde{\mathbf{H}} + \lambda \mathbf{I}_N)$. Both the matrices are almost full and are treated as such (Paatero, 1997).

2.3.2. PMF3

Paatero (1997) proposed several modifications to dGN in the PMF3 algorithm: the presence of a regularisation factor and a very specific non-linear update. Besides, PMF3 employs a line search procedure that is called whenever the algorithm diverges. In its original form, PMF3 also includes a weighted least squares loss function and possible non-negativity constraints on the parameters, but these are of no concern in this work. Since they can both be

safely removed from PMF3, they will be ignored in the brief description given here as well as in the simulations.

The loss function of PMF3 differs from (13) by the presence of a regularisation term:

$$L_{\text{PMF3}}(\mathbf{p}) = \sum_{m=1}^M r_m^2(\mathbf{p}) + \gamma \sum_{n=1}^N \widehat{p}_n^2 = \mathbf{r}(\mathbf{p})^T \mathbf{r}(\mathbf{p}) + \gamma \widehat{\mathbf{p}}^T \widehat{\mathbf{p}}, \quad (21)$$

where $\widehat{p}_n = (p_n - p_n^\bullet)$ and p_n^\bullet is the target value (0 in this case) for the p_n parameter. γ is a coefficient that varies depending on the problem at hand. Eq. (21) leads to the following modified system normal equations (see Appendix B):

$$\tilde{\mathbf{H}} \Delta \mathbf{p}'^{(s)} \equiv \left(\tilde{\mathbf{H}} + \left(\lambda^{(s)} + \gamma^{(s)} \right) \mathbf{I}_N \right) \Delta \mathbf{p}'^{(s)} = -\mathbf{J}^T \mathbf{r} + \gamma^{(s)} \widehat{\mathbf{p}}^{(s)}, \quad (22)$$

which is again solved for $\Delta \mathbf{p}'^{(s)}$ (the apex is to distinguish this update from the non-linear one described below).

As $\mathbf{p}^\bullet = \mathbf{0}$, the regularisation term is a scalar γ times the norm of the vector \mathbf{p} , and thus penalises high absolute values for the parameters. The goal of this procedure is to correct the loss function for the scaling indeterminacy as well as non-identifiability (Paatero, 1997). Paatero also suggests equating the influence of the different slabs in the penalty term by using different γ_s for each estimated parameter when there are essential scale differences between the slabs of array \mathbf{X} . The scaling convention illustrated in Section 2.3.1 attains a similar result in the sense that the three modes have exactly the same weight in the penalty, thus one γ was deemed sufficient.

The update strategy for γ followed the scheme suggested in Paatero (1997): γ is decreased by a factor γ_{update} when the loss function decreased (in relative terms) by less than a certain threshold for at least γ_{it} consecutive iterations. No indications are given in Paatero (1997) about the magnitude of γ (which is problem dependent) or the spacing of the thresholds. An initial value for γ that seemed reasonable and that heuristically appeared to be good for the analysed problems was 1. By similar means, five thresholds were set so that their base 10 logarithms were equally spaced between -1 and the \log_{10} of five times the convergence criterion in terms of relative fit (e.g. -5.3 for a 10^{-6} criterion).

The non-linear update $\Delta \mathbf{p}''$ is computed by solving the system:

$$\tilde{\mathbf{H}} \Delta \mathbf{p}''^{(s)} = -\mathbf{J}(\mathbf{p}')^T \mathbf{r}(\mathbf{p}') + \gamma^{(s)} \widehat{\mathbf{p}}^{(s)}, \quad (23)$$

where $\mathbf{p}' = \mathbf{p}^{(s)} + 0.5 \Delta \mathbf{p}'^{(s)}$. The final update is chosen as the one between $\Delta \mathbf{p}'^{(s)}$ and $\Delta \mathbf{p}''^{(s)}$ that provides the largest reduction in the loss function. The main reason for this rather complicated procedure seems to be avoiding to compute the full Hessian (which may not be positive definite (Madsen et al., 2004)) and still obtain an update that may be a better approximation of the Newton direction, which again is more appropriate than the dGN step in case of e.g. large residuals (Madsen et al., 2004). Furthermore, the Cholesky factor of $\tilde{\mathbf{H}}$ need not be recomputed to solve (23), which allows for some savings in the number of operations, and $\mathbf{J}(\mathbf{p}')^T \mathbf{r}(\mathbf{p}')$ can be efficiently determined without having to compute \mathbf{J} (Liu and Sidiropoulos, 2001). However, no reference or reason are given for the 0.5 coefficient to obtain \mathbf{p}' .

The actual implementation of PMF3 employs a soft line search procedure (described in Section 2.4) for both the linear and the non-linear update whenever the algorithm diverges. In the original algorithm, should the line search procedure fail to retrieve a step-length yielding a reduction of L_{PMF3} , a special step optimising one parameter at a time would follow. The strategy followed in the implementation of PMF3 used for the tests is slightly different and if 10 iterations of the line search fail, the step is rejected and the damping parameter is increased using the standard dGN procedure.

The memory requirements of PMF3 are the same as for the dGN algorithm.

2.4. Line search

Line search procedures are applicable for all iterative methods with a well-defined loss function. The reasons and the implementation vary depending on the algorithm. In PARAFAC-ALS, line search is employed solely to speed up the algorithm. After the s -th iteration, the variation of the three loading matrices with respect to iteration $s - 1$ is calculated (e.g. $\Delta\mathbf{A} = \mathbf{A}^{(s)} - \mathbf{A}^{(s-1)}$) and is used to linearly predict the corresponding matrix d iterations ahead (e.g. $\mathbf{A}^{(s+d)} = \mathbf{A}^{(s)} + d\Delta\mathbf{A}$), where d is determined empirically as a function of the number of iterations (Bro, 1998).

In the case of derivative based methods, this procedure copes with non-linearities of the loss function and with high residuals (Madsen et al., 2004). The basic idea is to consider the update $\Delta\mathbf{p}^{(s)}$ as a direction along which a step of length $\alpha^{(s)}$ is taken; the step length can be found as a solution to the univariate minimisation problem

$$\min_{\alpha} \tilde{L}(\mathbf{p} + \alpha\Delta\mathbf{p}) = \min_{\alpha} L'(\alpha). \quad (24)$$

The search for the solution minimiser α^* , which need not be determined with great accuracy (Madsen et al., 2004), is performed in the implemented PMF3 algorithm by a simple procedure interpolating a quadratic model on at least three values of $L'(\alpha)$ calculated at $\alpha = 0, 0.5$ and 1. If these three initial points bracket a minimum there are sufficient conditions for α^* to be estimated; otherwise new points are added until a minimum is bracketed.

2.5. Compression

The main problem with derivative based methods is that, even though they typically require fewer iterations than ALS, the number of operations per step and the memory requirements are significantly higher, up to the point of being inapplicable for larger arrays (Paatero, 1997). A possible solution is to reduce the dimensions of the array on which the Gauss–Newton method is used. The compression based on the Tucker3 model fulfills this task (Bro and Andersson, 1998; Kiers, 1998). Although it introduces additional complexity in the algorithm, this part eventually provides a large reduction in the number of operations. Furthermore, it needs only be done once, whereas several alternative PARAFAC models are usually fitted to find the most feasible one. Once the PARAFAC model has been calculated on the core extracted by the Tucker3 algorithm, the solution can be expanded to its original dimensions providing very good starting values for the PARAFAC-ALS standard algorithm, which is used only to refine it.

The compression of the array $\underline{\mathbf{X}}$ is based on the search for three truncated bases spanning the variation in each mode and the projection of $\underline{\mathbf{X}}$ on them. Calling these bases \mathbf{U} ($I \times L$), \mathbf{V} ($J \times M$) and \mathbf{Z} ($K \times N$) for the first, the second and the third mode respectively, the loadings of the PARAFAC model can be expressed as

$$\mathbf{A} = \mathbf{U}\mathbf{P}, \quad (25a)$$

$$\mathbf{B} = \mathbf{V}\mathbf{Q} \quad (25b)$$

and

$$\mathbf{C} = \mathbf{Z}\mathbf{R}, \quad (25c)$$

where \mathbf{P} is a $L \times F$ matrix, \mathbf{Q} is a $M \times F$ and \mathbf{R} is $N \times F$. The CANDELINC theorem (Carroll et al., 1980) states that, if a PARAFAC model is sought under the linear constraints expressed in (25a–c), it is only necessary to estimate \mathbf{P} , \mathbf{Q} and \mathbf{R} , which can be done by fitting a PARAFAC model to the array $\underline{\mathbf{G}}$ calculated as

$$\mathbf{G}^{(L \times M \times N)} = \mathbf{U}^T \mathbf{X}^{(I \times J \times K)} (\mathbf{Z} \otimes \mathbf{V}). \quad (26)$$

Typically $\underline{\mathbf{G}}$ is much smaller than $\underline{\mathbf{X}}$ due to the redundancy of information in the latter. The three bases and $\underline{\mathbf{G}}$ are found by applying the Tucker3 model, which is both fast and efficient, to $\underline{\mathbf{X}}$. The $\underline{\mathbf{G}}$ array is the core produced by Tucker3 while the bases are the three loading matrices.

For this application, it is not necessary to calculate an exact Tucker3 model because the compressed array is only an approximation of the original one and the final model would need to be refined anyhow. Hence, several simplifications are utilised to minimise computational cost of this step: first of all the number of iterations for Tucker3 is limited both by fixing a relatively small maximum number of allowed iterations and by increasing the convergence criterion. Tucker3 is initialised by using orthogonal matrices of random numbers. In order to balance the lower fit deriving from the low number of iterations, the number of components extracted in each mode (that is the number of vectors forming the truncated bases) is slightly increased; it has been observed empirically that a reasonable choice is equal to the number of factors sought by PARAFAC plus 2 (of course upper limited by the dimensions of the array) (Andersson and Bro, 2000; Bro, 1998).

With respect to memory consumption, the expressions given in Sections 2.2.1 and 2.3 are applicable here, with the sole difference that the array upon which the algorithm is applied has dimensions $(F + 2) \times (F + 2) \times (F + 2)$.

3. Experimental part

Nine algorithms are studied in this section using both simulated and real data: DTLD-GRAM, PARAFAC-ALS, ASD, SWATLD, dGN, PMF3, PARAFAC-ALS with compression, dGN with compression and PMF3 with compression. For space reasons the single implementations will not be discussed in detail; in general, the MATLAB 6.5 (www.themathworks.com) guidelines for improving performances were followed (e.g. maximising the use of built-in functions). DTLD and PARAFAC-ALS are part of the N-Way

Table 1
Design factors and levels for the artificial data sets

Rank		3,5
Noise ^a	Homoscedastic	1,5,10
	Heteroscedastic	0,1,5
Congruence		0.5,0.9

^aPercent of $\|\mathbf{X}^{(I \times JK)}\|_F^2$ (i.e. % of total variation).

Table 2
Condition number for **A**, **B** and **C** in the artificial data sets

	Congruence		
		0.5	0.9
Rank	3	2	5.29
	5	2.44	6.78

toolbox (Andersson and Bro, 2000). ASD was implemented according to Faber et al. (2003). The other m-files are available for download at www.models.kvl.dk.

3.1. Simulated data

Simulated data sets have the advantage that important features can be controlled so that their effect on the different methods may be assessed. Hence, a Monte Carlo study has been carried out based on 720 arrays of dimension $20 \times 20 \times 20$ considering the following aspects: rank of the systematic part of the multi-way array (F), congruence of the true components (see Appendix C) and amount and type of noise (Table 1). For each combination of these conditions 20 replicates were generated to counterpoise minor statistical fluctuations. The final number of data sets amounted to 2 ranks \times 3 levels of homoscedastic noise \times 3 levels of heteroscedastic noise \times 2 congruences \times 20 replicates, which is again equal to 720.

As to provide the proper collinearity, the components were generated according to the following scheme, applied to all the required loading matrices (Kiers et al., 1999):

- The Cholesky factor **R** is calculated of an $F \times F$ (where F is the number of factors) matrix having ones on the diagonal and the desired values of congruence between the loading vectors as off-diagonal elements.
- A column-wise orthogonal matrix **U** of size $I \times F$ (where I here represents the dimension in the mode the loading matrix refers to) is generated.
- The loading matrix is then defined as $\mathbf{L} = \mathbf{UR}$.

One thing to notice is that for all the data sets having the same number of underlying components and equal congruence, the condition number of the loadings matrices is the same (see Table 2). In particular, according to Kiers (1998), for congruence 0.5 the level

of collinearity is relatively low while for congruence 0.9 the underlying factors are only mildly collinear. With the employed method, higher values for the condition number are only attainable by increasing the congruence among the factors to an extremely high value (viz. to yield a condition number of 11, the average for severely collinear set in Kiers (1998), the congruence has to be 0.9956) or by reducing the numerical relevance of one or more of the factors. The first option was ruled out because such high values of congruence are seldom found in practice, while the second option was avoided to limit the number of design factors in the simulation.

Homoscedastic and heteroscedastic noise were generated from random identically (and normally) distributed numbers and two distinct arrays were added to the noise-less array $\mathbf{X} : \mathbf{R}_{ho}$ for the homoscedastic part and \mathbf{R}_{he} for the heteroscedastic one. In order to make the latter proportional, each element of \mathbf{R}_{he} was multiplied by the corresponding element in \mathbf{X} . Both \mathbf{R}_{ho} and \mathbf{R}_{he} were also normalised to a Frobenius norm of $\|\mathbf{X}^{(I \times JK)}\|_F$ and multiplied by two suitable scalars depending on the desired level of noise (see Appendix D).

Finally, for each experimental set up two models were fitted, extracting F or $F + 1$ factors. The aim was to acquire information on the stability of the methods when the wrong number of components is extracted. A total of 1440 models were fitted using each algorithm.

3.2. Real data sets

The algorithms were tested on two real data sets both constituted by fluorescence measurements:

1. Twenty-two solutions containing different amounts of four compounds (DOPA, hydroquinone, tryptophan and phenylalanine) were analysed on a Perkin-Elmer LS50 B spectrofluorometer. The excitation wavelengths ranged between 245 and 305 nm, with a step of 5 nm while in the emission mode the interval was: 260–390 nm with a step of 1 nm (Baunsgaard et al., 2000). The Rayleigh scatter was removed by subtracting from each sample a model for the scatter. Three replicates from each sample were then generated by excluding every third wavelength in the emission mode, i.e replicate one the wavelengths 260, 263 nm, etc. were removed, in replicate two the wavelengths 261, 264 nm, etc. were kept out while in the third the left out wavelengths started at 262 nm again with 3 nm increments. Hence, the data set consisted of three different arrays having dimensions $22 \times 87 \times 13$, for the first two replicates, and $22 \times 88 \times 13$ for the last.
2. Fifteen solutions containing DOPA, hydroquinone, tyrosine and tryptophan in different amounts were examined using a Cary Eclipse spectrofluorometer. The excitation wavelengths ranged between 230 and 300 nm measured at intervals of 5 nm, the emission was measured at 282–412 nm with 2 nm steps and the scatter was removed by subtracting a blank from each sample. Six replicates of each solution were analysed leading to six different arrays of dimensions $15 \times 66 \times 15$.

In both data sets, the different replicates have been modelled separately to give different realisations of the same model. As for the simulated data, the effect of over-factoring has been

Table 3

Condition numbers for the loading matrices and congruences between factors

	Condition number			Congruence (min – max)		
	A	B	C	A	B	C
Data set I	2.83	5.96	20.44	0.31–0.49	0.002–0.88	0.15–0.94
Data set II	8.99	12.24	8.07	0.46–0.51	0.15–0.86	0.54–0.94

Table 4

Tested algorithms and relative parameters for the simulated data

Algorithm ^a	Convergence criterion	Value	Max <i>n.</i> iterations	Other parameters
DTLD	—	—	—	—
ALS ^b , ALS _c , SWA	Relative fit	10 ^{−6}	10000	—
ASD	Relative fit	10 ^{−6}	10000	$\lambda = 10^{-3}$
dGN, dGN _c	Relative fit	10 ^{−6}	1000	$\lambda^{(0)} = \max(\text{diag}(\hat{\mathbf{H}}^{(0)}))^c$
	Gradient	10 ^{−9}		
PMF3, PMF3 _c	Relative fit	10 ^{−6}	1000	$\lambda^{(0)} = \max(\text{diag}(\hat{\mathbf{H}}^{(0)}))^c$
	Gradient	10 ^{−9}		$\gamma = 1, \gamma_{\text{update}} = 5, \gamma_{\text{it}} = 3$

^aDTLD: DTLD-GRAM; ALS: PARAFAC-ALS; SWA: SWATLD; ALS_c: PARAFAC-ALS with compression; dGN_c: damped Gauss–Newton with compression; PMF3_c: PFM3 with compression.

^bCovers also the refining step for ALS_c, dGN_c and PMF3_c.

^c $\text{diag}(\mathbf{H}) = [h_{11} \ h_{22} \ \dots \ h_{NN}]^T$ for an $N \times N$ matrix.

analysed by fitting both 4 (the expected rank, represented by the number of constituents) and 5 components.

Table 3 shows congruence and condition numbers of the estimated loading matrices when 4 components are extracted; as it can be seen, both data sets are mildly collinear according to Kiers (1998). In fact, data set II is more problematic than data set I if the evaluation is based on the Jacobian's practical condition number (Tomasi and Bro, 2004), hence likely giving rise to greater numerical problems.

3.3. Initialisation and conventions

Initial estimations for the parameters were necessary for all the algorithms but DTLD-GRAM and were obtained by running five standard PARAFAC-ALS iterations (the 'relative fit decrease' convergence criterion set to 10^{−4}) starting with ten different sets of matrices of uniformly distributed values and then choosing the best fitting set to continue the process with the main algorithm.

The settings for the different tuning parameters, namely convergence criteria, λ s and γ s, are shown in Tables 4 and 5 for respectively the artificial and the real data. The different choices were due to the number of models that needed to be calculated. Hence, for the real data sets it was possible to make the convergence criteria stricter without making the

Table 5

Modified convergence criteria for the real data sets

	Convergence criterion		Iterations
ALS ^a , SWA, ASD, ALS _c	Relative fit	10^{-8}	20000
dGN, PMF3,	Relative fit	10^{-8}	1000
dGN _c , PMF3 _c	Gradient	10^{-10}	

For dGN, PMF3 and ALS the criteria are applied also when the method is used on the compressed array.

^aCovers also to the refining step for ALS_c, dGN_c and PMF3_c.

calculation time intolerably long.

With respect to the scaling indeterminacy the components were scaled and given a sign according to the convention described in Appendix A: the loading vectors belonging to the same component were scaled so that their norm was the same and the sign of the loading vectors in **B** and **C** was chosen so that the sign of the sum of the elements for each factor was positive.

3.4. Criteria of interest

The initial assessment regarded the quality of the solution. In this respect, four different parameters have been considered: value of the loss function, occurrence of full recoveries (defined below) and of 2FDs and mean squared error (MSE) of the parameter estimates. For the real data sets, the concentration of the different constituents was available and the Root Mean Squared Error in Calibration (RMSEC—cf. Appendix D) relative to predicting the concentrations from the first mode loadings (scores) using linear regression was used as an additional evaluation parameter.

Due to the large number of models, a graphical comparison and control was not feasible. Therefore, the quality of the factor estimates was assessed numerically in terms of congruence between the known underlying factors and the extracted factors (Mitchell and Burdick, 2004; Tomasi and Bro, 2004). Specifically, one can consider a factor as completely recovered if there is one component in the estimated model having a congruence greater than a certain ‘recovery threshold’ (close to 1). Congruence tends to be quite an optimistic method and a limit lower than 0.99 for the single loading vector may not be good enough for automated procedures. Besides, the aim of this threshold was to provide a common comparison ground for the different algorithms and thus a value of 0.99, which ensures that the full recovery occurred, was preferred. For the single component the threshold was thus set to $0.99^3 \simeq 0.97$ (Tomasi and Bro, 2004).

Due to the permutation indeterminacy (Bro, 1998), the order of the factors in the three loading matrices is not determined beforehand, but comes as a consequence of the initial estimations provided to the algorithm. All the possible permutations of extracted factors (up to F) were compared with the correct loading matrices and the final choice (the ‘winning permutation’, represented by the permutation matrix **P**) was the one associated to the highest sum of the congruences for all the original factors (Mitchell and Burdick, 1994). Finally, full recovery occurs if all the F congruences in the latter permutation were greater than the

recovery threshold.

The occurrence of a 2FD is established when the congruence between two components within the same solution is less than or equal to -0.8 .

The MSE can be calculated separately for each of the loading matrices and is an indication of the distance between the extracted factors and the actual ones. For \mathbf{A} , MSE can be calculated as

$$\text{MSE}(\mathbf{A}, \hat{\mathbf{A}}, \mathbf{P}, \mathbf{S}_\mathbf{A}) = \frac{\|\mathbf{A} - \hat{\mathbf{A}}\mathbf{P}\mathbf{S}_\mathbf{A}\|_F^2}{IF}, \quad (27)$$

where \mathbf{A} refers to the underlying factors and has been adequately scaled (see Appendix A), $\hat{\mathbf{A}}$ is the extracted $I \times F$ (or $I \times F + 1$) loading matrix, and $\mathbf{S}_\mathbf{A}$ is an $F \times F$ scaling (diagonal) matrix solving the problem (Riu and Bro, 2003):

$$\begin{aligned} \min_{\mathbf{S}_\mathbf{A}, \mathbf{S}_\mathbf{B}, \mathbf{S}_\mathbf{C}} L(\mathbf{A}, \mathbf{B}, \mathbf{C}, \hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}, \mathbf{P}, \mathbf{S}_\mathbf{A}, \mathbf{S}_\mathbf{B}, \mathbf{S}_\mathbf{C}) \\ = \|\mathbf{A} - \hat{\mathbf{A}}\mathbf{P}\mathbf{S}_\mathbf{A}\|_F + \|\mathbf{B} - \hat{\mathbf{B}}\mathbf{P}\mathbf{S}_\mathbf{B}\|_F + \|\mathbf{C} - \hat{\mathbf{C}}\mathbf{P}\mathbf{S}_\mathbf{C}\|_F, \\ \text{subject to } \mathbf{S}_\mathbf{A}\mathbf{S}_\mathbf{B}\mathbf{S}_\mathbf{C} = \mathbf{I}_F \end{aligned} \quad (28)$$

In case of over-factoring, \mathbf{P} is represented by the first F columns of the $F + 1 \times F + 1$ winning permutation matrix (thus \mathbf{P} is $F \times F$ when F factors are extracted and $F + 1 \times F$, if a $F + 1$ components model is fitted). It has been observed that, if the variation captured by each factor is equally spread over the three corresponding loading vectors, the MSEs for the different loading matrices are similar in magnitude. For simplicity the MSE is presented as an average among the MSEs of \mathbf{A} , \mathbf{B} and \mathbf{C} .

The second main aspect considered in the tests was the computational efficiency, which is relatively complex to define. In general, one may say that the best method is the one that produces the solution using as few floating point operations (FLOPs) as possible. Nevertheless, with the advent of vector machines and the implementation of block algorithms, this parameter is inadequate to give a clear picture; other aspects such as memory traffic and data access are in fact equally important and very machine dependent (Golub and Van Loan, 1996). As mentioned, the algorithms were implemented as to yield (to the authors' knowledge) optimal performances in terms of time consumption under MatLab 6.5 and used only built-in functions. All the tests were performed on the same dedicated machine (mounting a 2.6 GHz Pentium IV processor, 512 MB of memory and running on WindowsXP) and the attention was concentrated on the time consumption and the number of iterations necessary to reach convergence.

3.5. Results and discussion

3.5.1. Simulated data

The first aspect to be considered was recovery capability (factor congruence). Overall, all the iterative algorithms managed to retrieve the correct solution (with a congruence threshold of 0.97) in more than 50% of the cases (Table 6). The best algorithm in this respect is SWATLD, with 63.6% of correct solutions, followed by damped Gauss–Newton

Table 6

Percentage of full recoveries (threshold 0.97) for the different algorithms on the simulated data sets

Rank	3				5				Global
Congruence	0.5		0.9		0.5		0.9		
Extr. Factors	3	4	3	4	5	6	5	6	
DTLD	89	68	8	7	23	13	0	0	26.1
ASD	100	79	19	12	99	87	4	1	50.1
SWA	100	100	46	40	99	99	12	12	63.6
ALS	100	59	51	36	99	70	12	11	54.7
dGN	100	82	52	41	99	86	13	11	60.5
PMF3	100	91	52	40	99	91	13	13	62.4
ALSc	100	64	51	39	99	76	12	12	56.8
dGNc	100	94	50	42	99	92	12	12	62.7
PMF3c	100	90	49	42	99	91	12	13	62.0

All the noise levels have are considered. The last column (Global) reports the percentage of full recoveries for each algorithm over the 1440 models of the Monte Carlo simulations.

with compression (62.7%) and PMF3 (62.4%). DTLD, the only direct method, attained the correct solution in only 26.1% of the cases and yielded the worst performance, followed by ASD with 50.1%. ALS falls in the lower part of the range with 54.7% full recoveries.

While DTLD did consistently worse than the other algorithms in all setups, the performances of ALS were equivalent to those of the derivative based algorithms and SWATLD when the rank was correctly estimated (Table 6). Conversely, ALS was heavily affected by an incorrect estimation of the rank and this sensitivity was reduced, but not eliminated, by compression. It can be seen from both Table 6 and Fig. 1 that the most robust method in this respect is SWATLD, whose performance is remarkably good: when the rank was overestimated, SWATLD performed equally or better than when rank was correctly set in 30 out of the 36 setups while for example ALS did worse in 21 setups in case of over-factoring. Also the derivative based methods seem considerably more robust than PARAFAC-ALS with respect to over-factoring. This is true also when the algorithms are applied together with compression, even though ALS' performance improves considerably.

The higher resistance of the derivative based methods to over-factoring is likely to be associated to the regularisation terms (both the damping parameter λ and the PMF3 regularisation term). A link has been established between rank of the array, rank of the model, occurrence of 2FDs (Krijnen, 1993; Kruskal et al., 1989; Paatero, 2000) and swamps, which are characterised by 2FDs and may lead to a premature stop of an algorithm. Regularisation may help in dealing with 2FDs (Paatero, 2000; Rayens and Mitchell, 1997) and thus it seems reasonable that it operates favourably when the rank is overestimated. 2FDs (with a congruence threshold of -0.8) could be observed considerably more often when the rank is overestimated (e.g. 55 cases against 4 for ALS for overestimated and correct rank and 125 against 15 for dGN). The sole exception in this sense was SWATLD, which did not yield any 2FD in any of the models, regardless of over-factoring. This may help explaining the efficiency of the latter method with respect to over-factoring, and that avoiding 2FDs during the early stages of the fitting procedure may significantly help retrieving the correct

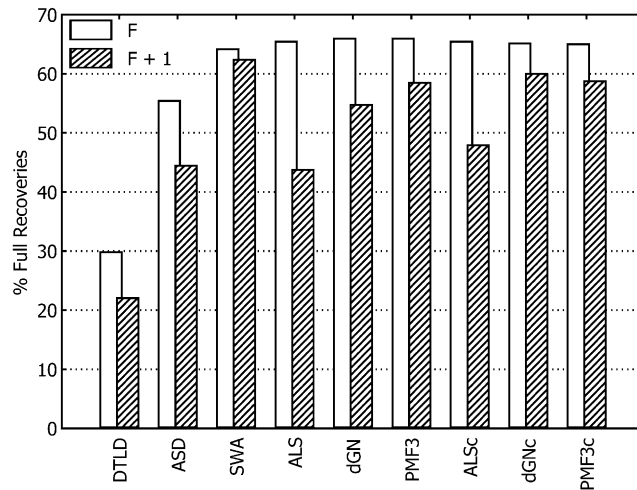


Fig. 1. Recovery (threshold 0.97) with correct number of factors (F) and in case of over-factoring ($F + 1$). All the conditions and replicates have been considered.

solution. The SWATLD solutions typically have a higher core consistency (Bro and Kiers, 2003) than the corresponding least squares solution (not shown), i.e. they seem to be biased towards describing only the strictly trilinear contributions and are likely less affected by possible interactions between factors that may emerge during convergence (e.g. the fact that part of the variation is associated to a factor is described by another). Nonetheless, it is difficult to establish what makes SWATLD so resistant to 2FDs and over-factoring. The fact that the algorithm uses three different loss functions and that its actual optimisation criterion is not known makes any conclusion in this direction rather inappropriate. Further analyses will be necessary to understand these reasons and will be the subject for future research. With respect to 2FDs, ALS yielded fewer in the final solution than any other method except SWATLD (4.1% of all models). The worst algorithm was ASD with 11.0% and this clearly contrasts with the claim of the original authors on the capability of this algorithm of avoiding 2FDs (Jiang et al., 2000). At the same time, PMF3 yields slightly fewer 2FDs than dGN (8.7% and 8.8% respectively—without compression) and compression increases their occurrence in all the methods to which it was applied (6.6% for ALSc, 9.7% for dGNc and 9.0% for PMF3c). The number of 2FDs for the least squares algorithms seems to follow the performances in terms of recovery capability, which suggests that the number of degeneracies is simply the consequence of the higher efficiency of some methods. Setting a smaller threshold to establish degeneracy (viz. -0.5) more or less equalises the number of 2FDs for ALS and the derivative based methods to approximately 20% of all models and seems to confirm that 2FDs become generally worse for dGN and PMF3. The small difference between dGN and PMF3 can be explained as an effect of regularisation.

The effect of the compression step on ALS, dGN and PMF3 is rather different. Particularly, the recovery capability of ALS and dGN visibly improves in case of over-factoring (plus 4.2% and 5.3% respectively when all ranks and degrees of collinearity are consid-

ered) remaining approximately the same (unchanged for ALS and -0.8% for dGN) when the rank is correctly set. On the contrary, PMF3 recovery capability remains largely unaltered (-1% and 0.3% respectively for correct rank and over-factoring). While the beneficial effects of compression may be explained in terms of removal of noise and spurious information, the different behaviour of PMF3 compared to dGN is not clearly understood yet. There are several differences between the two algorithms. In particular, dGN implements a more refined damping parameter's updating scheme (Madsen et al., 2004) than PMF3 (Paatero, 1997), whereas the second employs soft line search and a regularisation factor that operates in a similar fashion as the damping parameter, possibly supplementing to the shortcomings of its simple trust region implementation. In fact, preliminary additional results show that using the λ updating scheme of dGN for PMF3 yields results similar to those of dGN, with a reduction in the number of full recoveries for PMF3 and better results for PMF3c.

Collinearity between the underlying factors is the feature that most affects the number of full recoveries. For both ranks and when the number of components of the array is correctly set, a significant drop in the performance can be observed for all algorithms as collinearity increases (Table 6 and (Kiers, 1998)). It is particularly relevant that DTLD and ASD, which explicitly assume linear independence of loading vectors, are the most affected by collinearity. SWATLD, which works under similar basic assumptions, is affected to a much lesser extent and although visibly inferior to the least squares methods when the correct number of components is fit, it still performs slightly better than e.g. ALS in case of over-factoring. Again, the non-least squares criterion and the variety of ad-hoc solutions of this method make it difficult to explain this additional stability.

With respect to noise levels, the different algorithms (again with the exception of DTLD and ASD) are affected in a similar way (Fig. 2). An increase in the level of noise always leads to a worsening in the recovery capability when the rank is correctly set. This is partly due to lack of convergence to a meaningful solution, but also to the inclusion of some noise in the components that makes their congruence with the original factors smaller than the threshold (Kiers, 1998). Heteroscedastic noise has a greater effect than homoscedastic noise and considering the setups where the total amount of noise is the same (viz. $1 - 5/5 - 1$ and $5 - 5/10 - 0$), the recovery capability is worse when heteroscedastic noise is larger, dropping from 65% to 70% to approximately 50%. Conversely, an improvement could be observed for low or intermediate homoscedastic noise as heteroscedastic noise increased and the model rank exceeded the correct one. This was found true in the low collinearity case for all algorithms apart from DTLD. The improvement is consistent over the amount of heteroscedastic noise and is quite dramatic for ALS and ALS_c, with an increase from 30% of full recoveries at 1% homoscedastic and no heteroscedastic noise to 90% (80% without compression) at 5% heteroscedastic noise and 95% (with and without compression) when both types of noise were fixed at 5%. In the high congruence case, this behaviour could be observed at 1% homoscedastic and heteroscedastic noise, in which case the extraction of an additional component brought to an improvement between 5% (for dGN and PMF3) and 12.5% (ALS) in the occurrence of full recoveries. A possible explanation is that part of the noise is retained in the additional component and that a certain degree of systematic behaviour in the noise facilitates its extraction. At the same time, the additional factor prevents noise from altering the loadings, which explains the increased number of full

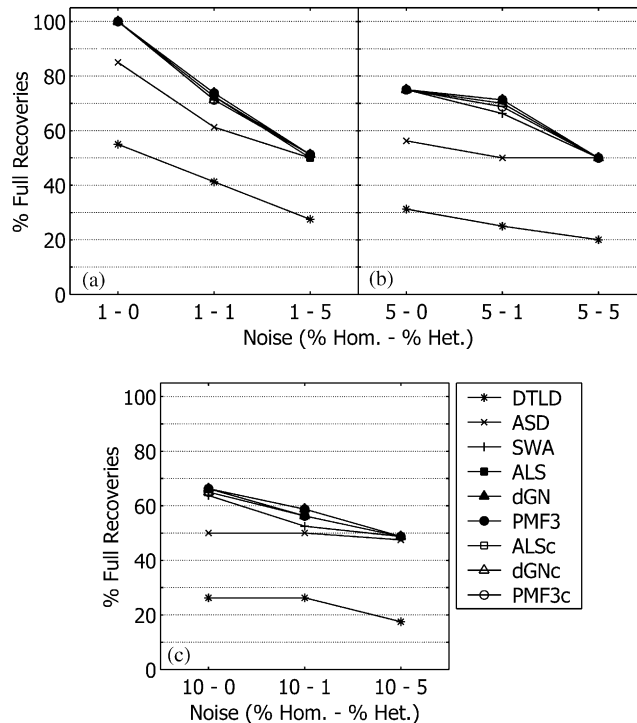


Fig. 2. Effect of noise on recovery capability (threshold 0.97) when the correct number of factors is extracted. All congruences and ranks are considered.

recoveries.

The second aspect on the quality of the solution to be analysed was how close the calculated solution was to the real underlying factors. Fig. 3 shows the median MSE for the case of correct model rank and low collinearity. The solution provided by the two non-least squares iterative algorithms (i.e. ASD and SWATLD) are noisier than the others. The use of the median penalises DTLD in the rank 5 case since this method does not yield the correct solution in at least 50% of cases. Nevertheless, it can still be seen that for rank 3 data sets characterised by low collinearity, when the performances in terms of recovery are sufficiently good, the MSE is still higher, albeit not by much, than for the least squares methods.

Similar conclusions can be drawn by studying the value of the loss function, or, more precisely, the occurrence of values of the loss function (6) that differ by less than 10^{-4} in relative terms from the minimum yielded by any of the algorithms (results not shown). Only in a negligible number of cases (0.4%, all of them in the low rank, low collinearity case with over-factoring) SWATLD reached the same minimum as the least squares methods. This is surprising considering the higher recovery capability of this algorithm, but confirms that this method does not find the least squares solution but rather one that is biased towards a higher core consistency. Analogous observations can be made for ASD, which never

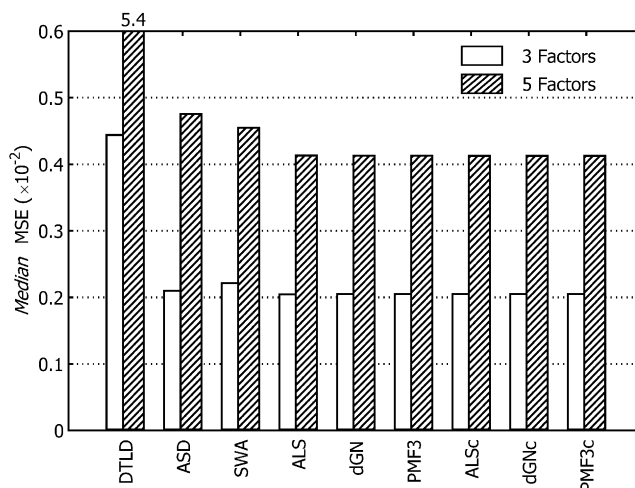


Fig. 3. Median MSE for the parameters estimates when the correct number of components is fitted. Only the low congruence case is depicted in the figure.

yields the lowest value of a least squares loss function. It is interesting to notice how the use of compression improves the results of all the algorithms in this respect. The most significant improvement is that of PARAFAC-ALS (plus 9.5% of attained minima), but also the two derivative methods (plus 8.5% and 6.0% respectively for dGN and PMF3) perform appreciably better.

The final assessment of the algorithms regarded their convergence rate and the computational costs expressed in terms of number of iterations and time consumption respectively. Although the number of iterations for the different methods varies significantly, particularly between alternating methods and derivative based ones (Paatero, 1997; Tomasi and Bro, 2004), it is still possible to assess the effect of the various design factors on convergence rate, the effect of compression on the various algorithms, and whether PMF3 (and correspondingly PMF3c) converges more rapidly than the standard dGN method (and dGNc).

All the considered design factors have an influence on the number of iterations in fitting, which increases with rank, congruence and over-factoring (Table 7), confirming previous results (Hopke et al., 1998; Kiers, 1998). With respect to noise, though, the behaviour is different and all the methods apart from SWATLD and PMF3 use fewer iterations when the rank is overestimated in at least 50% of the cases as heteroscedastic noise increase. The reason for this may be again that especially the heteroscedastic noise simplifies the fitting of an additional component holding the systematic part of the noise.

When using as initial values those found by fitting a model on the compressed space (Table 7), the number of PARAFAC-ALS full data iterations (FDIs (Kiers, 1998)) in 93–100% of the cases approximately 21.5% (median over all conditions). In particular, 2 FDIs were sufficient to refine the solution in 17.6% of the cases for PMF3c and in 19% for LMc, all but one for correctly estimated rank. The number of FDIs does not vary much depending on the algorithm used in fitting, and is identical to the minimum attained with any method

Table 7

Median number of iterations. All levels of noise are considered in each cell

Rank	3				5			
	0.5		0.9		0.5		0.9	
Extr. Factors	3	4	3	4	5	6	5	6
ASD	8	25	83	85	8	19	99	122
SWA	7	41	35	55	8	38	72	96
ALS	10	48	60	80	12	46	91	110
dGN	6	70	16	54	7	54	53	91
PMF3	7	63	17	51	7	55	47	85
ALSc	10 (2)	82 (8)	97 (5)	156 (16)	12 (2)	70 (10)	195 (28)	243 (34)
dGNc	7 (2)	75 (10)	17 (5)	61 (16)	7 (2)	67 (12)	54 (26)	87 (33)
PMF3c	8 (2)	51 (10)	18 (5)	50 (16)	8 (3)	47 (12)	47 (26)	78 (34)

The number in parenthesis for ALSc, dGNc and PMF3c indicates the number of PARAFAC-ALS iterations in the refinement step.

in 75.8% of the cases for PMF3c, 77.6% for dGNc and 79.7% for ALSc (not shown). This is particularly true for dGNc and PMF3c: the maximum difference in the number of FDIs for the refinement step is less than or equal to ten in 95.3% of the cases. With respect to the various design factors, the refinement step is affected by the same features as the PARAFAC-ALS algorithm, namely rank, congruence, noise and overfactoring, which determine an increase in the number of iterations necessary to attain convergence (results not shown).

The number of compressed data iterations (CDIs (Kiers, 1998)) for ALSc, dGNc and PMF3c depends on the varied conditions and is normally larger than the number of FDIs for the same algorithm (Table 7). However, whereas for ALSc the number of CDIs increases (or remains the same) in 89% of the cases compared to ALS, the same holds for the two derivative-based algorithms only when the rank is correctly set (84% and 81% for dGNc and PMF3c respectively). The opposite is true instead for PMF3, which requires fewer CDIs in case of over-factoring in more than one case out of two (i.e. 54.2%).

The fastest iterative methods were clearly ASD and SWATLD (Fig. 4). They required less time than any other iterative algorithm in 44.2% and 41.7% of all models respectively (not shown). ALS (13.6%) and dGN (0.5%) accounted for the remaining cases. As pointed out in Section 2.2.2, ASD employs compression by means of singular value decomposition, which explains the good performances of this method. On the other hand, as for the recovery capability, ASD is heavily affected by the collinearity of the underlying components and performs worse than SWATLD in more than one case out of two (Fig. 4) of the models with high congruence between the underlying factors. The 196 models (13.6%) for which ALS is the fastest method are more or less equally scattered between the various conditions, with a slight dominance for the high congruence cases (122 compared to 74).

In terms of time expense, ALSc was found slower than ALS 94.4% of the times (Table 8). The reason for this is that the cheaper iterations deriving from compression are counteracted for ALS by their increased number. On the contrary, the advantage of compression is apparent for the two derivative-based methods, which (without compression) are clearly the

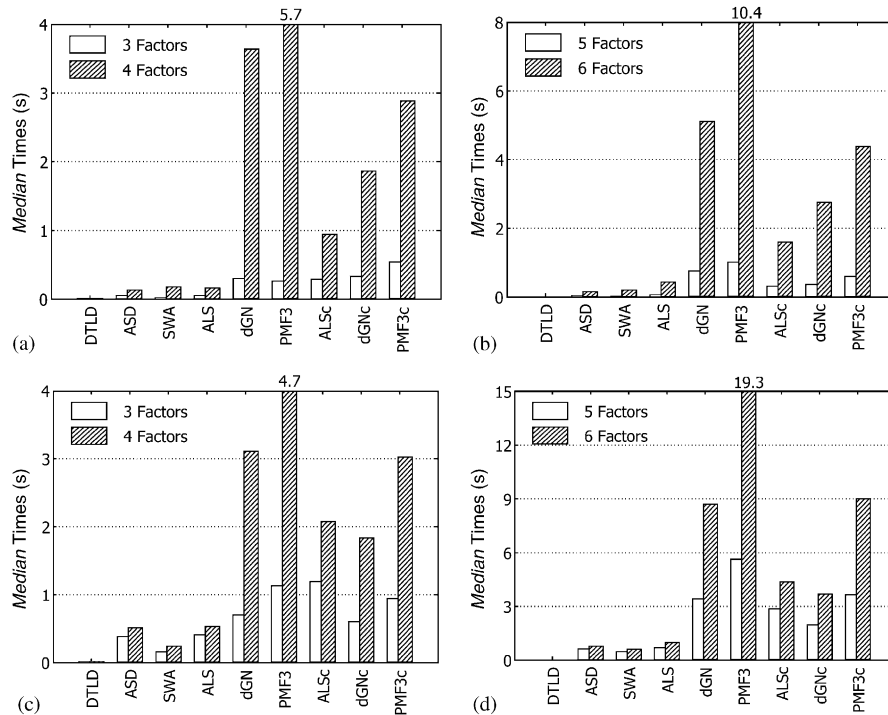


Fig. 4. Median time consumption. a,b) low congruence, c,d) high congruence case. All noise levels are considered.

most expensive ones (Fig. 4). Since compression makes the fitting step independent of the dimensions of \mathbf{X} , the bad performances of ALSc compared to ALS are likely caused by the small dimensions of the \mathbf{X} array used in the tests (Bro and Andersson, 1998; Kiers, 1998). However, ALSc is also slower than dGNc for 55% of the models and is only faster than dGN, PMF3, and PMF3c (Table 8). The advantage of using dGNc is particularly evident for the high congruence case (Fig. 4). Nevertheless, compression heightens the dependence of time expense on the rank of the fitted model and, for larger values of model's rank, ALSc might be faster than the derivative based methods owing to the longer time required by the Cholesky decomposition.

Finally, with respect to PMF3 and dGN, the latter required as many or fewer iterations in 59.1% of the cases when employed without compression and 55.5% if PMF3c and dGNc are considered (not shown). The ratio becomes even less favourable to PMF3 when the model's rank is known (82.5% and 81.1% for PMF3 and PMF3c respectively), but is inverted in case of over-factoring (29.7% and 35.7%). However, since the PMF3 iterations are more expensive, this algorithm was found to be slower in 94.1% of the cases (Table 8) and the difference in time consumption between the two algorithms is, in relative terms less than 10% only for 1.9% of the models.

3.5.2. Real data set I

Table 8
Comparison of time consumption for the different algorithms

	ASD	SWA	ALS	dGN	PMF3	ALSc	dGNc	PMF3c
ASD	—	49.5	72.8	92.2	96.9	94.4	90.6	94.8
SWA	49.7	—	72.3	97.5	99.7	97.1	97.3	99.0
ALS	27.2	27.7	—	89.0	98.2	94.4	84.9	93.3
dGN	7.7	2.4	11.0	—	94.1	43.1	30.6	58.5
PMF3	3.1	0.3	1.8	5.9	—	14.4	5.3	18.2
ALSc	5.6	2.9	5.6	56.9	85.6	—	44.9	71.7
dGNc	9.4	2.7	15.0	69.2	94.7	55.0	—	98.2
PMF3c	5.2	1.0	6.7	41.5	81.7	28.3	1.8	—

The value reported in each cell is the occurrence (expressed as a percentage over all 1440 models) of strictly lower time consumption for the algorithm on the row compared to that on the column.

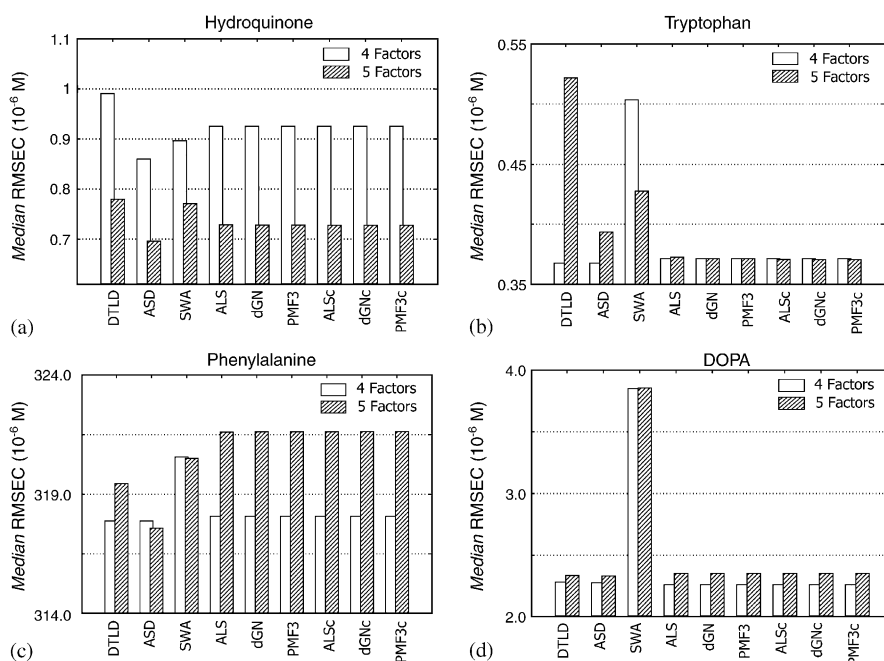


Fig. 5. Median of RMSEC over the three replicates for data set I.

Figs. 5 and 6 show the results from the analysis of data set I. All least-squares methods attain an almost identical solution for each of the replicates, both in terms of loss function and RMSEC. It is also apparent that the ALS algorithm is not as sensitive to over-factoring as the results on the simulated data may have suggested. The non-least squares algorithms (i.e. DTLD, SWATLD and ASD) retrieve factors that, although very close to the underlying ones (still full recovery occurs for all the methods), are visually identifiable as different (Fig.

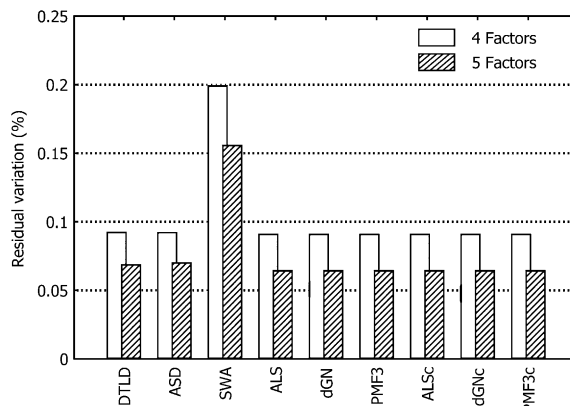


Fig. 6. Residual variation (as % of $\|\mathbf{X}^{(1 \times JK)}\|_F^2$) for data set I.

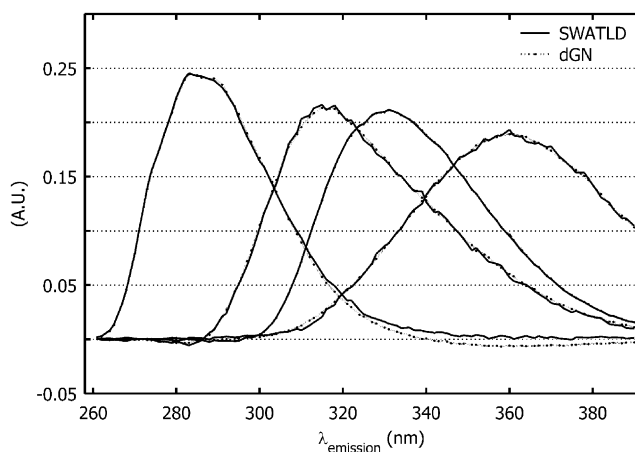


Fig. 7. Emission loadings for data set I (replicate 3) obtained with dGN and SWATLD. It can be noticed that the latter produces noisier profiles and that one of the loadings vectors present systematic differences at high emission wavelengths.

7) and result in different predictions and errors. In particular, SWATLD gives the worst performance for this data set, both in terms of the value of the loss function and of predicted concentrations. As for the simulated data, SWATLD yielded solutions with higher core consistency (mean over the replicates 99.48%) than the least squares ones (99.29%). The core consistency diagnostic may be regarded as an indication of the relevance of the ‘model error’, which includes the effect of small deviations from perfect trilinearity (Bro and Kiers, 2003). Such deviations are not present in simulated data sets that are not specifically designed to test this aspect, but are inherent to real data and are accounted for in the correct least

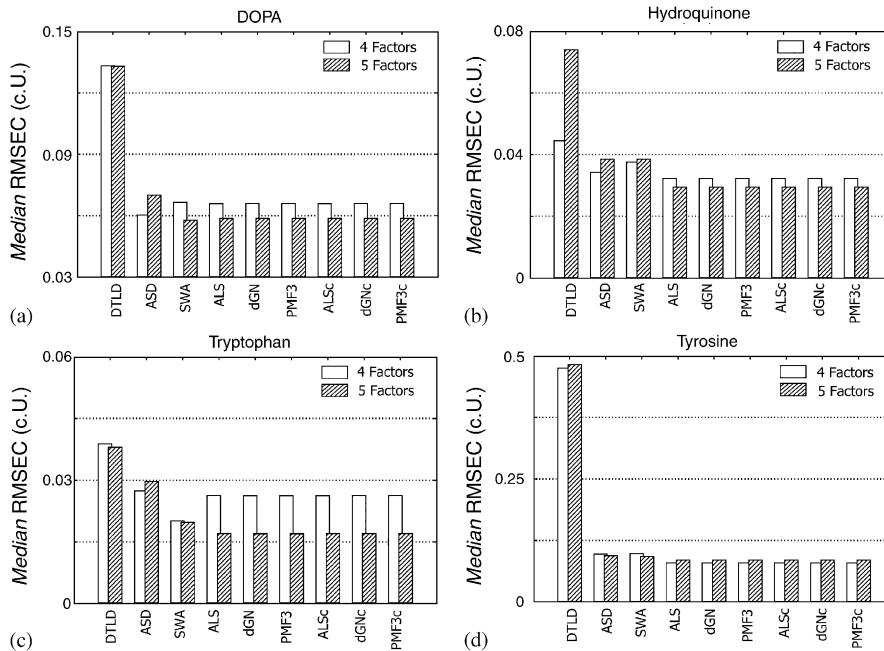


Fig. 8. Median of RMSEC over the six replicates for data set II.

squares solution (Bro and Kiers, 2003). The SWATLD algorithm seems to somewhat ignore such deviations, which results in different solutions and predictions. Similar arguments can be used for ASD, which attains the highest core consistency when the rank is over-estimated, although the differences in the loadings are considerably smaller. The fact that the smallest sum of squared residuals are obtained with the LS methods (both for correctly and over-estimated rank) makes evident once again that the actual optimisation criterion for DTLT, ASD and SWATLD in terms of PARAFAC loss function is not known or easily predictable.

3.5.3. Real data set II

The results for data set II (Figs. 8 and 9) do not differ much from those for data set I. All methods apart from DTLT managed to retrieve the correct solution and the residual variation for this algorithm is larger than for any other. The concentrations predicted using SWATLD are not as bad as for data set I, but this method yields once again a solution with larger residuals and better core consistency (mean over the replicates 99.1% compared to 98.9% for the least squares solutions).

If one excludes DTLT, the performance of ASD in predicting the four concentrations is the worst for three of the analytes. Note that the residual variation for these two methods increases with the number of extracted components. Over-factoring does not seem to hinder the retrieval of the correct components and the RMSEC remains largely unchanged. Yet, for the LS algorithms, it was necessary sometimes to restart the calculations from a new random estimation for the loading matrices (that is, more than 10 random starts were necessary)

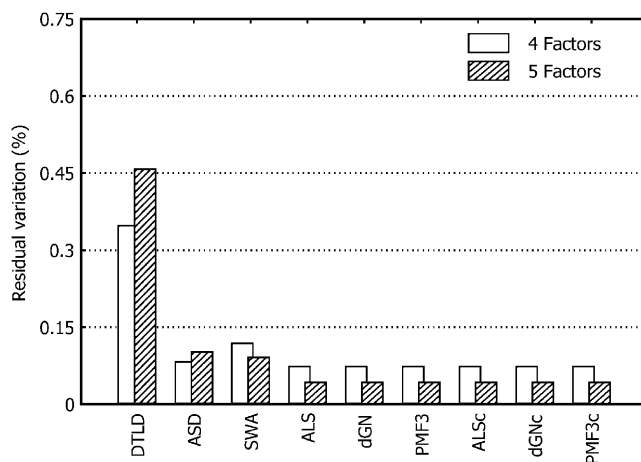


Fig. 9. Residual variation (as % of $\|\mathbf{X}^{(I \times J \times K)}\|_F^2$) for data set II.

as they tended to get stuck in degenerate solutions. This illustrates how these methods are sensitive to the initialisation values, apparently more than the non-least squares techniques. In any case, the correct solution has always been attained within a three attempts with further starts.

The high collinearity of some of the underlying factors did not impede the recovery of the correct components. As was the case for the simulated data sets, when the noise is small, the right components can be extracted also when their congruence is very high.

4. Conclusions

The tests on the nine algorithms showed quite clearly that the advantages of some of the recently proposed alternatives to PARAFAC-ALS were somewhat overestimated.

In particular, ASD appeared consistently inferior to the other iterative methods on both the simulated and the real data sets. Conversely, SWATLD showed a good capability of finding the real underlying components, albeit biased by the fact that it yields solutions with higher core consistency than the least squares one. The distortion induced in the loadings by the non-least squares criterion gives rise to conflicting performances of this algorithm on the real and the simulated data sets. However, the fact that SWATLD neither finds least squares solution nor explicitly optimises with respect to a well-characterised criterion (e.g. the core-consistency) should discourage the use of this algorithm alone for fitting a PARAFAC model and suggests that additional care should be used when evaluating the results it provides. On the other hand, its computational efficiency and resistance to a wrong estimation of the model's rank make this method an excellent candidate for the initialisation of least squares algorithms. The derivative-based methods showed better convergence properties than ALS, especially in case of highly collinear factors, but are too expensive (for time and memory

consumption) to be a feasible option in case of large data sets, such as those commonly encountered in spectroscopy and in most of the fields where the PARAFAC model is a possible choice. Consequently, their viability in these cases seems to require a preliminary compression step, which appears to have the advantage of improving the resistance to over-factoring. With respect to the latter point, dGNC with compression appears as more efficient than PMF3c (and PMF3) and ALS confirmed to be more sensitive to over-factoring than all the other methods apart from DTLD.

In conclusion, for difficult problems and when the quality of the solution is very important or the number of factors unknown, dGNC appears as the best choice, albeit more expensive than ALS. On the other hand, when the rank of the array is not known beforehand and one is interested in exploring the underlying trilinear phenomena linked to a given set of data, SWATLD can give some insight on the factors. For all other cases, the standard ALS represents a good trade-off between computational expense and quality of the solution.

An interesting possibility is to combine the different algorithms to exploit their best features and compensate their shortcomings. E.g. SWATLD could be used to provide the initial values for a PMF3/dGN algorithm with compression and a final refinement could be performed using the standard ALS algorithm.

Acknowledgements

The authors are grateful for support from STVF through the ODIN consortium. They would also like to thank N.M. Faber for providing some of the m-files.

Appendix A. Factor norm distribution over the loading vectors

Due the scaling indeterminacy each factor of a PARAFAC model with F components can be expressed as

$$\hat{\mathbf{X}} = \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T = \sum_{f=1}^F r_f \mathbf{a}_f (t_f \mathbf{c}_f \odot s_f \mathbf{b}_f)^T \equiv \sum_{f=1}^F \tilde{\mathbf{a}}_f (\tilde{\mathbf{c}}_f \odot \tilde{\mathbf{b}}_f)^T, \quad (\text{A.1})$$

where \mathbf{a}_f , \mathbf{b}_f and \mathbf{c}_f are the f -th columns of respectively \mathbf{A} , \mathbf{B} and \mathbf{C} , r_f , s_f and t_f are arbitrary non-zero scalars such that $r_f s_f t_f = 1$ and $\tilde{\mathbf{a}}_f = r_f \mathbf{a}_f$, $\tilde{\mathbf{b}}_f = s_f \mathbf{b}_f$ and $\tilde{\mathbf{c}}_f = t_f \mathbf{c}_f$. To avoid this ambiguity, r_f , s_f and t_f are normally defined as

$$r_f = \|\mathbf{b}\|_2 \cdot \|\mathbf{c}\|_2, \quad (\text{A.2a})$$

$$s_f = \|\mathbf{b}\|_2^{-1} \quad (\text{A.2b})$$

and

$$t_f = \|\mathbf{c}\|_2^{-1}, \quad (\text{A.2c})$$

or, in other words, such that $\|\tilde{\mathbf{b}}_f\|_2 = \|\tilde{\mathbf{c}}_f\|_2 = 1$. In order to equally weight the three loading matrices, a different choice was made for the implementation of dGN, PMF3 and SWATLD. By defining $q_f = (\|\mathbf{a}_f\|_2 \cdot \|\mathbf{b}_f\|_2 \cdot \|\mathbf{c}_f\|_2)^{1/3}$ and calculating r_f , s_f and t_f as $q_f \|\mathbf{a}_f\|_2^{-1}$, $q_f \|\mathbf{b}_f\|_2^{-1}$ and $q_f \|\mathbf{c}_f\|_2^{-1}$ respectively, it holds that

$$\begin{aligned} r_f s_f t_f &= q^3 \|\mathbf{a}_f\|_2^{-1} \cdot \|\mathbf{b}_f\|_2^{-1} \cdot \|\mathbf{c}_f\|_2^{-1} \\ &= \|\mathbf{a}_f\|_2 \cdot \|\mathbf{b}_f\|_2 \cdot \|\mathbf{c}_f\|_2 \|\mathbf{a}_f\|_2^{-1} \cdot \|\mathbf{b}_f\|_2^{-1} \cdot \|\mathbf{c}_f\|_2^{-1} = 1 \end{aligned} \quad (\text{A.3})$$

and that the norms of the three loading vectors $\tilde{\mathbf{a}}_f$, $\tilde{\mathbf{b}}_f$ and $\tilde{\mathbf{c}}_f$ are identical and equal to q_f .

It is worth noticing that no mention is made about the sign of the three scalars in Eqs. (A.2a–c). This problem is normally referred to as ‘sign indeterminacy’ and no specific convention is generally preferred. For the implemented algorithms, the choice has been to set the sign of each loading vector of \mathbf{B} and \mathbf{C} so that the sign of the sum of its elements was positive.

Appendix B. Improved updates for SWATLD

The loss function $L_C(\mathbf{C})$ for the SWATLD algorithm with respect to \mathbf{C} can be split in:

$$\begin{aligned} L_1(\mathbf{C}) &= \sum_{k=1}^K \left\| \left(\mathbf{A}^+ \mathbf{X}_{..k} - \mathbf{D}_k \mathbf{B}^T \right)^T \mathbf{D}_B^{-1} \right\|_F^2 \\ &= \sum_{j=1}^J \left\| \mathbf{D}_B^{-1} \left(\mathbf{A}^+ \mathbf{X}_{.j.} - \mathbf{D}_j \mathbf{C}^T \right) \right\|_F^2 \end{aligned} \quad (\text{B.1})$$

and

$$\begin{aligned} L_2(\mathbf{C}) &= \sum_{k=1}^K \left\| \left(\mathbf{X}_{..k} \mathbf{B}^{+T} - \mathbf{A} \mathbf{D}_k \right) \mathbf{D}_A^{-1} \right\|_F^2 \\ &= \sum_{i=1}^I \left\| \left(\mathbf{X}_{j..}^T \mathbf{B}^{+T} - \mathbf{C} \mathbf{D}_i \right) \mathbf{D}_A^{-1} \right\|_F^2, \end{aligned} \quad (\text{B.2})$$

where $\mathbf{X}_{i..}$, $\mathbf{X}_{.j.}$, \mathbf{D}_i and \mathbf{D}_j are defined similarly to $\mathbf{X}_{..k}$ and \mathbf{D}_k . Using the relation $\|\mathbf{R}\|_F^2 = \text{tr}(\mathbf{R}^T \mathbf{R})$, where $\text{tr}(\mathbf{R})$ denotes the trace of \mathbf{R} , the derivative of L_1 with respect to \mathbf{C}

is (Magnus and Neudecker, 1999):

$$\begin{aligned}
 \frac{\partial L_1(\mathbf{C})}{\partial \mathbf{C}} &= \frac{\partial}{\partial \mathbf{C}} \left[\sum_{j=1}^J \text{tr} \left(\mathbf{X}_j^T \mathbf{A}^{+T} \mathbf{D}_B^{-2} \mathbf{A}^+ \mathbf{X}_j \right) + \sum_{j=1}^J \text{tr} \left(\mathbf{C} \mathbf{D}_B^{-2} \mathbf{D}_j^2 \mathbf{C}^T \right) \right. \\
 &\quad \left. - 2 \sum_{j=1}^J \text{tr} \left(\mathbf{D}_B^{-2} \mathbf{D}_j \mathbf{A}^+ \mathbf{X}_j \mathbf{C} \right) \right] \\
 &= \sum_{j=1}^J \left[0 + 2 \mathbf{C} \mathbf{D}_B^{-2} \mathbf{D}_j^2 - 2 \left(\mathbf{D}_B^{-2} \mathbf{D}_j \mathbf{A}^+ \mathbf{X}_j \right)^T \right] \\
 &= 2\mathbf{C} - 2 \left(\sum_{j=1}^J \left(\mathbf{D}_j \mathbf{A}^+ \mathbf{X}_j \right)^T \right) \mathbf{D}_B^{-2}, \tag{B.3}
 \end{aligned}$$

where the last equality stems from $\sum_{j=1}^J \mathbf{D}_j^2 = \mathbf{D}_B^2$. Since $\sum_{j=1}^J \left(\mathbf{D}_j \mathbf{A}^+ \mathbf{X}_j \right)^T = \mathbf{X}^{(K \times IJ)} (\mathbf{B} \odot \mathbf{A}^{+T})$, Eq. (B.3) becomes

$$\frac{\partial L_1(\mathbf{C})}{\partial \mathbf{C}} = 2\mathbf{C} - 2\mathbf{X}^{(K \times IJ)} (\mathbf{B} \odot \mathbf{A}^{+T}) \mathbf{D}_B^{-2}. \tag{B.4}$$

Eq. (9) follows by applying the analogous derivation to $L_2(\mathbf{C})$ and by solving for \mathbf{C} the equation

$$\frac{\partial L_C(\mathbf{C})}{\partial \mathbf{C}} = \frac{\partial L_1(\mathbf{C})}{\partial \mathbf{C}} + \frac{\partial L_2(\mathbf{C})}{\partial \mathbf{C}} = 0, \tag{B.5}$$

which holds because one is looking for a stationary point.

Appendix C. Derivation of the system of normal equations for PMF3

The loss function for PMF3 can be expressed as

$$\tilde{L}_{\text{PMF3}}(\mathbf{p}) = \mathbf{r}^T \mathbf{r} + \gamma (\mathbf{p} - \mathbf{p}^\bullet)^T (\mathbf{p} - \mathbf{p}^\bullet), \tag{C.1}$$

where \mathbf{p}^\bullet is the target value ($\mathbf{0}$ in case of the regularisation: the penalty can be considered as some sort of equality soft constraint operating towards $\mathbf{p} = \mathbf{p}^\bullet = \mathbf{0}$).

A Taylor expansion truncated after the linear term yields:

$$\begin{aligned}
 \tilde{L}_{\text{PMF3}}(\Delta \mathbf{p}) &= \left(\mathbf{r}^0 + \mathbf{J} \Delta \mathbf{p} \right)^T \left(\mathbf{r}^0 + \mathbf{J} \Delta \mathbf{p} \right) \\
 &\quad + \gamma \left(\mathbf{p}^0 + \Delta \mathbf{p} - \mathbf{p}^\bullet \right)^T \left(\mathbf{p}^0 + \Delta \mathbf{p} - \mathbf{p}^\bullet \right), \tag{C.2}
 \end{aligned}$$

where $\mathbf{r}^0 = \mathbf{r}(\mathbf{p}^0)$. Expanding:

$$\begin{aligned}\tilde{L}_{\text{PMF3}}(\Delta\mathbf{p}) &= \mathbf{r}^{0T} \mathbf{r}^0 + 2\Delta\mathbf{p}^T \mathbf{J}^T \mathbf{r}^0 + \Delta\mathbf{p}^T \mathbf{J}^T \mathbf{J} \Delta\mathbf{p} + \gamma \left(\widehat{\mathbf{p}}^{0T} \widehat{\mathbf{p}}^0 + 2\Delta\mathbf{p}^T \widehat{\mathbf{p}}^0 + \Delta\mathbf{p}^T \Delta\mathbf{p} \right) \\ &= \mathbf{r}^{0T} \mathbf{r}^0 + \gamma \widehat{\mathbf{p}}^{0T} \widehat{\mathbf{p}}^0 + 2\Delta\mathbf{p}^T \left(\mathbf{J}^T \mathbf{r}^0 + \gamma \widehat{\mathbf{p}}^0 \right) + \Delta\mathbf{p}^T \left(\mathbf{J}^T \mathbf{J} + \gamma \mathbf{I}_N \right) \Delta\mathbf{p},\end{aligned}\quad (\text{C.3})$$

where $\widehat{\mathbf{p}}^0 = \mathbf{p}^0 - \mathbf{p}^\bullet$. The first derivative is then:

$$\frac{\partial \tilde{L}_{\text{PMF3}}(\Delta\mathbf{p})}{\partial \Delta\mathbf{p}} = 0 + 0 + 2 \left(\mathbf{J}^T \mathbf{r}^0 + \gamma \widehat{\mathbf{p}}^0 \right) + 2 \left(\mathbf{J}^T \mathbf{J} + \gamma \mathbf{I}_N \right) \Delta\mathbf{p} = 0, \quad (\text{C.4})$$

where the last holds because one is looking for a stationary point. Hence

$$\left(\mathbf{J}^T \mathbf{J} + \gamma \mathbf{I}_N \right) \Delta\mathbf{p} = - \left(\mathbf{J}^T \mathbf{r}^0 + \gamma \widehat{\mathbf{p}}^0 \right). \quad (\text{C.5})$$

Finally, including the Levenberg–Marquadt modification one yields Eq. (22).

The regularisation term operates to reduce the step length (provided that the γ has a suitable value), but also has the effect of preventing exceedingly high parameters (i.e. one of the possible effects of scaling indeterminacy and 2FDs), which are penalised.

Appendix D. Helpful definitions and equations

1. Given two matrices \mathbf{A} and \mathbf{B} with the same number f of columns the Khatri–Rao product \odot is defined as

$$\begin{aligned}\mathbf{A} \odot \mathbf{B} &= [\mathbf{a}_1 \otimes \mathbf{b}_1 \quad \mathbf{a}_2 \otimes \mathbf{b}_2 \quad \cdots \quad \mathbf{a}_F \otimes \mathbf{b}_F] \\ &= [\text{vec } \mathbf{b}_1 \mathbf{a}_1^T \quad \cdots \quad \text{vec } \mathbf{b}_F \mathbf{a}_F^T],\end{aligned}\quad (\text{D.1})$$

where \otimes is the Kronecker product (Magnus and Neudecker, 1999).

2. The congruence (also referred to as ‘uncorrected correlation’ or ‘Tucker congruence coefficient’) between the f -th factor $\mathbf{z}_f = \mathbf{a}_f \otimes \mathbf{b}_f \otimes \mathbf{c}_f$ and the g -th factor $\mathbf{z}_g = \mathbf{a}_g \otimes \mathbf{b}_g \otimes \mathbf{c}_g$, can be written as

$$\begin{aligned}\text{cong}(\mathbf{z}_f, \mathbf{z}_g) = \cos(\mathbf{z}_f, \mathbf{z}_g) &= \frac{\mathbf{a}_f^T \mathbf{a}_g}{\|\mathbf{a}_f\|_2 \|\mathbf{a}_g\|_2} \cdot \frac{\mathbf{b}_f^T \mathbf{b}_g}{\|\mathbf{b}_f\|_2 \|\mathbf{b}_g\|_2} \\ &\quad \cdot \frac{\mathbf{c}_f^T \mathbf{c}_g}{\|\mathbf{c}_f\|_2 \|\mathbf{c}_g\|_2}.\end{aligned}\quad (\text{D.2})$$

3. The relation between the scalar L_n multiplying an array \mathbf{R} and the level of noise Noise% (expressed as percentage of the total sum of squares) associated to it is the following:

$$\text{Noise}\% = \frac{L_n^2 \|\mathbf{R}\|_F^2 \cdot \|\mathbf{X}\|_F^2}{\|\mathbf{X}\|_F^2 + L_n^2 \|\mathbf{R}\|_F^2 \cdot \|\mathbf{X}\|_F^2} 100 = \frac{L_n^2}{1 + L_n^2} 100, \quad (\text{D.3})$$

where Noise% is the level, \mathbf{R} is the matricised array containing the noise normalised to a Frobenius norm of 1 and \mathbf{X} is the noiseless data.

L_n assumes values of 0.1005, 0.2294 and 0.3333 to attain respectively 1%, 5% and 10% noise.

4. Given the score matrix \mathbf{A} (i.e. the first loading matrix) and assumed that the f -th component corresponds to the g -th constituent in the analysed data set (i.e. g -th column \mathbf{y}_g of the concentration matrix \mathbf{Y}), the root mean squared error in calibration (RMSEC) for the g -th constituent is defined as

$$\text{RMSEC} = I^{-0.5} \cdot \|\mathbf{y}_g - \hat{\mathbf{y}}_g\|_2, \quad (\text{D.4})$$

where $\hat{\mathbf{y}}_g = [\mathbf{a}_f \quad \mathbf{1}] [\mathbf{a}_f \quad \mathbf{1}]^+ \mathbf{y}_g$ holds the predicted concentrations for the g -th constituent.

References

- Andersson, C.A., Bro, R., 2000. The N-way Toolbox for MATLAB. *Chemometrics Intell. Lab. Systems* 52 (1), 1–4.
- Baunsgaard, D., Munck, L., Norgaard, L., 2000. Analysis of the effect of crystal size and color distribution on fluorescence measurements of solid sugar using chemometrics. *Appl. Spectrosc.* 54 (11), 1684–1689.
- Björck, Å., 1996. *Numerical Methods for Least Squares Problems*. First ed. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.
- Bro, R., 1998. Multi-way analysis in the food industry. *Models Algorithms and Applications*, University of Amsterdam.
- Bro, R., Andersson, C.A., 1998. Improving the speed of multiway algorithms part II. Compression. *Chemometrics Intell. Lab. Systems* 42 (1–2), 105–113.
- Bro, R., Kiers, H.A.L., 2003. A new efficient method for determining the number of components in PARAFAC models. *J. Chemometrics* 17 (5), 274–286.
- Carroll, J.D., Chang, J.J., 1970. Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart–Young decomposition. *Psychometrika* 35 (3), 283–319.
- Carroll, J.D., Pruzansky, S., Kruskal, J.B., 1980. Candeline—a general approach to multidimensional-analysis of many-way arrays with linear constraints on parameters. *Psychometrika* 45 (1), 3–24.
- Chen, Z.P., Wu, H.L., Jiang, J.H., Li, Y., Yu, R.Q., 2000. A novel trilinear decomposition algorithm for second-order linear calibration. *Chemometrics Intell. Lab. Systems* 52 (1), 75–86.
- Chen, Z.P., Li, Y., Yu, R.Q., 2001. Pseudo alternating least squares algorithm for trilinear decomposition. *J. Chemometrics* 15 (3), 149–167.
- Faber, N.M., Bro, R., Hopke, P.K., 2003. Recent developments in CANDECOMP/PARAFAC algorithms: a critical review. *Chemometrics Intell. Lab. Systems* 65 (1), 119–137.
- Golub, G.H., Van Loan, C.F., 1996. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, London. pp. 1–47.
- Harshman, R.A., 1970. Foundations of the PARAFAC procedure: models and conditions for an 'explanatory' multi-modal factor analysis. *UCLA working papers in phonetics*, vol. 16, pp. 1–84.
- Hayashi, C., Hayashi, F., 1982. A new algorithm to solve PARAFAC-model. *Behaviormetrika* 11, 49.
- Ho, C.N., Christian, G.D., Davidson, E.R., 1978. Application of method of rank annihilation to quantitative analyses of multicomponent fluorescence data from video fluorometer. *Anal. Chem.* 50 (8), 1108–1113.
- Ho, C.N., Christian, G.D., Davidson, E.R., 1980. Application of the method of rank annihilation to fluorescent multicomponent mixtures of polynuclear aromatic-hydrocarbons. *Anal. Chem.* 52 (7), 1071–1079.
- Ho, C.N., Christian, G.D., Davidson, E.R., 1981. Simultaneous multicomponent rank annihilation and applications to multicomponent fluorescent data acquired by the video fluorometer. *Anal. Chem.* 53 (1), 92–98.
- Hopke, P.K., Paatero, P., Jia, H., Ross, R.T., Harshman, R.A., 1998. Three-way (PARAFAC) factor analysis: examination and comparison of alternative computational methods as applied to ill-conditioned data. *Chemometrics Intell. Lab. Systems* 43 (1–2), 25–42.

- Jiang, J.H., Wu, H.L., Li, Y., Yu, R.Q., 1999. Alternating coupled vectors resolution (ACOVER) method for trilinear analysis of three-way data. *J. Chemometrics* 13 (6), 557–578.
- Jiang, J.H., Wu, H.L., Li, Y., Yu, R.Q., 2000. Three-way data resolution by alternating slice-wise diagonalization (ASD) method. *J. Chemometrics* 14 (1), 15–36.
- Kiers, H.A.L., 1998. A three-step algorithm for CANDECOMP/PARAFAC analysis of large data sets with multicollinearity. *J. Chemometrics* 12 (3), 155–171.
- Kiers, H.A.L., Ten Berg, J.M.F., Bro, R., 1999. PARAFAC2—Part I. A direct fitting algorithm for the PARAFAC2 model. *J. Chemometrics* 13 (3–4), 275–294.
- Krijnen, W.P., 1993. *The Analysis of Three-Way Arrays by Constrained PARAFAC Methods*. University of Groningen, The Netherlands.
- Kruskal, J.B., Harshman, R.A., Lundy, M.E., 1989. How (MFA) data can cause degenerate PARAFAC solutions, among other relationships. In: Coppi, R., Bolasco, S. (Eds.), *Multiway Data Analysis*. Elsevier Science Publishers, Amsterdam, pp. 115–122.
- Levenberg, K., 1944. A method for the solution of certain problems in least squares. *Quart. Appl. Math.* 2, 164–168.
- Li, Y., Jiang, J.H., Wu, H.L., Chen, Z.P., Yu, R.Q., 2000. Alternating coupled matrices resolution method for three-way arrays analysis. *Chemometrics Intell. Lab. Systems* 52 (1), 33–43.
- Liu, X.Q., Sidiropoulos, N.D., 2001. Cramer–Rao lower bounds for low-rank decomposition of multidimensional arrays. *IEEE Trans. Signal Process.* 49 (9), 2074–2086.
- Lorber, A., 1985. Features of quantifying chemical-composition from two-dimensional data array by the rank annihilation factor-analysis method. *Anal. Chem.* 57 (12), 2395–2397.
- Madsen, K., Nielsen, H.B., Tingleff, O., 2004. *Methods for non-linear least squares problems*. second ed. Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark.
- Magnus, J.R., Neudecker, H., 1999. *Matrix Differential Calculus With Applications in Statistics and Econometrics*. revised ed. Wiley, New York.
- Marquadt, D., 1963. An Algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.* 11, 431–441.
- MATLAB 6.5, The Mathworks, www.themathworks.com.
- Mitchell, B.C., Burdick, D.S., 1994. Slowly converging parafac sequences—swamps and 2-Factor degeneracies. *J. Chemometrics* 8 (2), 155–168.
- Paatero, P., 1997. A weighted non-negative least squares algorithm for three-way ‘PARAFAC’ factor analysis. *Chemometrics Intell. Lab. Systems* 38 (2), 223–242.
- Paatero, P., 2000. Construction and analysis of degenerate PARAFAC models. *J. Chemometrics* 14 (3), 285–299.
- Rao, C.R., Mitra, S., 1971. *Generalized Inverse of Matrices and Its Applications*. Wiley, New York.
- Rayens, W.S., Mitchell, B.C., 1997. Two-factor degeneracies and a stabilization of PARAFAC. *Chemometrics Intell. Lab. Systems* 38 (2), 173–181.
- Riu, J., Bro, R., 2003. Jack-knife technique for outlier detection and estimation of standard errors in PARAFAC models. *Chemometrics Intell. Lab. Systems* 65 (1), 35–49.
- Sanchez, E., Kowalski, B.R., 1986. Generalized rank annihilation factor-analysis. *Anal. Chem.* 58 (2), 496–499.
- Sanchez, E., Kowalski, B.R., 1990. Tensorial resolution: a direct trilinear decomposition. *J. Chemometrics* 4, 29–45.
- Sidiropoulos, N.D., Bro, R., 2000. On the uniqueness of multilinear decomposition of N-way arrays. *J. Chemometrics* 14 (3), 229–239.
- Tomasi, G., Bro, R., 2004. PARAFAC and missing values. *Chemometrics Intell. Lab. Systems*, in press.
- Wu, H.L., Shibukawa, M., Oguma, K., 1998. An alternating trilinear decomposition algorithm with application to calibration of HPLC-DAD for simultaneous determination of overlapped chlorinated aromatic hydrocarbons. *J. Chemometrics* 12 (1), 1–26.
- Yates, F., 1933. The analysis of replicated experiments when the field results are incomplete. *Empire J. Exp. Agri.* 1, 129.

Paper II

PARAFAC and missing values

G. Tomasi, R. Bro

Chemometrics and Intelligent Laboratory Systems, 75, 2005,
163 -180.

PARAFAC and missing values

Giorgio Tomasi*, Rasmus Bro

Food Technology, Royal Veterinary and Agricultural University, Rolighedsvej 30, 1958 Frederiksberg C, Denmark

Received 1 December 2003; received in revised form 8 July 2004; accepted 9 July 2004

Available online 11 September 2004

Abstract

Missing values are a common occurrence in chemometrics data, and different approaches have been proposed to deal with them. In this work, two different concepts based on two algorithms are compared in their efficiency in dealing with incomplete data when fitting the PARAFAC model: single imputation (SI) combined with a standard PARAFAC-alternating least squares (ALS) algorithm, and fitting the model only to the existing elements using a computationally more expensive method (Levenberg–Marquadt) appropriately modified and optimised.

The performance of these two algorithms and the effect of the incompleteness of the data on the final model have been evaluated on the basis of a Monte Carlo study and real data sets with different amounts and patterns of missing values (randomly missing values, randomly missing spectra/vectors, and systematically missing spectra/vectors).

The evaluation is based on the quality of the solution as well as on computational aspects (time requirement and number of iterations). The results show that a PARAFAC model can be correctly determined even when a large fraction of the data is missing (up to 70%), and that the pattern matters more than the fraction of missing values. Computationally, the Levenberg–Marquadt-based approach appeared superior for the pattern of missing values typical of fluorescence measurements when the fraction of missing elements exceeded 30%.

© 2004 Elsevier B.V. All rights reserved.

Keywords: PARAFAC; Missing values; INDAFAC; Fluorescence

1. Introduction

In chemometrics, incomplete observations and missing values can be found in a large number of applications ranging from calibration problems to statistical process control. Recent studies have pursued the algorithmic problem in connection with missing values for two-way models [1–4], with specific focus on PCA and PLS and, to a certain extent, three-way models [1,5,6]. The aim of this paper is to study the effect of non-observed values on fitting a PARAFAC model and to compare the performances of two algorithms fitting such model in presence of missing values: PARAFAC-alternating least squares (ALS) with single imputation (ALS-SI) and the least squares approach called INcomplete DATA paraFAC

(INDAFAC) based on a suitably modified Levenberg–Marquadt algorithm.

This study is based on a Monte Carlo simulation where 2400 data sets were generated varying a specific set of conditions (rank of the array, percentage of missing elements and their pattern in the array, collinearity between factors, and level of noise) and on three real data sets comprising fluorescence measurements and having known rank.

1.1. PARAFAC model

If one considers a three-way array $\underline{\mathbf{X}}$ of dimensions $I \times J \times K$, the PARAFAC model can be expressed as

$$x_{ijk} = \sum_{f=1}^F a_{if} b_{jf} c_{kf} + r_{ijk} \quad i = 1 \dots I, j = 1 \dots J, k = 1 \dots K \quad (1)$$

* Corresponding author.

E-mail address: gt@kvl.dk (G. Tomasi).

where x_{ijk} is the measured value, a_{if} , b_{jf} , and c_{kf} represent the parameters to estimate, r_{ijk} are the residuals, and F is the number of sought factors.

By defining the three loading matrices:

$$\mathbf{A} = \{a_{if} | i = 1 \dots I, f = 1 \dots F\}$$

$$\mathbf{B} = \{b_{jf} | j = 1 \dots J, f = 1 \dots F\}$$

$$\mathbf{C} = \{c_{kf} | k = 1 \dots K, f = 1 \dots F\} \quad (2)$$

and employing the column-wise Khatri–Rao product (\odot) [7], Eq. (1) can be written as

$$\mathbf{X}^{(I \times JK)} = \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T + \mathbf{R}^{(I \times JK)}, \quad (3)$$

where $\mathbf{X}^{(I \times JK)}$ is the matricised form of the data array [the superscript $(I \times JK)$ identifying the way the array is matricised; [7], and the superscript T indicates the transpose operation.

Fitting the PARAFAC model to \mathbf{X} in the least squares sense can be expressed as the minimisation problem:

$$\arg \min_{\mathbf{A}, \mathbf{B}, \mathbf{C}} \|\mathbf{X}^{(I \times JK)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T\|_F^2 \quad (4)$$

where $\|\mathbf{Y}\|_F$ is the Frobenius norm (i.e., the squared root of the sum of the squared elements of the matrix \mathbf{Y}).

Solving problem (4) corresponds to fitting the PARAFAC model in the maximum likelihood sense provided that the residuals $\mathbf{r} = \text{vec} \mathbf{R}^{(I \times JK)}$ (where the vec operator is defined as in Ref. [8]) are normally distributed with mean 0 and variance $\sigma^2 \mathbf{I}$ [9], viz. that the noise is uncorrelated and homoscedastic. Albeit for real life problems this is hardly ever the case, it has been shown in several applications that such fitting is adequate also when slight deviations occur [7].

Numerous algorithms have been proposed for solving problem (4) [10,11], two of them, namely, PARAFAC-ALS with single imputation and PARAFAC-LM (where LM stands for Levenberg–Marquadt) can be effectively employed in presence of missing values and are described in Section 2.

1.2. Missing values patterns

Missing values may occur in data sets for a number of reasons: glitches and malfunctions of one or more sensors, irregular measurement intervals between samples, or different sampling frequencies for the various sensors. In some cases (e.g., fluorescence Emission/Excitation Matrices—EEM), the missing values are not

necessarily present originally in the data as obtained from the instrument, but are inserted as a postprocessing to yield data more suitable for being described by a multilinear model [6,7]. Depending on the cause for the missing values, their pattern within the array may change considerably, having different effects on the model fitting process.

In the simplest case to treat, but also the one that is most seldom found in practice, the missing elements are randomly scattered over the array without any specific pattern (Fig. 1a). One such situation may occur when several, distinct sensors are used to monitor one process in time and there are momentary malfunctions in the single sensor. Analogously, a survey of several variables both in time and space may not follow a particularly regular pattern, and certain sites (e.g., the least accessible ones) may be visited with lower frequency. Such a pattern is referred to as randomly missing values (RMV).

A second pattern, here denoted as randomly missing spectra (RMS), encompasses the case of entirely missing “tubes” (Fig. 1b), once again completely at random. This situation may occur when a process is monitored in time by means of a multivariate instrument (e.g., a spectrometer). If the measurement is not taken, either due to malfunctioning or caused by an irregular sampling scheme, a whole spectrum (i.e., a tube) will be missing. An analogous situation would present itself if a certain sensor or channel stops working and is not replaced until the process is terminated; only in this case the “tube” would be missing in the time mode of the array rather than in the spectral one.

Finally, the missing values pattern may be completely systematic, as, for example, would happen if the sensors used for the monitoring of a process have a different sampling frequency. Indeed, many cases of systematically missing values (SMV) can be identified. One that is particularly interesting, because it is common for the kind of data to which PARAFAC is often applied, is represented by EEM fluorescence measurements. In fluorescence, the signal registered at emission wavelengths lower than the excitation wavelength is physically zero (Fig. 1c). The presence of these zeros, however, may interfere with the multilinearity of the data [6], provoking artefacts in the final solution. At the same time, Raman and Rayleigh scatter (Fig. 1c), cannot be adequately modelled by PARAFAC components as they are not low-rank trilinear [5,6,12]. Because both these parts of the recordings are not connected to chemical information, the values in this range are normally set to missing, although this is also often suboptimal and associated with other kinds of modelling problems [5,12]. The pattern of the missing values within the array in the latter case is systematic and constant over the samples: entire tubes are missing across the sample mode (Fig. 1d). In the remaining part of this work, this pattern will be referred to as systematically missing spectra (SMS).

2. Algorithms

2.1. Alternating least squares with single imputation (ALS-SI)

2.1.1. PARAFAC-ALS

The most common algorithm used to fit a PARAFAC model is based on the alternating least squares idea [13]: the nonlinear problem (4) is split into smaller, linear subproblems that are solved iteratively until convergence is established. Because all the steps are optimised in the least square sense, the loss function $L(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \|\mathbf{X} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T\|_F^2$ is bound not to increase at each step and tends to a (possibly local) minimum.

Given initial estimates for \mathbf{B} and \mathbf{C} , Eq. (4) becomes linear with respect to the matrix \mathbf{A} , and an interim optimal least squares estimate of the latter can be computed as

$$\mathbf{A}^{(s)} = \mathbf{X}^{(I \times JK)} \left(\left(\mathbf{C}^{(s-1)} \odot \mathbf{B}^{(s-1)} \right)^T \right)^+, \quad (5a)$$

where $s-1=0$ indicates the initial estimates for \mathbf{B} and \mathbf{C} , respectively, and $+$ indicates the Moore–Penrose generalised inverse. The computation of $\mathbf{A}^{(s)}$ on the basis of $\mathbf{B}^{(s-1)}$ and $\mathbf{C}^{(s-1)}$, where s indicates the iteration number, is followed by analogous substeps determining $\mathbf{B}^{(s)}$ and $\mathbf{C}^{(s)}$:

$$\mathbf{B}^{(s)} = \mathbf{X}^{(J \times IK)} \left(\left(\mathbf{C}^{(s-1)} \odot \mathbf{A}^{(s)} \right)^T \right)^+ \quad (5b)$$

$$\mathbf{C}^{(s)} = \mathbf{X}^{(K \times IJ)} \left(\left(\mathbf{B}^{(s)} \odot \mathbf{A}^{(s)} \right)^T \right)^+ \quad (5c)$$

After Eq. (5c) the convergence is checked: if the value of $L(\mathbf{A}, \mathbf{B}, \mathbf{C})$ has decreased in relative terms less than a chosen small positive number (the convergence criterion), the algorithm is stopped; otherwise, it continues estimating $\mathbf{A}^{(s+1)}$ for fixed $\mathbf{B}^{(s)}$ and $\mathbf{C}^{(s)}$ (i.e., the next iteration step).

The Khatri–Rao product has a property that allows significant savings in the calculations. Specifically:

$$(\mathbf{B} \odot \mathbf{A})^T (\mathbf{B} \odot \mathbf{A}) = \mathbf{B}^T \mathbf{B} * \mathbf{A}^T \mathbf{A} \quad (6)$$

where $*$ is the Hadamard (element-wise) product. Following the fact that $\mathbf{M}^+ = (\mathbf{M}^T \mathbf{M})^+ \mathbf{M}^T$ [8], Eq. (5a) is solved as

$$\mathbf{A} = \mathbf{X}^{(I \times JK)} (\mathbf{C} \odot \mathbf{B}) (\mathbf{B}^T \mathbf{B} * \mathbf{C}^T \mathbf{C})^+, \quad (7)$$

where the indices relative to the iterations are skipped for clarity. The ALS algorithm has only linear convergence and slows down even further when it encounters so-called swamps, i.e., regions where two or more factors grow increasingly collinear and arbitrarily large maintaining opposite sign while the loss function decreases very slowly [14,15]. In order to accelerate the convergence, several strategies have been devised [7]. One that proved efficient in many cases uses a so-called line-search procedure [7,13],

which is based on the observation that the ALS algorithm, particularly when stuck in a swamp, often proceeds with increasingly shorter steps in very collinear directions for several consecutive iterations. The line-search acceleration tests, every given number of iterations, if a longer step along the latest computed update for the loading matrices leads to a larger decrease of the loss function [7].

2.1.2. Handling missing data

The ALS algorithm, as described in the previous section, cannot handle missing values and requires some modifications to operate in the presence of incomplete observations. One method, which has been successfully employed with other multilinear models [1,3,4,7], is represented by single imputation.

In such procedure, Eqs. (5a)–(5c) are applied, instead of the original array \mathbf{X} , to an array $\tilde{\mathbf{X}}$ defined as

$$\tilde{\mathbf{X}}^{(s)} = \mathbf{X} * \mathbf{M} + \mathbf{Y}^{(s)} * (\mathbf{1} - \mathbf{M}) \quad (8)$$

where $\mathbf{Y}^{(s)}$ is the interim model computed at the s -th iteration, and $\mathbf{1}$ is an array of ones having the same dimensions of \mathbf{X} . \mathbf{M} is an array whose elements are defined as

$$m_{ijk} = \begin{cases} 0 & \text{if } x_{ijk} \text{ is missing} \\ 1 & \text{if } x_{ijk} \text{ is not missing} \end{cases} \quad (9)$$

$\tilde{\mathbf{X}}$ contains no missing values and thus allows the use of the standard PARAFAC-ALS algorithm to estimate the model parameters. $\tilde{\mathbf{X}}^{(s)}$ is updated at every iteration on the base of Eq. (3). The zero-iteration approximation $\mathbf{Y}^{(0)}$ is reckoned depending on the pattern of the missing values. In general, it is taken as the average of the observed values in the corresponding columns/tubes or of the whole array.

The single imputation algorithm, under the conditions of normality (with zero mean and identical variance) and independence of the residuals, falls into the category of the Expectation Maximisation (EM) approach for incomplete data sets. The EM method was devised in the maximum-likelihood framework [16] and is divided in two steps: the E-step and the M-step. In the E-step, the conditional expectation of the likelihood function is computed given the observed data and the current estimated parameters. In least squares terms, this corresponds to calculating the loss function with respect to Eq. (8), i.e.,

$$\begin{aligned} L(\mathbf{A}^{(s)}, \mathbf{B}^{(s)}, \mathbf{C}^{(s)}) &= \|\tilde{\mathbf{X}}^{(s)} - \mathbf{Y}^{(s)}\|_F^2 \\ &= \|\tilde{\mathbf{X}}^{(s)} - \mathbf{A}^{(s)} (\mathbf{C}^{(s)} \odot \mathbf{B}^{(s)})^T\|_F^2 \end{aligned} \quad (10)$$

where the superscript relative to the unfolding has been skipped for clarity. The loss function (Eq. (10)) represents the expected value of the log-likelihood function (with changed sign) given the above assumptions on the residuals.

The M-step determines new estimates for the parameters maximising the likelihood function. This step is simply represented by Eqs. (5a)–(5c) and the computation of the corresponding \mathbf{Y} .

It has been demonstrated that Eq. (10) is bound not to increase and that the convergence of the procedure is linear with a convergence rate that is related to the amount of missing information [17,18].

This suggests that the already relatively slow convergence rate of ALS may be further reduced by an increased amount of missing values. Furthermore, whereas upon final convergence the estimates for the missing values have no influence on the estimated parameters, a large amount of missing elements may increase the risk of convergence to a local minimum as the interim model would describe for the largest part the (erroneous) imputed values. These two aspects pose the premises for the use of the modified Levenberg–Marquadt algorithm described in the following section.

2.2. Incomplete data PARAFAC (INDAFAC)

2.2.1. PARAFAC-LM

The Levenberg–Marquadt method is a modification of the Gauss–Newton (iterative) algorithm for solving non-linear least squares problems [19,20] and has been proposed for solving problem (4) in several instances [11,21,22]. In order to describe this method, it is necessary to introduce a vectorised notation for the PARAFAC model:

$$\mathbf{x} = \text{vec}\mathbf{X}^{(I \times JK)} = \text{vec}[\mathbf{A}(\mathbf{C} \odot \mathbf{B})^T] + \text{vec}\mathbf{R}^{(I \times JK)} \quad (11)$$

If one defines a vector $\mathbf{p} = \text{vec}[\mathbf{A}^T | \mathbf{B}^T | \mathbf{C}^T]$ holding the model parameters, problem (4) can be expressed as

$$\arg \min_{\mathbf{p}} \|\mathbf{r}(\mathbf{p})\|_2^2 = \arg \min_{\mathbf{p}} (\mathbf{x} - \mathbf{y}(\mathbf{p}))^T (\mathbf{x} - \mathbf{y}(\mathbf{p})) \quad (12)$$

where $\mathbf{r} = \text{vec}\mathbf{R}^{(I \times JK)}$ and $\mathbf{y} = \text{vec}[\mathbf{A}(\mathbf{C} \odot \mathbf{B})^T]$.

In the Gauss–Newton algorithm (and the Levenberg–Marquadt modification), an update $\Delta\mathbf{p}$ for all the parameters is computed at each iteration, and the new estimates for the model parameters are defined as $\mathbf{p}^{(s)} = \mathbf{p}^{(s-1)} + \Delta\mathbf{p}$. This method is based on a Taylor expansion of the residuals with respect to the interim parameters $\mathbf{p}^{(s)}$:

$$\mathbf{r}(\mathbf{p}^{(s)} + \Delta\mathbf{p}) = \mathbf{r}(\mathbf{p}^{(s)}) + \mathbf{J}\Delta\mathbf{p} + O(\|\Delta\mathbf{p}\|_2^2) \quad (13)$$

where \mathbf{J} is the Jacobian matrix of $\mathbf{r}(\mathbf{p})$, i.e., an $(I+J+K)F$ matrix whose elements are defined as

$$j_{mn} = \frac{\partial r_m}{\partial p_n} = -\frac{\partial y_m}{\partial p_n} \quad (14)$$

If one ignores the error term $O(\|\Delta\mathbf{p}\|_2^2)$ in Eq. (13), the update $\Delta\mathbf{p}$ for all the parameters can be computed as the solution to the linear least squares problem [19]:

$$\arg \min_{\Delta\mathbf{p}} \|\mathbf{r}(\mathbf{p}^{(s)}) + \mathbf{J}(\mathbf{p}^{(s)})\Delta\mathbf{p}\|_2^2 \quad (15)$$

There are several methods for solving Eq. (15). The one employed here is based on the system of normal equations:

$$\mathbf{J}^T \mathbf{J} \Delta\mathbf{p} = \mathbf{J}^T \mathbf{r} \quad (16)$$

which is solved by means of a Cholesky decomposition and back-substitution. The choice is justified by the sparsity of the Jacobian and by its dimensions [11]. Because of its computational complexity (each update requires approximately $O(N^3)$ operations, where N is the number of parameters), this solution is suited for small- and medium-size problems. Iterative methods, such as Preconditioned Conjugate Gradients, may be more efficient for large-scale problems [22].

The Gauss–Newton algorithm described thus far is particularly appealing, because it guarantees quadratic convergence provided that the initial estimates for the parameters are close enough to the solution and that the residuals at the solution are not too large [19,20]. On the other hand, if these conditions are not fulfilled, the algorithm may not converge at all. Furthermore, the method requires modifications if the Jacobian is rank-deficient [19], as it is the case when fitting a PARAFAC model: due to the scaling indeterminacy intrinsic to this model, $2F$ (for a three-way array) of the Jacobian singular values are zeros to machine precision [21,22].

The Levenberg–Marquadt modification (LM) of the Gauss–Newton algorithm copes with all these problems, thus yielding a globally convergent algorithm [19,20]. In the LM algorithm, the system of normal Eq. (16) is modified to

$$(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I}_{(I+J+K)F}) \Delta\mathbf{p} = \mathbf{J}^T \mathbf{r} \quad (17)$$

The algorithm belongs to the category of the trust region methods. In essence, a “trust region”, which radius is a function of λ , is a sphere centred in the current estimate $\mathbf{p}^{(s)}$ where the linear approximation for the residuals is assumed to hold. The update $\Delta\mathbf{p}$ is computed so that it minimises the residuals inside this region. If $\Delta\mathbf{p}$ leads to an insufficient decrease of the loss function, the update is rejected, the trust region is shrunk (i.e., λ is increased), and a new update is calculated. Various strategies exist to define whether the update should be accepted or rejected and how to update λ . The one used here is described in detail in Ref. [20] and is based on the ratio between the linearly predicted decrease of the loss function $L(\mathbf{p}) - \|\mathbf{r}(\mathbf{p}) + \mathbf{J}(\mathbf{p})\Delta\mathbf{p}\|_2^2$ and the actual decrease after the update $L(\mathbf{p}) - L(\mathbf{p} + \Delta\mathbf{p})$.

The scaling indeterminacy poses another problem related to the numerical stability of the algorithm. If the standard scaling convention for the loading matrices is used (i.e., $\|\mathbf{b}_f\|_2 = \|\mathbf{c}_f\|_2 = 1$), the “practical” condition number of the Jacobian (i.e., computed disregarding the scaling indeterminacy—see Section 3.2) may become exceedingly large (typically because $a_{if}b_{jf} = a_{if}c_{kf} \gg b_{jf}c_{kf}$). This can be avoided by setting the norm of the three loading vectors of the same component to be the same and equal to $q_f = (\|\mathbf{a}_f\|_2 \|\mathbf{b}_f\|_2 \|\mathbf{c}_f\|_2)^{1/3}$ [11].

2.2.2. Handling missing values (INDAFAC)

This PARAFAC-LM algorithm is significantly more memory demanding than ALS and more expensive per iteration, both in number of operations and computational time [11,21]. Apart from the greater stability with respect to the collinearity and overfactoring, the LM algorithm can be readily modified to treat the case of incomplete data without any imputation. If the loss function is transformed into

$$L(\mathbf{p}) = \|\mathbf{r}^* \text{vec}\mathbf{M}^{(I \times JK)}\|_2^2, \quad (18)$$

where \mathbf{M} is defined as in Eq. (9), the rows in the Jacobian corresponding to the missing observations can be eliminated as the residuals (and thus the loss function) do not change with respect to these elements. This has several advantages: the number of non-zero elements in the Jacobian drops from $3FIJK$ to $3pFIJK$, where $0 < p \leq 1$ is the fraction of non-missing values in the array, thus reducing the memory consumption; furthermore, as \mathbf{J} is extremely sparse, the computation of the products $\mathbf{J}^T\mathbf{J}$ and $\mathbf{J}^T\mathbf{r}$ becomes less and less expensive with the increase of the fraction of missing values.

Under the assumptions of normality with mean zero and identical variance of the residuals, this method of handling the missing values is an analogue to the modified Newton method using the empirical information matrix for maximum likelihood estimation [18].

3. Experimental

Numerous aspects can affect the quality of the estimated PARAFAC model [7]. The aim of this work is to study the behaviour of the two proposed algorithms in the presence of large amounts of missing values with different patterns. Only a few additional properties of the data have been considered in the simulations in order to simplify the setup of the experiments.

The experimental part was conducted in two different stages. The first comprised a Monte Carlo study on synthetic data sets. In the second, the presence of missing values was simulated in three fluorescence data sets of different compositions and degrees of collinearity.

The correct rank of the model was assumed known in all cases, the study of the effect of overfactoring in combination with the presence of missing values is left for future research.

Both algorithms were initialised using the same best fitting of 10 runs of ALS-SI limited to 10 iterations and started with loading matrices of random values. Both algorithms were stopped when the relative decrease in the value of the loss function $(L^{(s)} - L^{(s-1)})/L^{(s-1)}$ was less than 10^{-6} or a predetermined number of iterations was reached (10000 for ALS-SI and 1000 for INDAFAC). For INDAFAC, a second convergence criterion was set at 10^{-8} for the infinite norm of the gradient vector [20].

All the tests were run on a Pentium IV® 2.6-GHz computer with 512 MB memory, working under Windows XP. All the computations were run in MATLAB 6.5 (The Mathworks, Natick, MA, USA). Data sets I and II, the functions for generating the simulated sets and for the PARAFAC-LM algorithms, are available for download at the authors' group webpage (<http://www.models.kvl.dk>, June. 2004). The functions for PARAFAC-ALS with single imputation are part of the N-way toolbox (downloadable at <http://www.models.kvl.dk>, June. 2004).

3.1. Simulated data sets

The Monte Carlo study has been carried out on the basis of 2400 arrays generated considering the following aspects: rank of the array, degree of collinearity of the underlying components, amount of noise, percentage and pattern of missing values. The different conditions are summarised in Table 1. For each setup, 20 replicates were computed to account for minor statistical fluctuations. The dimension of the data sets was $30 \times 30 \times 30$.

The data sets were generated on the basis of Eq. (3). In order to control the collinearity between the underlying components, the loading matrices were generated using the following equation ([11]; here reported with respect to the first mode):

$$\mathbf{A} = \mathbf{V}\mathbf{L} \quad (19)$$

where \mathbf{V} is a column-wise orthonormal $I \times F$ matrix, and \mathbf{L} is the Cholesky factor of a square $F \times F$ matrix \mathbf{U} holding ones on the diagonal and the required cosine of the angle between the loading vectors¹ in the off-diagonal elements [23]. Consequently, $\mathbf{A}^T\mathbf{A} = \mathbf{L}^T\mathbf{V}^T\mathbf{V}\mathbf{L} = \mathbf{L}^T\mathbf{L} = \mathbf{U}$. All the factors were given the same magnitude.

The independent and homoscedastic noise was normally distributed with mean 0. Two desired levels of noise were attained using the following formula [11]:

$$\mathbf{R}^{(I \times JK)} = \frac{\text{Noise}\%}{100 - \text{Noise}\%} \|\mathbf{X}^{(I \times JK)}\|_F \tilde{\mathbf{R}}^{(I \times JK)} \quad (20)$$

where $\tilde{\mathbf{R}}^{(I \times JK)}$ is a matrix of normally distributed (mean 0) random values having a Frobenius norm of 1. Noise% indicates the percentage of noise over the total variation in the array $\mathbf{X} + \mathbf{R}$.

The pattern for the missing values in the array in the RMV case was determined using the first $pIJK$ elements of a random permutation of the integers on the interval $[1, IJK]$. In the RMS case, the tubes (i.e., spectra) were removed in the third mode (Fig. 1b), and the position of the missing tubes was determined using the first pIJ elements of a random permutation of the integers on the interval $[1, IJ]$. In both cases, it was checked that no slab contained only missing

¹ The cosine of the angle between two vectors \mathbf{a} and \mathbf{b} (also referred to as congruence) is defined as: $u(\mathbf{a}, \mathbf{b}) = \cos(\widehat{\mathbf{a}, \mathbf{b}}) = \mathbf{a}^T\mathbf{b}/(\|\mathbf{a}\|\|\mathbf{b}\|)$.

Table 1
Design factors and levels in the Monte Carlo study

Factors	Levels
Percentage of missing values	30, 40, 50, 60, 70
Pattern of missing values	RMV ^a , RMS ^b , SMS ^c
Congruence ^d	0.5, 0.9
Noise ^e	0.5, 2
Model rank	3, 4

^a Randomly missing values.

^b Randomly missing spectra.

^c Systematically missing spectra.

^d Cosine of the angle between the components in the $IJK \times 1$ space.

^e Expressed as a percentage of the total variation (i.e., of $\| \text{vec} \mathbf{X}^{(I \times JK)} \|_2$).

values. In the SMS case, the missing values for each sample were set in two identical triangles at two opposite vertices of each horizontal slab (as in Fig. 1d), their size was defined so that the required fraction of missing was best approximated.

3.2. Real data sets

The two algorithms were tested on three different data sets of fluorescence measurements:

- (1) Twenty-two solutions of four substances (DOPA, hydroquinone, tryptophan, and phenylalanine) were

analysed on a Perking-Elmer LS50 spectrofluorometer [24]. The 13 excitation wavelengths ranged between 245 and 305 nm with steps of 5 nm, whereas in emission the range comprised 131 wavelengths measured between 260 and 390 nm with a step of 1 nm. Three “artificial” data sets were generated out of the single measured one by selecting every third variable in the emission mode in each replicate set [11]. Thus, replicate set one used emission variable number 1, 4, 7, etc., and replicate set two used variable 2, 5, 8, etc. The procedure thus yielded two arrays of size $22 \times 87 \times 13$ and one of size $22 \times 88 \times 13$. The Rayleigh scatter was removed by subtracting from each sample a “model” of the scatter. The Raman scatter was not treated.

- (2) Fifteen solutions of DOPA, hydroquinone, tyrosine, and tryptophan were analysed by means of a Cary Eclipse spectrofluorometer. The excitation mode comprised wavelengths between 230 and 300 nm measured at intervals of 5 nm (15 variables). In emission, the wavelengths varied between 282 and 412 nm with 2-nm steps (66 variables). Full factorial design with two concentration levels per constituent was employed, and six instrumental replicates were measured, thus generating six arrays of size $15 \times 66 \times 15$. The influence of Rayleigh and Raman scatter was minimized by subtracting a blank.

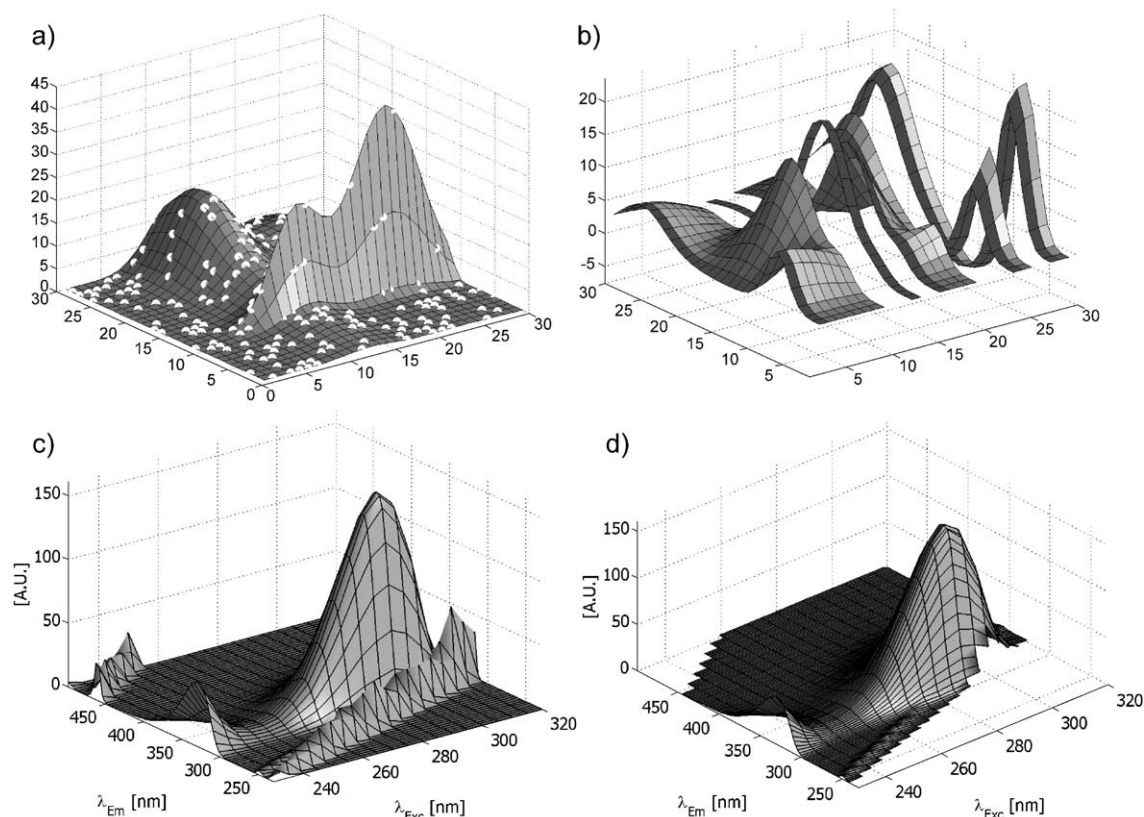


Fig. 1. Patterns of missing values on a single horizontal slab: (a) randomly missing values (RMV); (b) randomly missing spectra (RMS); (c and d) EEM fluorescence landscape (c) and the corresponding systematically missing spectra (SMS) pattern after the Rayleigh scatter removal (d).

- (3) Forty-seven solutions of five compounds (catechol, hydroquinone, indole, tryptophan, and tyrosine) were measured with a Varian Cary Eclipse spectrofluorometer. The emission ranged from 230 to 500 nm with intervals of approximately 2 nm, while the excitation varied between 230 and 305 nm with 5-nm steps [25]. The Rayleigh scatter on the original data set was removed by setting the corresponding elements in each sample to missing [6,7]. The Raman scatter was not treated. The data set was further reduced to a size of $47 \times 80 \times 16$ by selecting the emission wavelengths between 276 and 434 nm in order to remove the largest part of the missing values. A small part (5%), though, remained at the low-emission/high-excitation wavelengths in a region that did not interfere with the resolution of the constituents.

Five replicates with random patterns of missing values (i.e., for RMV and RMS) were generated to account for minor statistical fluctuations. For SMS, five runs were also tested using different starting values but the same pattern.

Preliminary tests showed that for some of the constituents of the real data sets, the predictions worsened already at 30% of missing values in the SMS pattern. Therefore, two more levels were added, and the real data sets were analysed with fractions of missing elements varying from 10% to 70% with increments of 10%.

Table 2 shows the degree of collinearity, the explained variation, and the core consistency [26] of the three data sets of the underlying components obtained with the complete data set. According to Kiers [27], on the basis of the condition number of the loading matrices, both data set I and II can be classified as mildly collinear and data set III as severely collinear. None of them however entirely falls into any category defined according to this criterion. The Jacobian matrix associated with the PARAFAC model contains more information on the numerical difficulty and collinearity of the problem. Due to its rank deficiency, the true condition number cannot be employed for diagnostic purposes. However, the number of numerically zero singular values related to

the scaling indeterminacy is known beforehand to be $2F$; consequently, one can consider the “practical” condition number $\gamma_{\mathbf{J}}$ instead:

$$\gamma_{\mathbf{J}} = \frac{\sigma_1}{\sigma_{(I+J+K-2)F}}, \quad (21)$$

where σ_1 is the largest singular value of \mathbf{J} , and $\sigma_{(I+J+K-2)F}$ is the last non-zero singular value after having taken into account the scaling indeterminacy. As mentioned in Section 2.2.1, the scaling convention affects the value of $\gamma_{\mathbf{J}}$, therefore the loading vectors were scaled so that $\|\mathbf{a}_f\|_2 = \|\mathbf{b}_f\|_2 = \|\mathbf{c}_f\|_2$.

$\gamma_{\mathbf{J}}$ appears as more univocal than the condition numbers of the separate loading matrices when it comes to describing the degree of collinearity. Although no actual threshold can be given nor suggested for $\gamma_{\mathbf{J}}$, a ranking can be clearly observed between the three problems, where data set I is the least collinear, data set III is the most collinear, and data set II is in the middle. Furthermore, $\gamma_{\mathbf{J}}$ also helps in describing the effect of the missing values on the fitting procedure: $\gamma_{\mathbf{J}}$ increases systematically with the percentage of missing values. In particular, for the real data sets, with 70% missing elements, values of $\gamma_{\mathbf{J}}$ in the order of 10^6 were observed upon final convergence.

It should be noted that the factors were extracted from the complete data sets and are thus affected by small non-linearities in the recorded signal that may show up as small interaction terms between the factors. Such phenomena can be theoretically described in terms of model error and might be “quantified” by the core consistency diagnostic [26]. It can clearly be seen that data set III is particularly problematic in this respect: the lower value of the core consistency reflects the presence of relatively unstable components (or of deviations from low-rank trilinearity) and thus may imply a more difficult problem than for data set I and II, where the PARAFAC model is clearly more adequate. The problem of model error is further complicated by the effect of certain patterns of missing values (see Appendix A).

Finally, it can be seen in Table 2 that, in spite of the fact that in one mode the angle between two factors may be small (with cosines up to 0.98), the whole components are in fact rather well separated (mostly as a

Table 2

Diagnostic parameters for the three real data sets: condition numbers for the Jacobian \mathbf{J} and the three loading matrices \mathbf{A} , \mathbf{B} , and \mathbf{C} ; minimum and maximum congruence between factors for the three loading matrices and their Khatri–Rao products; core consistency.

Data set	Condition number				Congruence (min-max)							Core consistency (%) ^b
	J^a	A	B	C	A	B	C	A⊙B	B⊙C	A⊙C	A⊙B⊙C	
I	18.43	2.83	20.4	5.96	0.31–0.49	0.002–0.88	0.15–0.94	0.00–0.38	0.05–0.37	0.00–0.32	4 · 10 ^{−4} –0.32	98.9
II	35.85	8.99	12.24	8.07	0.46–0.50	0.15–0.86	0.55–0.94	0.07–0.42	0.26–0.46	0.06–0.35	0.06–0.35	99.3
III	115.06	5.96	41.93	40.6	0.36–0.55	0.22–0.98	0.52–0.97	0.11–0.52	0.22–0.45	0.08–0.43	0.08–0.43	69.6

^a Computed according to Eq. (21).

^b Computed according to Ref. [26].

consequence of the design in the concentration levels). The same holds for the Khatri–Rao products $\mathbf{A} \odot \mathbf{B}$, $\mathbf{B} \odot \mathbf{C}$, and $\mathbf{A} \odot \mathbf{C}$, whose pseudoinverse is necessary for the computation of the interim solutions (Eqs. (5a)–(5c)). This suggests that the three real data problems in themselves are not particularly difficult provided that all the information is available.

For the real data sets, the position of the missing values was selected on the same basis as for the artificial ones. With specific reference to data set III, the missing values resulting from the removal of the Rayleigh scatter are completely covered by the artificially imposed ones in the SMS case. This was not entirely the case for the other two patterns, where the “naturally” missing values summed up with artificially set ones leading to a fraction of missing values slightly larger than the required value, but still reasonably close to it.

3.3. Criteria of interest

Two main aspects were considered in this work: the statistical quality of the retrieved solutions and the computational aspects of the two algorithms. Specifically, the diagnostics discussed in the following subsections were used.

3.3.1. Recovery capability

One true factor ($\mathbf{z}_f = \mathbf{c}_f \otimes \mathbf{b}_f \otimes \mathbf{a}_f$) is considered as recovered if there is one component ($\hat{\mathbf{z}}_g = \hat{\mathbf{c}}_g \otimes \hat{\mathbf{b}}_g \otimes \hat{\mathbf{a}}_g$) in the fitted solution having a congruence with it greater than a certain threshold. If there are no extreme baselines, a threshold for the single loading vector that guarantees recovery is 0.99; correspondingly, the criterion for the component of three loading vectors may be set at $0.99^3 = 0.97$. The two criteria are not entirely the same; in fact, with the latter, it is possible that a component is considered as recovered also when one of the corresponding loading vectors has a congruence lower than 0.99. In practice, the results using a threshold for the whole component of 0.97 give slightly more optimistic results, but in general the interpretation does not change much. While for the Monte Carlo simulation, the true components are known, for the real data sets the factors found from fitting the model to the original array (i.e., without artificially set missing values) were taken as the correct underlying ones, although this is clearly only an approximation.

The underlying model can be considered as fully recovered (retrieved) if all its factors have been recovered according to a threshold of 0.97. The recovery capability is the percentage full retrievals over the total number of computed models [11].

Because the factor order (permutation) in the solution is not uniquely defined [7,13], all possible permutations of the extracted factors need to be compared with the underlying components to establish full recovery. The correct permutation is defined as the one yielding the highest sum of the

cosines with the “real” one [15,28]. In other words, the “winning” permutation \mathbf{P}_{win} is found as a solution to

$$\mathbf{P}_{\text{win}} = \arg \max_{\mathbf{P}} \text{tr}((\mathbf{A}^T \hat{\mathbf{A}} \mathbf{P})^* (\mathbf{B}^T \hat{\mathbf{B}} \mathbf{P})^* (\mathbf{C}^T \hat{\mathbf{C}} \mathbf{P})) \quad (22)$$

where \mathbf{A} , \mathbf{B} , and \mathbf{C} are the real (column-wise normalised) loading matrices, $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, and $\hat{\mathbf{C}}$ are the estimated (column-wise normalised) loading matrices, \mathbf{P} are all the possible permutation matrices for F columns and $\text{tr}(\mathbf{M})$ indicates the trace of the square matrix \mathbf{M} .

3.3.2. Congruence product

The quality of the solution was also assessed on the basis of the product of the congruences for the whole components or the single loading matrices:

$$\phi_z = \prod_{f=1 \dots F} u(\hat{\mathbf{z}}_f, \mathbf{z}_f), \quad (23)$$

3.3.3. Mean-squared error

The value of the Mean-Squared Error (MSE) for the model parameters. With respect to \mathbf{A} , the MSE is computed as

$$\text{MSE}(\mathbf{A}, \hat{\mathbf{A}}, \mathbf{P}_{\text{win}}, \mathbf{S}_{\mathbf{A}}) = \frac{\|\mathbf{A} - \hat{\mathbf{A}} \mathbf{P}_{\text{win}} \mathbf{S}_{\mathbf{A}}\|_F}{IF} \quad (24)$$

where $\mathbf{S}_{\mathbf{A}}$ is a scaling matrix found as the solution to

$$\arg \min_{\mathbf{S}_{\mathbf{A}}, \mathbf{S}_{\mathbf{B}}, \mathbf{S}_{\mathbf{C}}} (\|\mathbf{A} - \hat{\mathbf{A}} \mathbf{P}_{\text{win}} \mathbf{S}_{\mathbf{A}}\|_F + \|\mathbf{B} - \hat{\mathbf{B}} \mathbf{P}_{\text{win}} \mathbf{S}_{\mathbf{B}}\|_F + \|\mathbf{C} - \hat{\mathbf{C}} \mathbf{P}_{\text{win}} \mathbf{S}_{\mathbf{C}}\|_F) \text{ subject to } \mathbf{S}_{\mathbf{A}} \mathbf{S}_{\mathbf{B}} \mathbf{S}_{\mathbf{C}} = \mathbf{I}_F \quad (25)$$

Such a procedure is necessary because trivial differences in scaling may yield unnecessarily high values for the MSE [11,29].

3.3.4. Loss function value

The value of the loss function is important to establish the capability of the two different algorithms to reach a (global) minimum.

3.3.5. Error in calibration

The presence of the concentration matrices for the three real data sets allows the use of one additional quality diagnostic: the Root-Mean-Squared Error in Calibration (RMSEC) in a linear regression model based on the loadings in the first mode. Only the scores of the component associated to the sought constituent are used in addition to an intercept. Thus, for the f -th constituent:

$$\text{RMSEC}_f = \sqrt{\frac{\|\mathbf{y}_f - \hat{\mathbf{y}}_f\|_2^2}{I}} \quad (26)$$

where $\hat{\mathbf{y}}_f = \mathbf{a}_f \quad 1^+ \mathbf{y}_f$.

3.3.6. Numerical assessments

The efficiency of the two algorithms is assessed in terms of time consumption and number of iterations necessary to reach convergence. Especially with respect to the latter, it is well known that ALS methods require many more (less expensive) iterations [11,21], thus a direct comparison is not feasible. On the other hand, the number of iterations may be of value, in relative terms, to assess the effect of a certain feature on the convergence of the same algorithm.

4. Results and discussion

4.1. Simulated data sets

The first aspect that was considered was the capability of full recovery for the two tested algorithms. In this respect, INDAFAC performed slightly better, managing to retrieve the true underlying components for 77.8% of all the synthetic arrays compared to the 77.3% of ALS-SI.

The feature in the designed sets that affected the most the number of complete recoveries is the pattern of the missing values Table 3. As expected, the RMV case is the easiest to be dealt with, and, apart from a small number of cases, the correct factors are recovered in all the replicates by both algorithms. The RMS proved to be somewhat more difficult to solve, and in a minor fraction of cases full recovery was not attained. The SMS pattern appeared to be much more problematic, with an occurrence of full recoveries that is not comparable with the other two.

The reasons for the lower recoveries of RMS and SMS must be sought both in the convergence to local minima and the presence of artefacts related to slabs with a large fraction of missing values (Fig. 2). In the former case, it was sufficient to restart the algorithm a number of times to yield the correct solutions, whereas in the latter, restarting did not accelerate convergence and yielded solutions with artefacts in the same positions. In these cases, not even initialising the algorithms using the real underlying factors prevented the emergence of artefacts in the solution, which was associated to a lower value of the loss function. In fact, the artefacts of the type shown in Fig. 2 are only indirectly a function of the

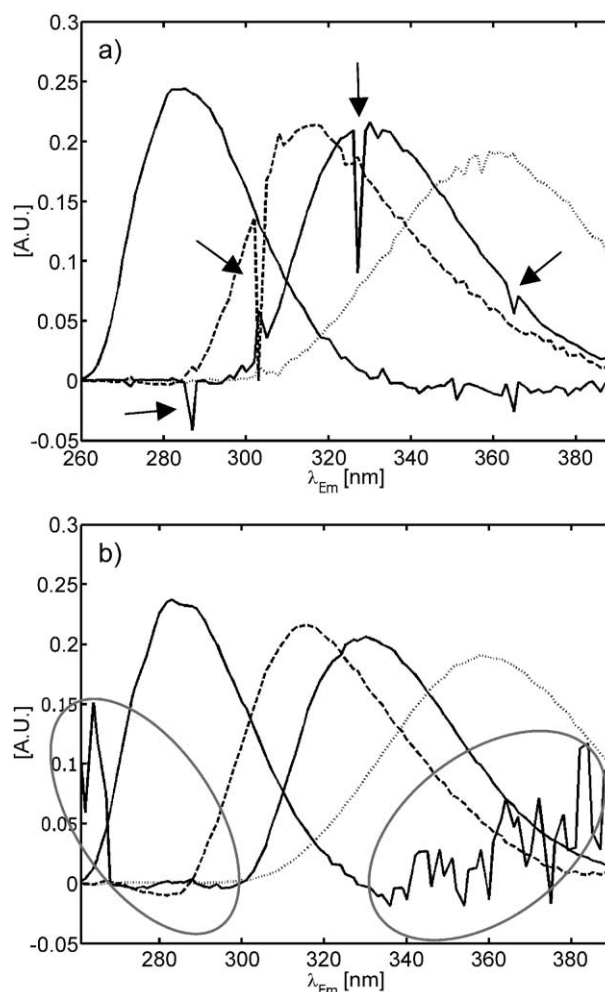


Fig. 2. Different kinds of artefacts in the emission loadings of data set I associated to the RMS (a) and SMS (b) missing values patterns.

fraction of missing values in the corresponding slab; they are determined by the fact that the few values that remain in a slab do not contain enough information with respect to the sought components. Occurrence and magnitude of the artefacts are not easy to predict as they are affected by the different sources of variation (including the non-trilinear ones such as scatter or noise [6]), as well as interactions between factors during convergence allowed for by the

Table 3

Percentage of full recoveries according to a threshold of 0.97 for the simulated data sets with respect to the separate design factors

Pattern	Algorithm	Rank		Congruence		Missing values (%)					Noise (%)	
		3	4	0.5	0.9	30	40	50	60	70	0.5	2.0
RMV ^a	ALS-SI	100.0	99.3	100.0	99.3	100.0	100.0	100.0	99.4	98.8	100.0	99.3
	INDAFAC	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
RMS ^b	ALS-SI	90.5	88.5	97.3	81.8	96.9	94.4	93.8	88.8	73.8	90.8	88.3
	INDAFAC	91.0	89.0	97.3	82.8	97.5	91.9	94.4	90.0	76.3	91.0	89.0
SMS ^c	ALS-SI	41.5	44.0	59.5	26.0	78.8	67.5	38.1	25.0	4.4	46.5	39.0
	INDAFAC	46.5	46.3	59.3	33.5	82.5	71.3	46.3	25.0	6.9	50.3	42.5

^a Randomly missing values.

^b Randomly missing spectra.

^c Systematically missing spectra.

pattern of missing values (see Appendix A). It is beneficial to both speed of convergence and quality of the model to remove whenever possible the problematic slabs, but at the current stage there are few, if any, tools that allow their identification [4,30]. Their development exceeds the purposes of the work, and further studies will be necessary in this direction.

With respect to the other factors in the design, congruence has the most visible effects, along with the fraction of missing values. The Levenberg–Marquadt algorithm performs in general better for highly collinear factors [31], and this property is retained in the presence of missing values (Table 3); the INDAFAC algorithm yields (apart from one single setting) the correct solution more often than ALS-SI. It is once again evident how the RMV case creates very few problems to either of the algorithms, while the other two patterns yield a consistently decreasing number of fully retrieved models. Nevertheless, in the RMS case, even with 70% missing values, the loadings are correctly estimated in more than three cases out of four (for INDAFAC). In the SMS pattern, these percentages decrease to the point that in less than 50% of the cases overall the correct factors are recovered and full recovery hardly ever occurs for 70% missing values. Although, a slight worsening in the quality of the solution could be observed for both MSE (Fig. 3) and ϕ (not shown) as a result of an increase of rank from 3 to 4, this was too small to significantly affect the recovery capability (Table 3). Analogous observations could be made in the RMS case for the noise level, although this factor affects the quality of the solution more than the rank (Fig. 3). For the SMS pattern, the recovery capability is affected also by noise, which most likely influences magnitude and occurrence of artefacts [6].

All these results were confirmed by ANOVA models applied separately to the missing values patterns and having the number of full recoveries over the 20 replicates as response variable. As a result of such ANOVA models, also the interaction between congruence of the underlying factors and fraction of missing values appeared to be significant for ALS-SI for both RMS ($p < 0.012$) and SMS ($p < 0.005$) patterns and for INDAFAC only in the RMS case ($p < 0.023$). Such interaction is not unexpected and means that missing values affect more critically data sets with more collinear components.

If one looks at full recovery in the three modes separately (Table 4), it is apparent how, in the SMS case, the recovery of **A**, **B**, and **C** differs, and that **A** is correctly estimated more often than the other two loading matrices. This is particularly important for calibration purposes, where **A** is used to determine the concentrations. Also in the RMS case, there is an asymmetry in the retrieval of the various modes. Although this is not apparent in the recovery capability relative to this mode, it shows in the quality of the estimations: ϕ_C is larger than ϕ_A and ϕ_B in approximately

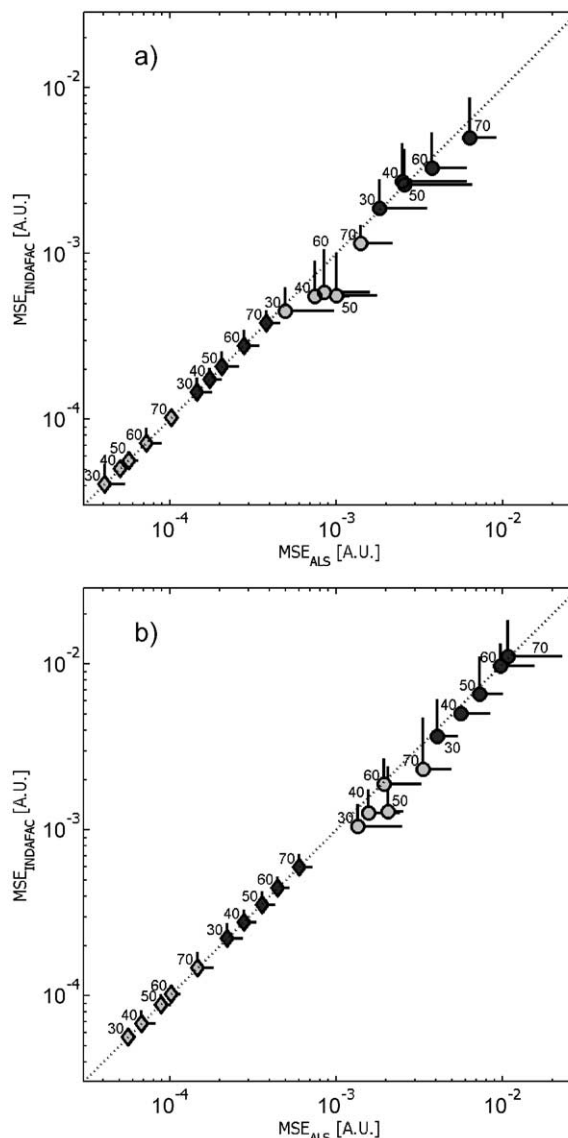


Fig. 3. Effect of various design factors on the median MSE for both algorithms for the RMV pattern: (a) rank 3, (b) rank 4; (◆) low congruence, (○) high congruence, (open symbol) low noise, (closed symbol) high noise. The numbers are the percentage of missing values. The lines departing from each symbol are the standard deviations for the MSE and the algorithm corresponding to the direction of the line.

60% of the cases, whereas in case of symmetry between modes (e.g., in the RMV pattern), this percentage should be around 33%.

The different outcomes related to the RMS and SMS patterns can be in part explained if one considers the matricised form of the array $\underline{\mathbf{X}}$. In the RMS case, **C** spans the column space of $\mathbf{X}^{(K \times LJ)}$, which columns are formed of either completely missing elements or all real ones. Thus, with respect to **C**, complete information is always available, and the difficulties in retrieving the correct solution in this mode are associated only with how collinear are the columns of $\mathbf{B} \odot \mathbf{A}$ after the removal of the rows corresponding to the columns with missing values. Equivalently, for

Table 4

Mode recovery for the different patterns of missing values RMV, RMS, and SMS. All the factors and levels are considered. FR (fully recovered) indicates the percentage of full recoveries for the specified loading matrix according to a congruence threshold of 0.99. BR (best recovered) for loadings matrix **A** is the occurrence of $\phi_A > \max(\phi_B, \phi_C)$ (the values for **B** and **C** are found mutatis mutandis). Symmetry between modes yields identical BR for the three loading matrices, i.e., approximately 33%

Pattern	Algorithm	A		B		C	
		FR (%)	BR (%)	FR (%)	BR (%)	FR (%)	BR (%)
RMV ^a	ALS-SI	99.6	32.8	99.6	33.6	99.8	33.6
	INDAFAC	100.0	33.4	100.0	33.0	99.8	33.6
RMS ^b	ALS-SI	89.1	15.6	89.3	25.8	90.3	58.6
	INDAFAC	89.4	14.9	90.1	25.3	90.3	59.9
SMS ^c	ALS-SI	56.0	70.4	41.9	13.8	42.9	15.9
	INDAFAC	58.1	65.3	46.0	16.4	48.5	18.4

^a Randomly missing values.

^b Randomly missing spectra.

^c Systematically missing spectra.

SMS, **A** spans the column space of $\mathbf{X}^{(I \times JK)}$, which again contains only completely full or completely missing columns. This explains the better performance in the first mode in the SMS case and links the difficulty of the problem to the collinearity of the columns of $\mathbf{C} \odot \mathbf{B}$ once the rows corresponding to the missing elements are removed. Yet, the much greater difficulty in SMS compared to RMS remains largely unexplained by these arguments. The small simulation illustrated Appendix A suggests that the SMS pattern may interfere with trilinearity by allowing for “interactions” between the two loading matrices **B** and **C**.

Figs. 3 and 4 show the effect of the various factors on the quality of the solution in terms of MSE. Fig. 3 describes the behaviour of the two algorithms when both converge to a meaningful solution. The plot is derived on the RMV pattern, but it applies also to the other two when both algorithms converge. The solution of the two algorithms is substantially identical for the low-congruence case (i.e., the symbols lie on the diagonal), although INDAFAC tends to yield a lower value of MSE. This is particularly evident in the high-congruence case. It can also be seen that the effect of rank is very limited compared to that of noise and particularly to that of congruence. These observations are consistent with what was observed on the base of the recovery capability. Note that a value of approximately 10^{-2} for the MSE appears as a good choice to establish full recovery and yields, apart from a limited number of cases, the same results as the aforementioned 0.97 threshold for the congruence. In this sense, convergence to local minima or solutions with large artefacts can be easily identified in Fig. 4. In particular, it can be seen that when INDAFAC does not converge, it yields significantly larger values of MSE (i.e., symbols lying over the diagonal in the plot) than those of ALS-SI in analogous conditions. This may be related to some sort of stabilising effect associated to the fact that (1) imputed values in ALS-SI are indirectly found through linear combinations of the given values, and (2) if one looks at the value of the loss function, INDAFAC clearly outperforms ALS-SI in finding a minimum; it attained the lowest value of the loss function in 97.25% of the cases.

Only in a fraction of these, though, the discrepancy (in relative terms) was larger than 0.01%, namely, in 52.7% (7.3% if one considered a difference of more than 1%) of the 2400 data sets. In all the 2.75% of the cases when ALS-SI found a lower minimum, the difference was larger than 0.01% (2% with a 1% threshold). Thus, when both algorithms do not converge, the better MSE of ALS-SI is likely due to a lower capability of attaining a minimum (albeit a non-relevant one) of this algorithm. Had ALS-SI been able to determine the solution better, this would have been just as bad in terms of MSE as the one obtained with INDAFAC.

Table 5 shows the median number of iterations and of computational time with respect to the percentage and the pattern of the missing values. As expected, the number of iterations increases with the number of missing values and grows more rapidly for ALS-SI than for INDAFAC. It is also very relevant that the number of iterations increases more as a result of the pattern; as an average, the SMS pattern requires 20 times as many iterations as the RMV case for ALS-SI. The ratio for INDAFAC varies with the fraction of missing values, but is at most in the order of 8–10. This trend in the number of iterations hardly ever turns into an advantage in terms of time for the RMV pattern; only in 33% of the cases INDAFAC is faster at 60% missing elements and in 40% of the cases at 70%. For all the other levels of missing values, ALS-SI was faster. Vice versa, in the SMS case, INDAFAC is faster in about one-third of the cases (35%) already at 30% missing and in at least four out of five cases for 50% of missing elements or more.

4.2. Real data sets

Data set I turned out to be most simple to fit, in perfect accordance with the expectations based on $\gamma_{\mathbf{J}}$ and core consistency. Both algorithms recovered the correct components in all the replicate models up to 70% missing values for both the RMV and RMS pattern (not shown). The results also confirmed that SMS is the most challenging among the studied patterns. Artefacts similar to those of

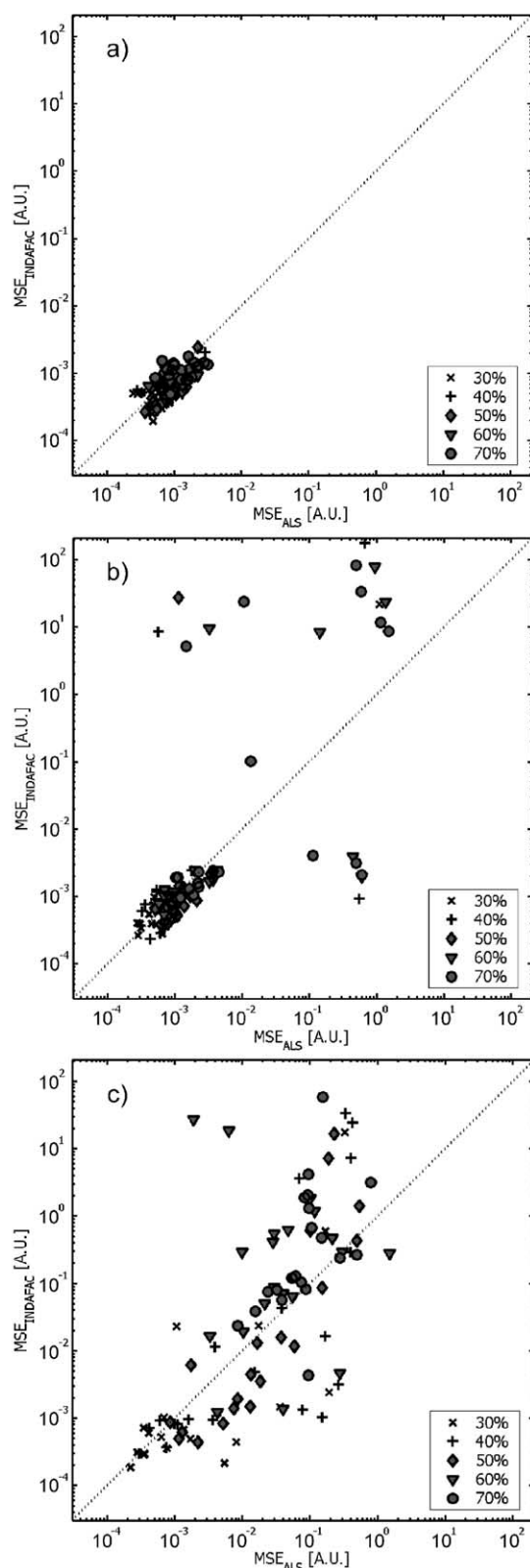


Fig. 4. MSE for the **A** matrix in the rank 3, high congruence and low-noise case. All replicates are displayed. When the MSE exceeds 10^{-2} , the model can be considered as not converged to a meaningful solution. (a) RMV pattern, (b) RMS pattern, (c) SMS pattern.

Fig. 2 appeared in most of the cases, and both INDAPAC and ALS-SI were affected to the same extent. Nevertheless, **A**, which is the most relevant one for calibration purposes, was correctly recovered (again according to a threshold of 0.99 for each of the columns) in all the instances. Fig. 5 shows how the RMSEC varies with respect to the amount of missing elements for ALS-SI. The three studied patterns present remarkable differences. Whereas for both RMV and RMS, the effect of the missing values on the concentration estimates is very small, in the SMS case, there is an improvement in the predictions as the percentage of missing values increases for all constituents apart from phenylalanine. The reason for this appears to be the effect of the Raman scatter ([5,6]; i.e., the small ridge visible in Fig. 1d on the right hand of the main peak). Because of its position and magnitude, the Raman scatter is often left in the data and is given less attention than the Rayleigh one [25]. The setting of some elements to missing in the SMS pattern progressively cancels the Raman rather than removing it all at once from one level of missing to the following. The behaviour of the regression models then follows exactly the pattern described elsewhere [6] for the much more intense Rayleigh scatter; the predictions improve so long as more scatter, but not significant parts of the spectra, is removed. The degree of overlap of the single components determines whether the Raman removal will have an effect. E.g., phenylalanine lies on top of the Raman scatter ridge at the lowest excitation and emission wavelengths. Correspondingly, it is hardly affected by the removal of the Raman or by the setting of further missing values until 60% or 70% is reached. Contrariwise, e.g., tryptophan's main peak lies mostly off the Raman ridge, and its predictions benefit by the increase of the missing values. This observation is also consistent with the fact that both the RMV and RMS patterns have hardly any influence on the quality of the predictions. In theory, it might be that the removal of a certain part of the signal reduces the collinearity between the columns of product **C****B** relative to these two constituents and the other constituents, but this does not seem to be the case here.

Fitting a PARAFAC model to data set II proved to be somewhat more difficult, once again in accordance with the considerations made in Section 3.2. Although the components were correctly recovered up to 70% missing values with RMV patterns, both algorithms failed to retrieve the underlying factors in 10–20% of the cases with the RMS pattern (not shown). The **A** matrix alone was correctly estimated at 70% of missing values only in 70% of the cases. The higher difficulty associated to fitting the PARAFAC model in presence of an SMS pattern is made apparent by the fact that full recovery is no longer guaranteed for the components when 20% of the elements are missing (for the **A** matrix, this happens at 50% of missing values). The variation of the RMSEC as a function of the percentage of missing values is more erratic than in

Table 5
Median number of iterations (# it.) and of computational time for the two algorithms for both simulated and real data sets

Pattern	Data set	Algorithm	Percentage of missing values														
			10		20		30		40		50		60		70		
			# It.	Time (s)	# It.	Time (s)	# It.	Time (s)	# It.	Time (s)	# It.	Time (s)	# It.	Time (s)	# It.	Time (s)	
RMV ^a	Monte Carlo	ALS-SI	–	–	–	–	23	0.5	23	0.5	32	0.7	37	0.9	52	1.2	
		INDAFAC	–	–	–	–	9	3.2	9	3.1	9	3.0	9	2.8	9	2.7	
	I	ALS-SI	40	1.2	40	1.2	46	1.4	50	1.6	56	1.8	72	2.4	101	3.4	
		INDAFAC	9	8.8	9	8.1	9	7.5	9	6.9	9	6.4	9	5.8	10	5.6	
	II	ALS-SI	76	1.6	70	1.5	90	1.9	81	1.8	111	2.5	125	2.8	176	4.1	
		INDAFAC	30	12.1	18	7.8	23	7.8	16	6.1	20	6.3	22	5.9	25	5.9	
	III	ALS-SI	243	12.2	235	13.6	292	15.8	349	19.9	376	20.9	520	29.6	598	35.0	
		INDAFAC	20	30.6	16	24.1	19	24.0	20	23.1	21	21.7	23	21.0	23	19.7	
RMS ^b	Monte Carlo	ALS-SI	–	–	–	–	48	1.0	55	1.3	92	2.1	191	4.6	335	8.2	
		INDAFAC	–	–	–	–	9	3.4	10	3.4	10	3.3	13	3.3	16	3.4	
	I	ALS-SI	38	1.1	42	1.3	48	1.5	50	1.6	77	2.5	348	11.4	718	24.1	
		INDAFAC	9	8.7	9	8.1	9	7.5	9	6.9	10	6.8	11	6.5	14	7.2	
	II	ALS-SI	89	1.8	80	1.7	109	2.4	111	2.4	161	3.6	411	9.3	1114	25.8	
		INDAFAC	37	13.2	27	10.0	28	9.2	29	8.5	32	8.7	37	8.8	83	17.1	
	III	ALS	242	12.9	273	14.6	271	14.9	352	20.0	436	23.1	486	29.1	728	43.1	
		INDAFAC	19	30.0	19	27.3	18	23.3	23	25.8	30	27.2	29	23.2	26	20.0	
	SMS ^c	Monte Carlo	ALS-SI	–	–	–	–	184	4.2	288	6.3	671	15.1	798	18.5	1267	30.4
			INDAFAC	–	–	–	–	15	4.7	18	5.0	35	8.3	46	9.0	78	13.0
I		ALS-SI	38	1.1	93	2.7	216	6.5	368	11.4	446	13.9	1294	42.0	2436	81.1	
		INDAFAC	9	8.7	14	11.7	15	11.2	18	11.9	48	22.3	58	25.3	83	30.3	
II		ALS-SI	104	2.1	320	6.6	785	16.4	1165	25.0	1843	40.3	4052	91.1	7222	164.1	
		INDAFAC	28	10.2	33	12.6	25	8.9	29	8.9	49	12.9	59	12.9	97	19.3	
III		ALS-SI	295	15.1	411	21.3	640	34.9	1770	90.8	3583	192.1	6110	359.1	10000	583.2	
		INDAFAC	25	37.7	27	36.9	30	34.9	37	37.7	73	62.0	81	60.5	131	88.1	

^a Randomly missing values.

^b Randomly missing spectra.

^c Systematically missing spectra.

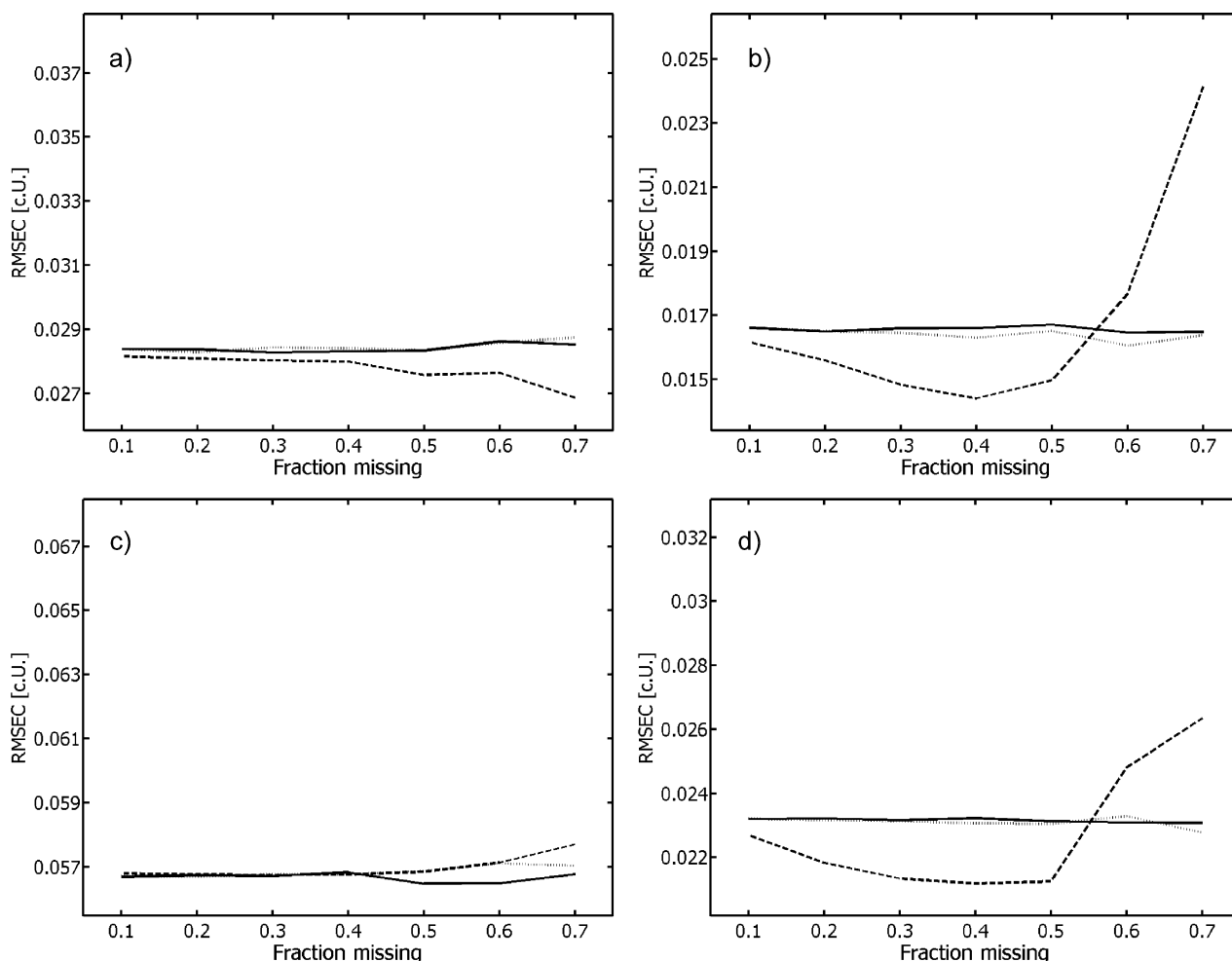


Fig. 5. Median RMSEC for the four constituents of data set I: (a) DOPA, (b) hydroquinone, (c) phenylalanine, and (d) tryptophan. The solid line refers to the RMV pattern, the dotted line to the RMS pattern, and the dashed line to the SMS pattern. All plots refer to the ALS-SI solutions.

data set I, most likely reflecting the higher difficulty, although some specific aspects are retained (Fig. 6). Tryptophan prediction improves with an increasing percentage of missing values, reaching a maximum at 70%. For DOPA, the minimum is reached between 30% and 50%, and then the quality slightly deteriorates. These observations are consistent with the hypothesis of influence of the Raman scatter; further analyses of the raw data showed that the subtraction of a blank is insufficient to completely remove the Raman scatter peaks. For the other constituents, the behaviour is quite the opposite and the predictions worsen considerably along with an increasing amount of missing information. Data set III (Fig. 7) essentially confirms the results illustrated thus far: tryptophan predictions slightly improve with the percentage of missing elements in the SMS case, with a clear worsening starting at 60%; hydroquinone and tyrosine behave analogously to data set II. The different behaviour of the same analyte (particularly of hydroquinone) with respect to the different sets is probably to be associated to more aspects than the sole Raman scatter (e.g., the higher difficulty of the fitting problems or

instrumental effects); nonetheless, the results for the three data sets appear rather consistent.

The two algorithms, in terms of quality of the predictions, performed equivalently for all data sets and patterns. In general, the relative difference of the RMSEC between the two was contained within 0.1%, becoming larger only when the models themselves become very unstable.

With respect to the computational efficiency, the real data sets confirmed all the observations made for the simulations: the number of iterations required for the SMS case is much higher than for the two other patterns, to the point that at 70% missing 10000 iterations were not sufficient for ALS-SI to converge to a solution. Table 5 makes apparent the correctness of the classification of the problems in terms of γ_J : the number of iterations increases going from data set I to II to III, which is clearly the most difficult problem.

With respect to the time consumption, INDAFAC was faster on data sets II and III in 93–100% of the replicates for the SMS pattern starting at about 30–40% missing elements. On the other hand, ALS-SI was faster in the vast majority of

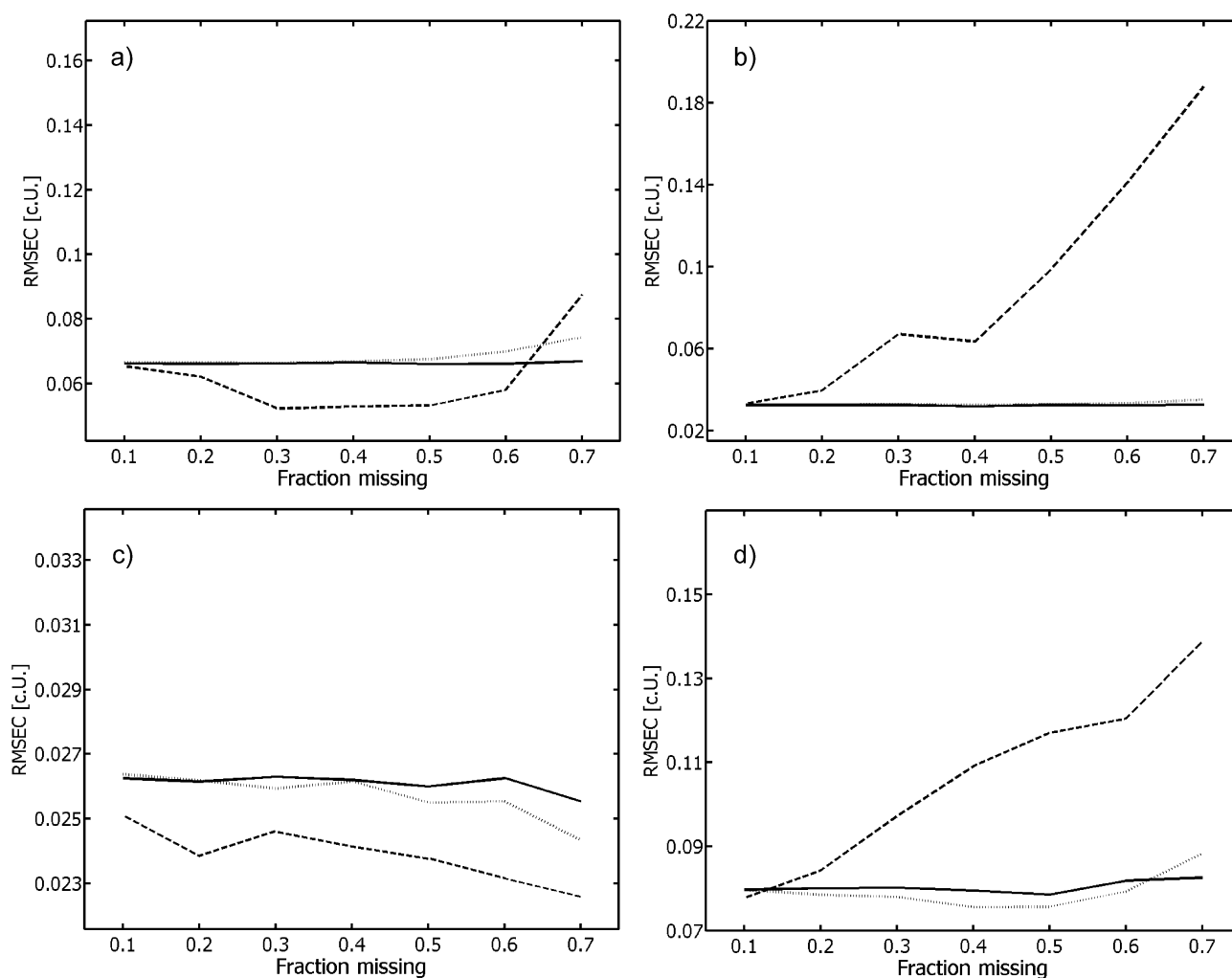


Fig. 6. Median RMSEC for the four constituents of data set II: (a) DOPA, (b) hydroquinone, (c) tryptophan, and (d) tyrosine. The solid line refers to the RMV pattern, the dotted line to the RMS pattern, and the dashed line to the SMS pattern. All plots refer to the ALS-SI solutions.

the cases (for all the data sets and patterns) when 10% or 20% elements were missing.

5. Conclusions

Two algorithms for fitting the PARAFAC model in presence of missing values, ALS with single imputation, and INDAFAC—based on a Levenberg–Marquadt method for non-linear least squares—have been tested by means of a Monte Carlo simulation and on three fluorescence data sets of various complexity. In terms of capability of recovering the correct solution, they performed almost equally, although INDAFAC appeared slightly better for difficult problems (e.g., when the underlying components are very collinear). In terms of time consumption, the derivative-based algorithm is faster when the fraction of missing values exceeds 30% for patterns typical of fluorescence data and 60% when they are uniformly scattered over the array.

A classification was proposed for the possible patterns of missing elements within an array: randomly missing values (RMV) and spectra (RMS), and systematically missing values (SMV) and spectra (SMS). A clear association has been shown between these patterns and the performances of the two algorithms in fitting a PARAFAC model.

The most remarkable result is that a PARAFAC model can be successfully fit even when 70% of the values are missing compared to, for example, PCA on a matrix, for which the limit appears to be in the order of 25–40% [1,4]. The reason for this can be found in the trilinear structure of the PARAFAC model and its added rigidity. In spite of very large fractions of missing values, it was possible to adequately predict the concentration of analytes in synthetic solutions of up to five constituents.

Furthermore, possible explanations were given for the different behaviour of the algorithms with respect to the pattern of missing values, especially with respect to the SMS case, which is by far the most common in, e.g.,

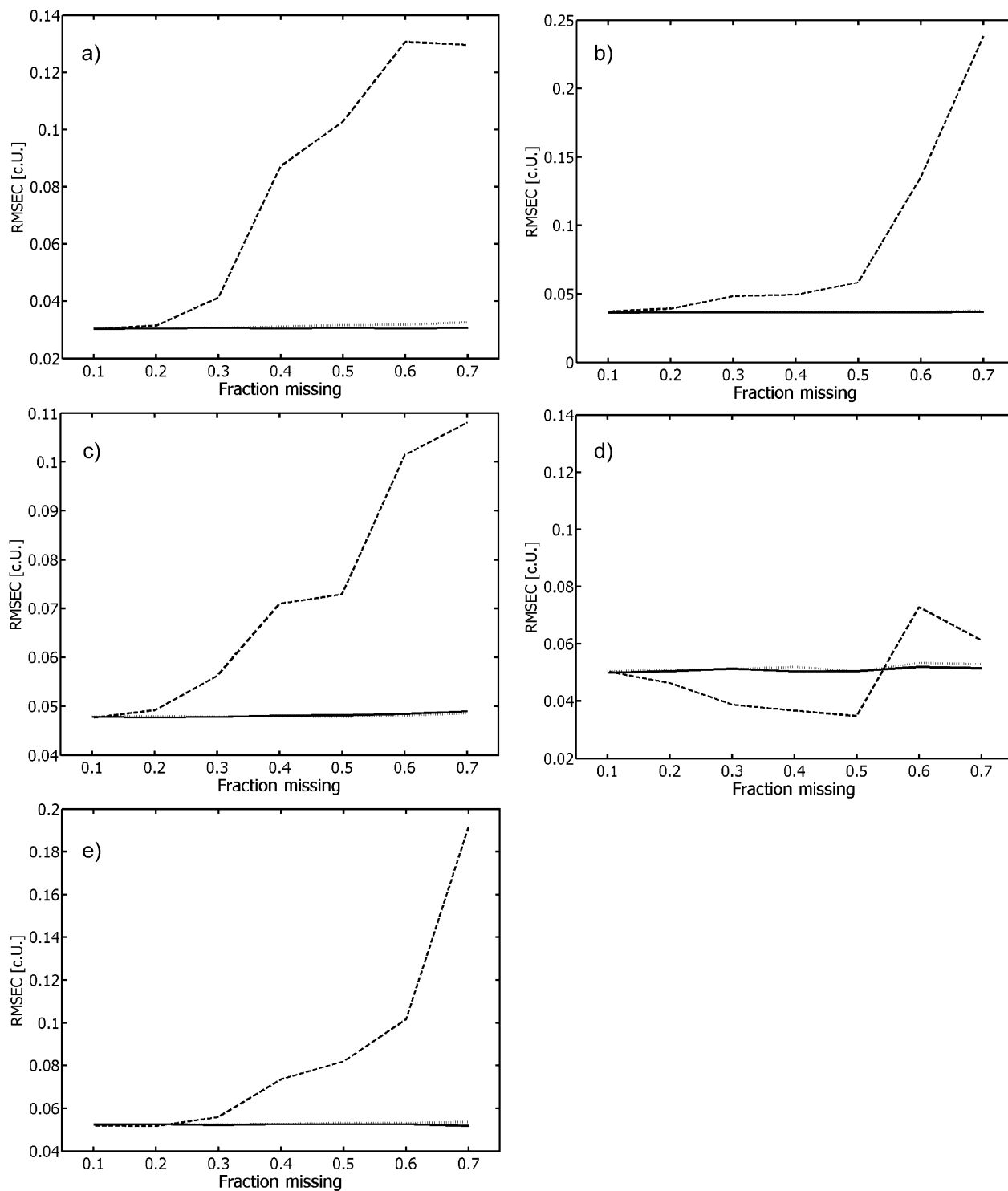


Fig. 7. Median RMSEC for the four constituents of data set III: (a) cathecol, (b) hydroquinone, (c) indole, (d) tryptophan, and (e) tyrosine. The solid line refers to the RMV pattern, the dotted line to the RMS pattern, and the dashed line to SMS pattern. All plots refer to the ALS-SI solutions.

fluorescence spectroscopy. Possibly, this will provide new tools for studying the application of more complex missing values patterns that do not interfere with multilinearity to the same extent as the SMS case presented here [25], or of ad hoc techniques for dealing with missing values and non-multilinear variation [31,32].

Finally, a new and very general tool has been proposed for establishing the difficulty of fitting a PARAFAC model: the Jacobian (practical) condition number γ_J , which accounts not only for the collinearity in any of the loading matrices but also of the “interaction” between the model and the data it is fitted to.

Acknowledgments

The authors would like to acknowledge F. van den Berg for the fruitful discussion, and Å. Rinnan for providing data set III.

Appendix A. Effect of the SMS pattern on the convergence

A noiseless $2 \times 20 \times 20$ array $\underline{\mathbf{X}}$ was generated where $\mathbf{A}=\mathbf{I}$, \mathbf{b}_1 , and \mathbf{c}_1 were Gaussian curves with $\mu=3$, and $\sigma=2$, \mathbf{b}_2 and \mathbf{c}_2 were Gaussians with $\mu=17$ and $\sigma=2$ (Fig. 8a). The two components are orthogonal, and on the complete array with random initialisation, the algorithm always converged to the correct solution within the first five iterations. On the other hand, if 50% of the values were set to missing according to an SMS pattern (Fig. 8b), both algorithms never converged within the first 10000 iterations. Fig. 8c shows the loading vectors of \mathbf{B} and \mathbf{C} for such problem: small peaks are visible in each loading vector in correspondence with the peak of the other component. These small “ghost” peaks have no

effect on the loss function as they are entirely included in the missing areas (Fig. 8d), on the other hand, they interfere with the trilinear structure, as part of the second component is described by the first and vice versa. The problem becomes apparent if one computes the Tucker core and the core consistency [26] relative to the two solutions. When missing values are present, the core consistency is lower than 100% (99.28% in the case showed in the figure), and the Tucker core, while dominated by the two elements on the “superdiagonal”, contains small values with opposite signs and almost equal magnitude ($1.4 \cdot 10^{-5}$) at positions g_{211} and g_{122} :

$$\mathbf{G} = \begin{bmatrix} 4.0 \cdot 10^{-2} & 4.1 \cdot 10^{-3} & 4.1 \cdot 10^{-3} & -1.4 \cdot 10^{-5} \\ 1.5 \cdot 10^{-5} & 4.3 \cdot 10^{-3} & 4.2 \cdot 10^{-3} & 4.1 \cdot 10^{-2} \end{bmatrix}.$$

As the iterative method proceeds (both ALS-SI and INDACFAC), the decrease in the loss function becomes increasingly small, the loss function tends to zero, and the core consistency to 100%. Repeated tests confirmed this observation. The shape of the ghost peaks in real life would be affected by noise or other nonmultilinear structures in the data [6].

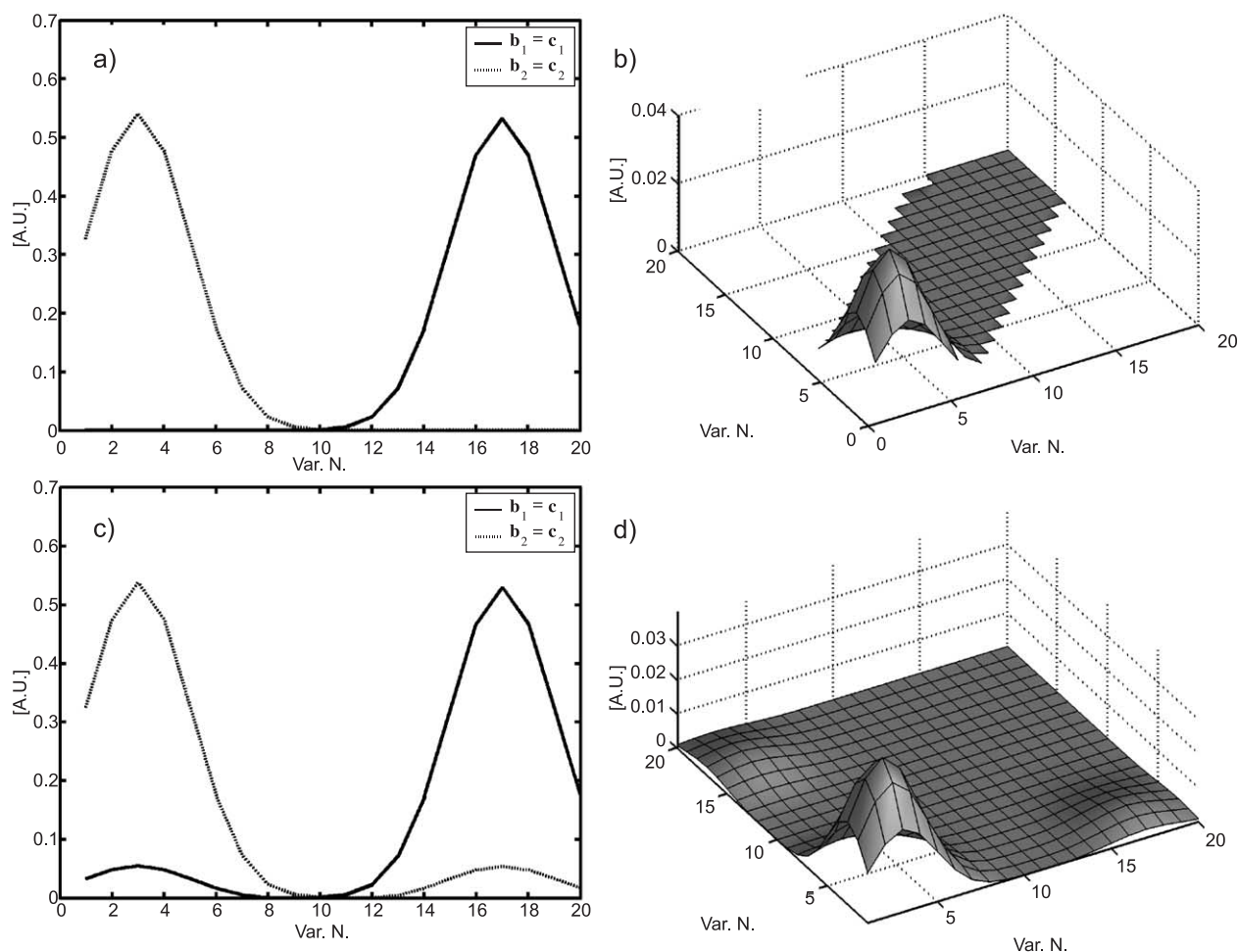


Fig. 8. (a) \mathbf{B} and \mathbf{C} of a noiseless $2 \times 20 \times 20$ array; (b) landscape of the first horizontal slab of $\underline{\mathbf{X}}$; (c) \mathbf{B} and \mathbf{C} of a noiseless $2 \times 20 \times 20$ array with 50% missing values in the SMS pattern; (d) landscape of the first horizontal slab in the PARAFAC model of $\underline{\mathbf{X}}$.

References

- [1] B. Walczak, D.L. Massart, *Chemom. Intell. Lab. Syst.* 58 (2001) 15–27.
- [2] B. Walczak, D.L. Massart, *Chemom. Intell. Lab. Syst.* 58 (2001) 29–42.
- [3] P.R.C. Nelson, P.A. Taylor, J.F. MacGregor, *Chemom. Intell. Lab. Syst.* 35 (1996) 45–65.
- [4] B. Grung, R. Manne, *Chemom. Intell. Lab. Syst.* 42 (1998) 125–139.
- [5] L.G. Thygesen, A. Rinnan, S. Barsberg, J.K.S. Moller, *Chemom. Intell. Lab. Syst.* 71 (2004) 97–106.
- [6] C.M. Andersen, R. Bro, *J. of Chemom.* 17 (2003) 200–215.
- [7] R. Bro, *Multi-way Analysis in the Food Industry. Models, Algorithms, and Applications*, PhD thesis, University of Amsterdam, 1998.
- [8] J.R. Magnus, H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics Rev.*, John Wiley and Sons, New York, NY, USA, 1999.
- [9] L. Vega-Montoto, P.D. Wentzell, *J. Chemom.* 17 (2003) 237–253.
- [10] N.M. Faber, R. Bro, P.K. Hopke, *Chemom. Intell. Lab. Syst.* 65 (2003) 119–137.
- [11] G. Tomasi, R. Bro, A comparison of methods fitting the PARAFAC model, submitted for publication.
- [12] R. Bro, *Chemom. Intell. Lab. Syst.* 38 (1997) 149–171.
- [13] R.A. Harshman, *UCLA Work. Pap. Phon.* 16 (1970) 1–84.
- [14] P. Paatero, *J. Chemom.* 14 (2000) 285–299.
- [15] B.C. Mitchell, D.S. Burdick, *J. Chemom.* 8 (1994) 155–168.
- [16] A.P. Dempster, N.M. Laird, D.B. Rubin, *J. R. Stat. Soc., B, Methodol.* 39 (1977) 1–38.
- [17] R.J.A. Little, D.B. Rubin, *Statistical Analysis with Missing Data*, John Wiley and Sons, New York, NY, USA, 1987.
- [18] G.J. McLachlan, T. Krishnan, *The EM Algorithm and Extensions*, John Wiley and Sons, New York, NY, USA, 1997.
- [19] Å. Björck, *Numerical Methods for Least Squares Problems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996, p. 339.
- [20] K. Madsen, H.B. Nielsen, O. Tingleff, *Methods for Non-linear Least Squares Problems*, Dept. Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2004.
- [21] P. Paatero, *Chemom. Intell. Lab. Syst.* 38 (1997) 223–242.
- [22] P. Paatero, *J. Comput. Graph. Stat.* 8 (1999) 854–888.
- [23] H.A.L. Kiers, J.M.F. Ten Berge, R. Bro, *J. Chemom.* 13 (1999) 275–294.
- [24] D. Baunsgaard, *Factors Affecting 3-way Modelling (PARAFAC) of Fluorescence Landscapes*, The Royal Veterinary and Agricultural University, Frederiksberg, Denmark, 1999.
- [25] Å. Rinnan, *Application of PARAFAC on spectral data*, PhD thesis, The Royal Veterinary and Agricultural University, Frederiksberg, Denmark, 2004.
- [26] R. Bro, H.A.L. Kiers, *J. Chemom.* 17 (2003) 274–286.
- [27] H.A.L. Kiers, *J. Chemom.* 12 (1998) 155–171.
- [28] B.C. Mitchell, D.S. Burdick, *Chemom. Intell. Lab. Syst.* 20 (1993) 149–161.
- [29] J. Riu, R. Bro, *Chemom. Intell. Lab. Syst.* 65 (2003) 35–49.
- [30] K.R. Gabriel, S. Zamir, *Technometrics* 21 (1979) 489–498.
- [31] P.K. Hopke, P. Paatero, H. Jia, R.T. Ross, R.A. Harshman, *Chemom. Intell. Lab. Syst.* 43 (1998) 25–42.
- [32] R. Bro, N.D. Sidiropoulos, A.K. Smilde, *J. Chemom.* 16 (2002) 387–400.

Paper III

Use of the properties of the Khatri-Rao product for
rapid computation of Jacobian, Hessian and gradi-
ent of the PARAFAC model

G. Tomasi

In preparation.

Use of the properties of the Khatri-Rao product for the rapid computation of Jacobian, Hessian and gradient of the PARAFAC model.

Giorgio Tomasi*

The Royal Veterinary and Agricultural University (KVL), Denmark

1. Introduction

The PARAFAC/CANDECOMP model was introduced in 1970 (Carroll and Chang, 1970; Harshman, 1970) as a method to extract information from a three-way array $\underline{\mathbf{X}}$ of dimensions $I \times J \times K$. The original and still most employed algorithm to fit this model is based on the *Alternating Least Squares* (ALS) approach (Bro, 1998). PARAFAC-ALS is relatively easy to implement, constraints of various sorts can be applied and it can be straightforwardly extended to higher orders (Bro, 1998; Carroll and Chang, 1970). Conversely, this algorithm has several well known shortcomings (Mitchell and Burdick, 1994; Paatero, 1999; Hopke et al., 1998; Kiers, 1998; Bro, 1998), which contributed to giving rise to numerous alternative methods (Faber et al., 2003; Tomasi and Bro, 2006).

Standard nonlinear least squares optimisation algorithms such as the Levenberg-Marquadt method can be more efficient in attaining a solution than PARAFAC-ALS (Tomasi and Bro, 2006; Tomasi and Bro, 2005; Hopke et al., 1998; Paatero, 1997), but have seldom been employed because of higher demands in terms of computational resources (Paatero, 1997; Wang and Hopke, 2001; Bro, 1998). In particular, the cross-product $\mathbf{J}^T \mathbf{J}$ and products with a vector of the Jacobian matrix \mathbf{J} are necessary for most nonlinear least squares solvers, but for the PARAFAC case \mathbf{J} is far too large for many problems of practical interest, even when its sparsity is exploited for its storage (Wang and Hopke, 2001). This represents an evident limit for some implementations like the *Multilinear Engine* (ME) (Paatero, 1999), a program for fitting multilinear models, and amongst them PARAFAC, based on a Preconditioned Conjugate Gradients (PCG) algorithm (Wang and Hopke, 2001).

On the other hand, \mathbf{J} is also redundant and structured, which suggests that its products could be efficiently calculated without recurring to its explicit computation and to sparse matrices. Element-wise formulae for some PARAFAC Jacobian's products do exist and are employed by the Equation Oriented System (EOS) (Wang and Hopke, 2001) and, for the three-way case, by the PMF3 method (Positive Matrix Factorisation for 3-way arrays), which is, in essence, a slightly modified Levenberg-Marquadt algorithm (Paatero, 1997). In particular, the former advocates a number of rules to compute the products of the Jacobian (or of its transpose) with a vector to yield the vectors necessary to the PCG algorithm on which EOS relies. However, owing to modern computers' architecture, using so-called block-algorithms and algorithms rich in higher-order operations (i.e. matrix/matrix multiplications) is necessary to yield optimal performances in terms of computational efficiency (Golub and Van Loan, 1996). This is even more important in the MATLAB environment, in which routines based on explicit loops are much slower than programs based on optimised built-in functions (The Mathworks, 2002). Thus, for example, the actual computations of the Jacobian products with a vector are performed in EOS by a compiled routine (Wang and Hopke, 2001).

A first step towards efficient algorithms was made by Liu & Sidiropoulos (Liu and Sidiropoulos, 2001), who developed, albeit for other purposes, some formulae for the computation of two products of the Jacobian matrix (viz. $\mathbf{J}^T \mathbf{J}$ and $\mathbf{J}^T \mathbf{v}$) for the three-way case. Their results are here obtained and generalised to the N -way case on the basis of certain properties of the Khatri-Rao product (Rao and Mitra, 1971) that are described in the appendix to this work. The same properties allow for a generalisation of the formulae for weighted least squares fitting criterion employed by PMF3, to express the rules of EOS for the products of the Jacobian with a vector in terms of full matrices and to derive concise equations for the rapid computation of the second derivatives of the PARAFAC model and of the true Hessian matrix. Additionally, the chain rule for matrix functions (Magnus and Neudecker, 1999) is used to significantly reduce the workload for the two products $\mathbf{J}^T \mathbf{v}$ and $\mathbf{J} \mathbf{v}$ and to formalise the use of 'sub-expressions', which were introduced with ME (Paatero, 1999).

All such formulae are tested on simulated problems of various size to assess the cost per iteration of some common nonlinear least squares solvers (namely Levenberg-Marquadt, damped Newton and nonlinear conjugate gradients) compared to that of PARAFAC-ALS.

1.1 Notation

Defining an N -way array $\underline{\mathbf{X}}$ of dimensions $I_1 \times \dots \times I_N$ as:

$$\underline{\mathbf{X}} = \{x_{i_1 \dots i_N} \mid i_n = 1, \dots, I_n, n = 1, \dots, N\},$$

a rank F PARAFAC model can be written as

$$x_{i_1 \dots i_N} = \sum_{f=1}^F \left(\prod_{n=1}^N a_{i_n f}^{(n)} \right) + r_{i_1 \dots i_N}, \quad (1)$$

where $r_{i_1 \dots i_N}$ denotes the residuals for the single elements.

The model parameters $a_{i_n f}^{(n)}$ are typically grouped in N loading matrices \mathbf{A}_n (one per mode of the array)

$$\mathbf{A}_n = \{a_{i_n f}^{(n)} \mid i_n = 1, \dots, I_n, f = 1, \dots, F\},$$

whose f -th column and i_n -th row are identified by $\mathbf{a}_{n,f}$ and $\bar{\mathbf{a}}_{n,i_n}$ respectively.

A very convenient notation emerges from matricising the N -way array, i.e., by rearranging its elements to obtain a matrix of suitable dimensions (Bro, 1998;Kiers, 2000). Using the loading matrices, the matricised PARAFAC model can be expressed as

$$\mathbf{X}^{(I_1 \times I_2 \dots I_N)} = \mathbf{A}_1 (\mathbf{A}_N \odot \dots \odot \mathbf{A}_2)^T + \mathbf{R}^{(I_1 \times I_2 \dots I_N)}, \quad (2)$$

where \odot denotes the column-wise Khatri-Rao product (Rao and Mitra, 1971) and $(I_1 \times I_2 \dots I_N)$ defines the way the array $\underline{\mathbf{X}}$ is matricised (Bro, 1998;Kiers, 2000).

The model can also be expressed in a vectorised fashion (Bro, 1998), which is more convenient for some of the formulae derived in this work. Thus:

$$\text{vec } \mathbf{X}^{(I_1 \times I_2 \dots I_N)} = (\mathbf{A}_N \odot \dots \odot \mathbf{A}_1) \mathbf{1}_F + \text{vec } \mathbf{R}^{(I_1 \times I_2 \dots I_N)}, \quad (3)$$

where $\mathbf{1}_F$ is an $F \times 1$ vector of ones, the vec operator concatenates the columns of a matrix (Magnus and Neudecker, 1999) and the vector $\text{vec } \mathbf{X}^{(I_1 \times I_2 \dots I_N)} \equiv \mathbf{x}$ has $\prod_{n=1}^N I_n \equiv M$ elements. Note that the vectorisation of a matricised array is independent of where the \times sign appears in the parentheses so long as the order of the dimensions remains unchanged. Thus,

$$\text{vec } \mathbf{X}^{(I_1 \times I_2 I_3 \dots I_{N-1} I_N)} = \text{vec } \mathbf{X}^{(I_1 I_2 \times I_3 \dots I_{N-1} I_N)} = \dots = \text{vec } \mathbf{X}^{(I_1 I_2 I_3 \dots I_{N-1} \times I_N)}.$$

Another useful equation expresses the model for a three-way array $\underline{\mathbf{X}}$ in terms of the matrices obtained by fixing one of the indexes. For example:

$$\mathbf{X}_{i_3} = \mathbf{A}_1 \mathbf{D}_{i_3} \mathbf{A}_2^T + \mathbf{R}_{i_3}, \quad (4)$$

where \mathbf{X}_{i_3} is the matrix obtained by fixing the index in the third mode to i_3 and \mathbf{D}_{i_3} is a diagonal matrix holding the elements of the i_3 -th row of \mathbf{A}_3 .

In order to shorten the notation, a matrix \mathbf{Z} is defined as

$$\mathbf{Z} = \mathbf{A}_N \odot \dots \odot \mathbf{A}_1,$$

and the subscripts $-n$ and $-nn'$ (or, in case of ambiguity, $-\{n, n'\}$) are used to identify matrix products and notations including all indexes from 1 to N apart from the n -th and n' -th. Therefore:

$$\begin{aligned} \mathbf{Z}_{-n} &= \mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \odot \dots \odot \mathbf{A}_1 \\ \mathbf{Z}_{-nn'} &= \mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \odot \dots \odot \mathbf{A}_{n'+1} \odot \mathbf{A}_{n'-1} \odot \dots \odot \mathbf{A}_1 \\ I_{-n} &= I_2 \dots I_{n-1} I_{n+1} \dots I_N \\ I_{-nn'} &= I_2 \dots I_{n-1} I_{n+1} \dots I_{n'-1} I_{n'+1} \dots I_N \end{aligned}$$

Note that the order of the subscripts is relevant and the indexes are always in descending order for \mathbf{Z} matrices and ascending for the dimensions.

The subscripts n and nn' are used to concisely specify the way an N -way array is matricised and denote the modes that represent the rows in the matricised array. Thus:

$$\begin{aligned}\mathbf{X}_n &= \mathbf{X}^{(I_n \times I_{-n})} \\ \mathbf{R}_n &= \mathbf{X}_n - \mathbf{A}_n \mathbf{Z}_{-n}^T \\ \mathbf{X}_{nn'} &= \mathbf{X}^{(I_n I_{n'} \times I_{-nn'})} \\ \mathbf{R}_{nn'} &= \mathbf{X}_{nn'} - (\mathbf{A}_{n'} \odot \mathbf{A}_n) \mathbf{Z}_{-nn'}^T\end{aligned}$$

Additionally, the matrices \mathbf{E} , \mathbf{E}_{-n} and $\mathbf{E}_{-nn'}$ are defined as

$$\mathbf{E} \equiv (\mathbf{A}_1^T \mathbf{A}_1) * \dots * (\mathbf{A}_N^T \mathbf{A}_N) \equiv \mathbf{A}_n^T \mathbf{A}_n * \mathbf{E}_{-n} \equiv \mathbf{A}_n^T \mathbf{A}_n * \mathbf{A}_{n'}^T \mathbf{A}_{n'} * \mathbf{E}_{-nn'},$$

where $*$ indicates the Hadamard (i.e. element-wise) product (Magnus and Neudecker, 1999), and $\mathbf{K}_{I_n F}$, \mathbf{M}_n and $\mathbf{M}_{nn'}$ identify the commutation matrices that perform the following operations (Kapteyn et al., 1986; Magnus and Neudecker, 1999):

$$\begin{aligned}\text{vec } \mathbf{A}_n &= \mathbf{K}_{I_n F} \text{vec } \mathbf{A}_n^T \\ \text{vec } \mathbf{X}_1 &= \mathbf{M}_n \text{vec } \mathbf{X}_n^T \\ \text{vec } \mathbf{X}_1 &= \mathbf{M}_{nn'} \text{vec } \mathbf{X}_{nn'}^T\end{aligned}$$

Finally, the symbol \otimes will denote the Kronecker product (Magnus and Neudecker, 1999).

1.2 Derivative based methods for nonlinear least squares problems

Fitting the PARAFAC model in the least squares sense means to solve the nonlinear minimisation problem

$$\arg \min_{\mathbf{A}_1, \dots, \mathbf{A}_N} L(\mathbf{A}_1, \dots, \mathbf{A}_N) = \arg \min_{\mathbf{A}_1, \dots, \mathbf{A}_N} \left[(\text{vec } \mathbf{X}_1 - \mathbf{Z} \mathbf{1}_F)^T (\text{vec } \mathbf{X}_1 - \mathbf{Z} \mathbf{1}_F) \right]. \quad (5)$$

In some cases, e.g., when some elements in \mathbf{X} are missing, a weighted least squares approach should be used and problem (5) is transformed in

$$\arg \min_{\mathbf{A}_1, \dots, \mathbf{A}_N} L(\mathbf{A}_1, \dots, \mathbf{A}_N) = \arg \min_{\mathbf{A}_1, \dots, \mathbf{A}_N} \left[(\text{vec } \mathbf{X}_1 - \mathbf{Z} \mathbf{1}_F)^T \mathbf{W} (\text{vec } \mathbf{X}_1 - \mathbf{Z} \mathbf{1}_F) \right], \quad (6)$$

where \mathbf{W} is a diagonal matrix (Tomasi and Bro, 2005; Andersson et al., 1999; Vega-Montoto and Wentzell, 2003; Bro et al., 2002).

Typically, both (5) and (6) are solved through an *Alternating Least Squares* algorithm, whereby the nonlinear problem is split in smaller linear ones by fixing all the loading matrices apart from one (Smilde et al., 2004). Other common nonlinear least squares algorithms (i.e., Newton, Quasi-Newton, Gauss-Newton and PCG methods) are iterative and are based on the fact that any sufficiently smooth function $L(\mathbf{p})$ can be approximated in a point $\mathbf{p} + \Delta \mathbf{p}$ by a Taylor-series expansion truncated at the quadratic term:

$$L(\mathbf{p} + \Delta \mathbf{p}) \approx L(\mathbf{p}) + \mathbf{g}^T \Delta \mathbf{p} + \frac{1}{2} \Delta \mathbf{p}^T \mathbf{H} \Delta \mathbf{p} \equiv \tilde{L}(\Delta \mathbf{p}), \quad (7)$$

where \mathbf{g} is the gradient and \mathbf{H} is the Hessian of the function $L(\mathbf{p})$ at the point \mathbf{p} and their elements are defined respectively as:

$$\mathbf{g}_k = \frac{\partial L(\mathbf{p})}{\partial p_k} \text{ and } h_{kk'} = \frac{\partial^2 L(\mathbf{p})}{\partial p_k \partial p_{k'}}.$$

In the PARAFAC case, the vector \mathbf{p} of parameters can be defined as $\mathbf{p} \equiv \text{vec}[\mathbf{A}_1^T | \mathbf{A}_2^T | \dots | \mathbf{A}_N^T]$ and has $F \sum_{n=1}^N I_n \equiv K$ elements.

For least squares problems, Hessian and gradient assume the following form (the dependence on \mathbf{p} is omitted for clarity) (Gill et al., 1986; Björck, 1996):

$$\mathbf{g} = \mathbf{J}^T \mathbf{r} \quad (8a)$$

and

$$\mathbf{H} = \mathbf{J}^T \mathbf{J} + \sum_{m=1}^M r_m \mathbf{H}_m \equiv \mathbf{J}^T \mathbf{J} + \mathbf{Q}, \quad (8b)$$

where $\mathbf{J}(\mathbf{p})$ is the Jacobian of \mathbf{r} , i.e. an $M \times K$ matrix whose elements are defined as

$$j_{mk} = \frac{\partial r_m}{\partial p_k}$$

and \mathbf{H}_m is the Hessian calculated at r_m .

A stationary point of $\tilde{L}(\Delta \mathbf{p})$, satisfies the linear system (Madsen et al., 2004):

$$\mathbf{H} \Delta \mathbf{p} = -\mathbf{g}. \quad (9)$$

The solution of system (9) provides a direction $\Delta \mathbf{p}$ that is used to update the current approximation of \mathbf{p} :

$$\mathbf{p}^{(s)} = \mathbf{p}^{(s-1)} + \Delta \mathbf{p}, \quad (10)$$

where s denotes the iteration number. If the residuals are sufficiently small and the linear approximation holds, \mathbf{Q} can be ignored and $\mathbf{J}^T \mathbf{J}$ can be used as an approximation of the Hessian. This is the theoretical basis for the Gauss-Newton algorithm (Madsen et al., 2004). If, on the other hand, the full Hessian is used, the algorithm is referred to as Newton method.

A sufficient condition for the update to be a descent direction (i.e., leading to a decrease in the loss function value), is the positive-definiteness of the Hessian or of its approximation, which is in general not guaranteed (Frandsen et al., 2004). For example, owing to the scaling indeterminacy of PARAFAC (Paatero, 1997), \mathbf{J} is rank deficient and has $(N-1)F$ zero singular values, which can cause serious problems for the convergence of the Gauss-Newton algorithm (Gill et al., 1986; Björck, 1996). Therefore, in order to obtain a globally convergent algorithm, the updates are found as the solution of

$$(\mathbf{H} + \lambda \mathbf{D}) \Delta \mathbf{p} = -\mathbf{g}, \quad (11)$$

where \mathbf{D} is a positive definite diagonal matrix, λ is called the damping parameter and should be large enough to yield positive-definiteness. For the Gauss-Newton algorithm it is sufficient to substitute $\mathbf{J}^T \mathbf{J}$ for \mathbf{H} (Frandsen et al., 2004). In the latter case, the algorithm is sometimes referred to as Levenberg-Marquardt method.

When problem (6) need be solved the same methods mentioned thus far can be used, but equations (8a) and (8b) need be modified into (Magnus and Neudecker, 1999):

$$\mathbf{g} = \mathbf{J}^T \mathbf{W} \mathbf{r} \quad (12a)$$

$$\mathbf{H} = \mathbf{J}^T \mathbf{W} \mathbf{J} + \sum_{m=1}^M r_m w_m \mathbf{H}_m, \quad (12b)$$

where \mathbf{W} is a diagonal matrix. Equations (12a-b) show how fast routines for $\mathbf{J}^T \mathbf{W} \mathbf{J}$ and $\mathbf{g} = \mathbf{J}^T \mathbf{W} \mathbf{r}$ when \mathbf{W} is diagonal are also required to avoid the explicit computation of the Jacobian or element-wise calculations.

The Conjugate Gradients (CG) method for nonlinear least squares problems (Gill et al., 1986), albeit slower than Gauss-Newton or Newton methods, is often the sole available algorithm for large systems (e.g., when the number of parameters is in the order of several thousands) (Frandsen et al., 2004; Gill et al., 1986). Moreover, a variant of the CG algorithm may be used to solve linear systems like (11) when the application of a direct method (e.g., the Cholesky decomposition) is not feasible (Björck, 1996). When only an approximate solution to system (9) is found by means of an iterative procedure, the algorithm is referred to as Truncated Newton method (Gill et al., 1986). The main advantage the (linear or nonlinear) CG algorithm is that no large matrices need be stored or computed as the Jacobian is always treated as a product with a vector (Frandsen et al., 2004; Gill et al., 1986). Thus, when such products can be efficiently reckoned, CG methods become particularly appealing. Preconditioning is used to speed up the convergence (Gill et al., 1986) as well as to enforce certain constraints like non-negativity (Paatero, 1999). However, for the purposes of this paper, it is only relevant that the two products of the Jacobian $\mathbf{J}^T \mathbf{v}$ and $\mathbf{J} \mathbf{v}$ are required by this method and that for the nonlinear case, the CG algorithm requires a line search procedure that may entail several loss function evaluations

(Madsen et al., 2004). With respect to the latter aspect, it is noteworthy that for PARAFAC models an analytical solution is available to compute the optimal step length given a certain direction of update (Franc, 1992). As for other operations considered in this work, exploiting the properties of the Khatri-Rao product allows to extend the procedure to the N -way case and to yield great savings compared to the original formulation. For sake of completeness, this result is reported in Appendix 4.

2. Theory

2.1 The Jacobian matrix \mathbf{J}

The Jacobian matrix for the matrix function $\mathbf{R}_1 = \mathbf{X}_1 - \mathbf{A}_1 \mathbf{Z}_{-1}^T$ with respect to $\mathbf{A}_1, \dots, \mathbf{A}_N$ can be obtained by computing the differential of $\text{vec } \mathbf{R}_1$ (Magnus and Neudecker, 1999)

$$\begin{aligned} d(\text{vec } \mathbf{R}_1) &= d(\text{vec } \mathbf{X}_1 - \mathbf{Z} \mathbf{1}_F) = -\sum_{f=1}^F d\mathbf{z}_f = -\sum_{f=1}^F d(\mathbf{a}_{N,f} \odot \dots \odot \mathbf{a}_{1,f}) = \\ &= -\sum_{f=1}^F \sum_{n=1}^N (\mathbf{a}_{N,f} \odot \dots \odot d\mathbf{a}_{n,f} \odot \dots \odot \mathbf{a}_{1,f}) = -\sum_{f=1}^F \sum_{n=1}^N \mathbf{M}_n (d\mathbf{a}_{n,f} \odot \mathbf{z}_{-n,f}) =, \\ &= -\sum_{f=1}^F \sum_{n=1}^N \mathbf{M}_n \text{vec}[\mathbf{z}_{-n,f} (d\mathbf{a}_{n,f}^T)] = -\sum_{n=1}^N \mathbf{M}_n \sum_{f=1}^F (\mathbf{I}_{I_n} \otimes \mathbf{z}_{-n,f}) d(\mathbf{a}_{n,f}) \end{aligned} \quad (13)$$

where \mathbf{I}_{I_n} is the identity matrix of order I_n , the effect of \mathbf{M}_n on the order of the terms in the Khatri-Rao product is explained in Lemma 2 of Appendix 1, and the relation $\text{vec } \mathbf{ABC} = (\mathbf{C}^T \otimes \mathbf{A}) \text{vec } \mathbf{B}$ (Magnus and Neudecker, 1999) is used for the last equality. It could then be noticed that

$$\begin{aligned} \sum_{f=1}^F (\mathbf{I}_{I_n} \otimes \mathbf{z}_{-n,f}) d(\mathbf{a}_{n,f}) &= [\mathbf{I}_{I_n} \otimes \mathbf{z}_{-n,1} | \dots | \mathbf{I}_{I_n} \otimes \mathbf{z}_{-n,F}] d[\text{vec}(\mathbf{A}_n)] = \\ &= (\mathbf{I}_{I_n} \otimes \mathbf{Z}_{-n}) \mathbf{K}_{I_n F}^T \mathbf{K}_{I_n F} d[\text{vec}(\mathbf{A}_n^T)] = \\ &= (\mathbf{I}_{I_n} \otimes \mathbf{Z}_{-n}) d(\text{vec } \mathbf{A}_n^T) \end{aligned} \quad (14)$$

Thus,

$$d(\text{vec } \mathbf{R}_1) = -\sum_{n=1}^N \mathbf{M}_n (\mathbf{I}_{I_n} \otimes \mathbf{Z}_{-n}) d(\text{vec } \mathbf{A}_n^T) \quad (15)$$

and the Jacobian can be partitioned as (Magnus and Neudecker, 1999):

$$\mathbf{J} = [\mathbf{J}_1 | \dots | \mathbf{J}_N], \quad (16)$$

where

$$\mathbf{J}_n = -\mathbf{M}_n (\mathbf{I}_{I_n} \otimes \mathbf{Z}_{-n}). \quad (17)$$

According to (17), the number of non-zero elements in \mathbf{J}_n is equal to $I_n F I_{-n} = FM$ and for the whole \mathbf{J} is equal to NFM . Consequently, the density β of the Jacobian depends only on the order of the array N and on its size (viz.,

$$\beta = \frac{NFM}{MK} = N \left(\sum_{n=1}^N I_n \right)^{-1} \text{) and is typically very low. For normal sized problems, the storage of } \mathbf{J} \text{ and its explicit use for}$$

computing, e.g., $\mathbf{J}^T \mathbf{J}$ are feasible only using sparse matrices. Sparse direct methods (e.g., QR) may appear attractive to solve system (10), but the matrix $\mathbf{J}^T \mathbf{J}$ is almost full (only the diagonal blocks are not) and the Jacobian itself has the strong Hall property (Björck, 1996). Thus, the Cholesky factor will also be almost dense, ruling out such methods (Björck, 1996). However, equation (17) also shows that \mathbf{J} is redundant and suggests that its structure could be exploited to reduce computational load and storage.

2.2 Jacobian products

2.2.1 $\mathbf{J}^T \mathbf{J}$

The matrix $\mathbf{J}^T \mathbf{J}$ is symmetric and can be partitioned in N^2 blocks:

$$\mathbf{J}^T \mathbf{J} = \begin{bmatrix} \mathbf{J}_1^T \mathbf{J}_1 & \cdots & \mathbf{J}_1^T \mathbf{J}_N \\ \vdots & \ddots & \vdots \\ \mathbf{J}_N^T \mathbf{J}_1 & \cdots & \mathbf{J}_N^T \mathbf{J}_N \end{bmatrix}. \quad (18)$$

The blocks on the diagonal can be readily computed from equation (17)

$$\mathbf{J}_n^T \mathbf{J}_n = (\mathbf{I}_{I_n} \otimes \mathbf{Z}_{-n}^T) \mathbf{M}_n^T \mathbf{M}_n (\mathbf{I}_{I_n} \otimes \mathbf{Z}_{-n}) = (\mathbf{I}_{I_n} \otimes \mathbf{Z}_{-n}^T \mathbf{Z}_{-n}) = (\mathbf{I}_{I_n} \otimes \mathbf{E}_{-n}). \quad (19)$$

Each off-diagonal block $\mathbf{J}_n^T \mathbf{J}_{n'}$ has dimensions $(I_n F \times I_{n'} F)$ and can be partitioned in $F \times F$ blocks as

$$\mathbf{J}_n^T \mathbf{J}_{n'} = \begin{bmatrix} \mathbf{J}_{n,1}^T \mathbf{J}_{n',1} & \cdots & \mathbf{J}_{n,1}^T \mathbf{J}_{n',I_{n'}} \\ \vdots & \ddots & \vdots \\ \mathbf{J}_{n,I_n}^T \mathbf{J}_{n',1} & \cdots & \mathbf{J}_{n,I_n}^T \mathbf{J}_{n',I_{n'}} \end{bmatrix}, \quad (20)$$

where block $\mathbf{J}_{n,i_n}^T \mathbf{J}_{n',i_{n'}}$ pertains to the parameters of the i_n -th row of \mathbf{A}_n and of the $i_{n'}$ -th row of $\mathbf{A}_{n'}$. As can be seen from equation (17),

$$\mathbf{J}_{n,i_n} = -\mathbf{M}_n (\mathbf{e}_{i_n} \otimes \mathbf{Z}_{-n}) = -\mathbf{M}_n (\mathbf{e}_{i_n} \mathbf{1}_F^T \odot \mathbf{Z}_{-n}) = \mathbf{A}_N \odot \cdots \odot \mathbf{A}_{n+1} \odot \mathbf{e}_{i_d} \mathbf{1}_F^T \odot \mathbf{A}_{n-1} \odot \cdots \odot \mathbf{A}_1. \quad (21)$$

Hence, using equation (A.7), one obtains that

$$\mathbf{J}_{n,i_n}^T \mathbf{J}_{n',i_{n'}} = \mathbf{E}_{-nn'} * \bar{\mathbf{a}}_{n',i_{n'}}^T \bar{\mathbf{a}}_{n,i_n}. \quad (22)$$

One can immediately see that the off-diagonal blocks of $\mathbf{J}^T \mathbf{J}$ are structurally dense, which prevents in many cases the efficient use of sparse algorithms to solve directly the system of normal equations (Paatero, 1997; Tomasi and Bro, 2006). The term $\mathbf{E}_{-nn'}$ is independent of i_n and $i_{n'}$, thus:

$$\mathbf{J}_n^T \mathbf{J}_{n'} = \begin{bmatrix} \bar{\mathbf{a}}_{n',1}^T \bar{\mathbf{a}}_{n,1} & \cdots & \bar{\mathbf{a}}_{n',I_{n'}}^T \bar{\mathbf{a}}_{n,1} \\ \vdots & \ddots & \vdots \\ \bar{\mathbf{a}}_{n',1}^T \bar{\mathbf{a}}_{n,I_n} & \cdots & \bar{\mathbf{a}}_{n',I_{n'}}^T \bar{\mathbf{a}}_{n,I_n} \end{bmatrix} * (\mathbf{1}_{I_n \times I_{n'}} \otimes \mathbf{E}_{-nn'}), \quad (23)$$

where $\mathbf{1}_{I_n \times I_{n'}}$ is a $I_n \times I_{n'}$ matrix of ones.

Computing $\mathbf{J}_n^T \mathbf{J}_{n'}$ using equation (23) requires explicit loops over the I_n and $I_{n'}$ and can be quite slow in MATLAB (The Mathworks, 2002). However, a pattern is clearly visible in the first right end term that could be exploited to accelerate the calculation. Using *Lemma 1*, each small block of the first right hand term, can be written as:

$$\bar{\mathbf{a}}_{n',i_{n'}}^T \bar{\mathbf{a}}_{n,i_n} = \bar{\mathbf{a}}_{n',i_{n'}}^T \otimes \bar{\mathbf{a}}_{n,i_n} = (\bar{\mathbf{a}}_{n',i_{n'}}^T \otimes \mathbf{1}_F^T) * (\bar{\mathbf{a}}_{n,i_n} \otimes \mathbf{1}_F). \quad (24)$$

It follows immediately that equation (23) can be written as:

$$\mathbf{J}_n^T \mathbf{J}_{n'} = \mathbf{B} * \mathbf{C} * (\mathbf{1}_{I_n \times I_{n'}} \otimes \mathbf{E}_{-nn'}), \quad (25)$$

where

$$\mathbf{B} = \begin{bmatrix} \bar{\mathbf{a}}_{n',1}^T \otimes \mathbf{1}_F^T & \cdots & \bar{\mathbf{a}}_{n',I_{n'}}^T \otimes \mathbf{1}_F^T \\ \vdots & \ddots & \vdots \\ \bar{\mathbf{a}}_{n',1}^T \otimes \mathbf{1}_F^T & \cdots & \bar{\mathbf{a}}_{n',I_{n'}}^T \otimes \mathbf{1}_F^T \end{bmatrix} = \mathbf{I}_n \left\{ \begin{bmatrix} \mathbf{A}_{n'}^T \otimes \mathbf{1}_F^T \\ \vdots \\ \mathbf{A}_{n'}^T \otimes \mathbf{1}_F^T \end{bmatrix} \right\} = [\mathbf{1}_{I_n} \otimes (\mathbf{A}_{n'} \otimes \mathbf{1}_F)^T]^T$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{1}_F \otimes \bar{\mathbf{a}}_{n,1} & \cdots & \mathbf{1}_F \otimes \bar{\mathbf{a}}_{n,I_n} \\ \vdots & \ddots & \vdots \\ \mathbf{1}_F \otimes \bar{\mathbf{a}}_{n,I_n} & \cdots & \mathbf{1}_F \otimes \bar{\mathbf{a}}_{n,I_n} \end{bmatrix} = \overbrace{[\mathbf{A}_n \otimes \mathbf{1}_F \quad \cdots \quad \mathbf{A}_n \otimes \mathbf{1}_F]}^{I_{n'}} = (\mathbf{1}_{I_{n'}}^T \otimes \mathbf{A}_n \otimes \mathbf{1}_F).$$

Equations (19) and (25) are the extension to higher orders of the formulas derived by Liu & Sidiropoulos (Liu and Sidiropoulos, 2001). Differently from the latter work, the permutational and scaling indeterminacy are not dealt with in (19) and (25). Therefore, \mathbf{I}_{I_n} and $\mathbf{I}_{I_{n'}}$ are used instead of \mathbf{I}_{I_n-1} and $\mathbf{I}_{I_{n'}-1}$.

It is noteworthy that there is no reduction in the number of operations by using (25) instead of (23). However, the former is more suited for the MATLAB environment as Kronecker products with a vector or a matrix of ones require no actual multiplication and can be produced via appropriate indexing. Moreover, the matrices $\mathbf{A}_n \otimes \mathbf{1}_F$ need only be formed once.

Taking advantage of the symmetry of $\mathbf{J}^T \mathbf{J}$, its computation requires: $2 \sum_n I_n F^2 = 2FK$ operations for computing the $\mathbf{A}_n^T \mathbf{A}_n$, approximately $0.5(N-1)N^2 F^2$ to compute \mathbf{E}_{-n} and $\mathbf{E}_{-nn'}$, and $2(K^2 - F^2 \sum_n I_n^2)$ for the actual reckoning of the off-diagonal blocks, which is clearly the most time consuming step.

2.2.2 $\mathbf{J}^T \mathbf{W} \mathbf{J}$ when \mathbf{W} is diagonal

Equations (19) and (25) are not suited for the product $\mathbf{J}^T \mathbf{W} \mathbf{J}$ and are thus inapplicable when some elements of $\underline{\mathbf{X}}$ are set to missing, a fairly common occurrence for the type of data on which PARAFAC is fitted (Tomasi and Bro, 2005). However, if \mathbf{W} is diagonal and holds the elements of the vector $\mathbf{w} = \text{vec } \mathbf{W}_1$, where $\underline{\mathbf{W}}$ is an array of the same size of $\underline{\mathbf{X}}$, it is possible to compute all the non-zero elements in the partitions $\mathbf{J}_n^T \mathbf{W} \mathbf{J}_{n'}$ without having to explicitly compute the Jacobian and using only full matrices. Using *Lemmas 5, 6 and 7* (cf. Appendix 1), the partitions $\mathbf{J}_{n,i_n}^T \mathbf{W} \mathbf{J}_{n',i_{n'}}$ can be expressed as

$$\mathbf{J}_{n,i_n}^T \mathbf{W} \mathbf{J}_{n',i_{n'}} = \begin{cases} \delta_{i_n i_{n'}} \mathbf{Z}_{-n}^T \mathbf{D}_{\bar{\mathbf{w}}_{i_n}} \mathbf{Z}_{-n} & \text{if } n = n' \\ \bar{\mathbf{a}}_{n',i_{n'}}^T \bar{\mathbf{a}}_{n,i_n} * \mathbf{Z}_{-nn'}^T \mathbf{D}_{\bar{\mathbf{w}}_{i_n i_{n'}}} \mathbf{Z}_{-nn'} & \text{if } n \neq n' \end{cases} \quad (26)$$

where $\mathbf{D}_{\bar{\mathbf{w}}_{i_n}}$ is a diagonal matrix holding the i_n -th row of \mathbf{W}_n and $\mathbf{D}_{\bar{\mathbf{w}}_{i_n i_{n'}}}$ is a diagonal matrix holding the row of $\mathbf{W}_{nn'}$ relative to the two indexes i_n and $i_{n'}$ (i.e., the vectorised sub-array obtained by fixing the indexes in the n -th and n' -th mode to i_n and $i_{n'}$ respectively) and $\delta_{i_n i_{n'}}$ denotes the Kronecker delta.

The computation of $\mathbf{J}_n^T \mathbf{W} \mathbf{J}_{n'}$ can be further accelerated by noticing the similarity between $\mathbf{Z}_{-n}^T \mathbf{D}_{\bar{\mathbf{w}}_{i_n}} \mathbf{Z}_{-n}$ and $\mathbf{Z}_{-nn'}^T \mathbf{D}_{\bar{\mathbf{w}}_{i_n i_{n'}}} \mathbf{Z}_{-nn'}$ and equation (4). $\mathbf{Z}_{-n}^T \mathbf{D}_{\bar{\mathbf{w}}_{i_n}} \mathbf{Z}_{-n}$ expresses the PARAFAC model of the i_n -th slab of a three-way array $\underline{\mathbf{U}}$ of size $F \times F \times I_n$, where the three loading matrices are $\mathbf{A}_1 = \mathbf{A}_2 = \mathbf{Z}_{-n}^T$ and $\mathbf{A}_3 = \mathbf{W}_n$. Likewise, $\mathbf{Z}_{-nn'}^T \mathbf{D}_{\bar{\mathbf{w}}_{i_n i_{n'}}} \mathbf{Z}_{-nn'}$ is the PARAFAC model of one slab of a three-way array $\underline{\mathbf{V}}$ of size $F \times F \times I_n I_{n'}$, where the three loading matrices are $\mathbf{A}_1 = \mathbf{A}_2 \equiv \mathbf{Z}_{-nn'}^T$ and $\mathbf{A}_3 \equiv \mathbf{W}_{nn'}$. Hence, one can define:

$$\begin{aligned} \mathbf{U}^{(F \times F I_n)} &\equiv \mathbf{Z}_{-n}^T (\mathbf{W}_n \odot \mathbf{Z}_{-n}^T)^T &\Rightarrow \mathbf{U}^{(FF \times I_n)} &= (\mathbf{Z}_{-n}^T \odot \mathbf{Z}_{-n}^T) \mathbf{W}_n^T \\ \mathbf{V}^{(F \times F I_n I_{n'})} &\equiv \mathbf{Z}_{-nn'}^T (\mathbf{W}_{nn'} \odot \mathbf{Z}_{-nn'}^T)^T &\Rightarrow \mathbf{V}^{(FF \times I_n I_{n'})} &= (\mathbf{Z}_{-nn'}^T \odot \mathbf{Z}_{-nn'}^T) \mathbf{W}_{nn'}^T \end{aligned} \quad (27)$$

and rewrite equation (19) as:

$$\mathbf{J}_{n,i_n}^T \mathbf{W} \mathbf{J}_{n',i_{n'}} = \delta_{i_n i_{n'}} \mathbf{U}_{i_n}, \quad (28)$$

where \mathbf{U}_{i_n} denotes the i_n -th frontal slab of $\underline{\mathbf{U}}$. Similarly, equation (25) becomes

$$\mathbf{J}_n^T \mathbf{W} \mathbf{J}_{n'} = \mathbf{B} * \mathbf{C} * \mathbf{V}^{(F I_n \times F I_{n'})}, \quad (29)$$

where \mathbf{B} and \mathbf{C} are defined as in (25).

The computational workload is largely increased by the fact that the cross products of the loading matrices cannot be used throughout $\mathbf{J}^T \mathbf{J}$. Thus, in addition to what is necessary for (29), one needs approximately $2F^2M$ additional operations for each block to calculate \mathbf{U} and \mathbf{V} . The overall load is then in the order of $(N+1)NMF^2 + 2K^2$.

2.2.3 $\mathbf{J}^T \mathbf{v}$

The vector $\mathbf{b} = \mathbf{J}^T \mathbf{v}$ can be partitioned in N blocks corresponding to the columns relative to the different loading matrices in the Jacobian:

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_N \end{bmatrix} = [\mathbf{J}_1 \quad \cdots \quad \mathbf{J}_N]^T \mathbf{v} = \begin{bmatrix} \mathbf{J}_1^T \mathbf{v} \\ \vdots \\ \mathbf{J}_N^T \mathbf{v} \end{bmatrix}. \quad (30)$$

By defining a set of matrices \mathbf{V}_n according to the relation $\text{vec } \mathbf{V}_n^T = \mathbf{M}_n^T \mathbf{v}$, one can calculate \mathbf{b}_n as:

$$\mathbf{b}_n = \mathbf{J}_n^T \mathbf{v} = -(\mathbf{I}_{I_n} \otimes \mathbf{Z}_{-n})^T \mathbf{M}_n^T \mathbf{v} = -(\mathbf{I}_{I_n} \otimes \mathbf{Z}_{-n}^T) \text{vec } \mathbf{V}_n^T = -\text{vec}(\mathbf{Z}_{-n}^T \mathbf{V}_n^T) = -\text{vec}(\mathbf{V}_n \mathbf{Z}_{-n})^T. \quad (31)$$

Hence, to compute \mathbf{b} , it is sufficient to appropriately matricise the multi-way array $\underline{\mathbf{V}}$ of size $I_1 \times \dots \times I_N$ obtained by rearranging the elements of vector \mathbf{v} and to multiply it by the corresponding set of Khatri-Rao products. The computational cost of the product $\mathbf{J}^T \mathbf{v}$ is $2 \sum_n FI_{-n} I_n = 2NMF$.

The chain rule for matrix functions (Magnus and Neudecker, 1999) allows for a considerable reduction in the total workload when at least the size in one mode exceeds the rank of the fitted model. Thus, by denoting with $\mathbf{J}_{2\dots N}$ the partition of the Jacobian of $\mathbf{X}_1 - \mathbf{A}_1 \mathbf{Z}_{-1}^T$ relative to the parameters of the loading matrices $\mathbf{A}_2, \dots, \mathbf{A}_N$ and by using the chain rule, one can write

$$\mathbf{J}_{2\dots N} = -\tilde{\mathbf{J}}_{\mathbf{Z}_{-1}}^T \tilde{\mathbf{J}}_{2\dots N}, \quad (32)$$

where $\tilde{\mathbf{J}}_{\mathbf{Z}_{-1}}$ indicates the Jacobian of $\mathbf{A}_1 \mathbf{Z}_{-1}^T$ with respect to \mathbf{Z}_{-1}^T and $\tilde{\mathbf{J}}_{2\dots N}$ is the Jacobian of \mathbf{Z}_{-1}^T with respect to $\mathbf{A}_2, \dots, \mathbf{A}_N$.

If performed explicitly (32) would not bring any computational advantage (Paatero, 1999), but, when $\mathbf{J}^T \mathbf{v}$ is required, one yields (cf. Appendix 2)

$$\mathbf{J}^T \mathbf{v} = \left([\mathbf{J}_1 | \tilde{\mathbf{J}}_{\mathbf{Z}_{-1}}^T \tilde{\mathbf{J}}_{2\dots N}] \right)^T \mathbf{v} = \begin{bmatrix} \mathbf{J}_{\mathbf{A}_1}^T \mathbf{v} \\ -\tilde{\mathbf{J}}_{2\dots N}^T (\tilde{\mathbf{J}}_{\mathbf{Z}_{-1}}^T \mathbf{v}) \end{bmatrix} = - \begin{bmatrix} \text{vec}(\mathbf{V}_1 \mathbf{Z}_{-1})^T \\ \text{vec}(\tilde{\mathbf{V}}_2 \tilde{\mathbf{Z}}_{-2})^T \\ \vdots \\ \text{vec}(\tilde{\mathbf{V}}_N \tilde{\mathbf{Z}}_{-N})^T \end{bmatrix}, \quad (33)$$

where $\tilde{\mathbf{V}}$ is the $F \times I_2 \times \dots \times I_N$ array defined by the relation $\tilde{\mathbf{V}}_1 \equiv \mathbf{A}_1^T \mathbf{V}_1$ and $\tilde{\mathbf{Z}}_{-n} \equiv (\mathbf{Z}_{-\{1,n\}} \odot \mathbf{I}_F)$.

The computational advantage derives from the fact that (33) involves the product and the rearrangement of a matrix where I_1 is reduced to F and that $\tilde{\mathbf{Z}}_n$ is sparse with density F^{-1} . Since algorithmically the different modes of an N -way array are treated identically, it is obvious that the greatest advantage is yielded by reducing the largest mode and for higher ratios between size of the array and model's rank. Equation (33) formalises the use of sub-expressions of ME with respect to the product $\mathbf{J}^T \mathbf{v}$ (Paatero, 1999).

In the special case where $\mathbf{v} = \text{vec } \mathbf{R}_1$, the vector \mathbf{b} is equal to the gradient $\mathbf{g} = \mathbf{J}^T \mathbf{r}$ and

$$\mathbf{g}_n = -\text{vec}(\mathbf{R}_n \mathbf{Z}_{-n})^T, \quad (34)$$

but, since $\mathbf{R}_n = \mathbf{X}_n - \mathbf{A}_n \mathbf{Z}_{-n}^T$, eq. (34) can be simplified as:

$$\mathbf{g}_n = -\text{vec}[(\mathbf{X}_n - \mathbf{A}_n \mathbf{Z}_{-n}^T) \mathbf{Z}_{-n}]^T = -\text{vec}[\mathbf{X}_n \mathbf{Z}_{-n} - \mathbf{A}_n \mathbf{E}_{-n}]^T. \quad (35)$$

Equivalently, defining

$$\tilde{\mathbf{R}}_1 \equiv \mathbf{A}_1^T [\mathbf{X}_1 - \mathbf{A}_1 \mathbf{Z}_{-1}^T] = \mathbf{A}_1^T \mathbf{X}_1 - \mathbf{E}_1 \mathbf{Z}_{-1}^T, \quad (36)$$

equation (33) can be rewritten as:

$$\mathbf{J}^T \mathbf{v} = - \begin{bmatrix} \text{vec}(\mathbf{X}_1 \mathbf{Z}_{-1} - \mathbf{A}_1 \mathbf{E}_{-1})^T \\ \text{vec}(\tilde{\mathbf{X}}_2 \tilde{\mathbf{Z}}_{-2} - \mathbf{A}_2 \mathbf{E}_{-2})^T \\ \vdots \\ \text{vec}(\tilde{\mathbf{X}}_N \tilde{\mathbf{Z}}_{-N} - \mathbf{A}_N \mathbf{E}_{-N})^T \end{bmatrix}, \quad (37)$$

where $\tilde{\mathbf{X}}_1 \equiv \mathbf{A}_1^T \mathbf{X}_1$.

Consequently, the residuals are not necessary to calculate the gradient and one could spare approximately $M(2F+1)$ operations per each calculation of the gradient. On the other hand, although the savings would grow rapidly with the dimensions of \mathbf{X} and with the model's rank, the practical advantage for fitting algorithms seems limited, since the residuals are computed at each iteration to evaluate the loss function. Equation (35) was obtained by Franc (Franc, 1992) in a different fashion and not using explicitly the properties of the Khatri-Rao product.

This product obviously encompasses the case $\mathbf{J}^T \mathbf{W} \mathbf{v}$, which should always be computed as $\mathbf{J}^T (\mathbf{W} \mathbf{v})$ to avoid an unnecessary matrix/matrix multiplication. Moreover, if \mathbf{W} is diagonal and holds the elements of the vector \mathbf{w} , the fastest method is indeed $\mathbf{J}^T (\mathbf{w} * \mathbf{v})$. Thus, the gradient for problem (6) when \mathbf{W} is diagonal, is computed as:

$$\mathbf{g}_n = -\text{vec}(\mathbf{R}'_n \mathbf{Z}_{-n})^T, \quad (38)$$

where $\mathbf{R}' = \mathbf{R} * \mathbf{W}$. Note that, owing to \mathbf{W} , the simplifications leading to equations (35) and (37) are not possible, but when many values are set to missing \mathbf{W} can be stored as sparse and there can be a considerable reduction in the computational load.

Finally, it is noteworthy that no array permutation is actually necessary to compute the products $\mathbf{V}_n \mathbf{Z}_{-n}$. The definition of a new class of matricised arrays $\bar{\mathbf{V}}_n \equiv \mathbf{V}^{(I_n \times I_{n+1} \dots I_N I_1 \dots I_{n-1})}$ for $n = 1 \dots N$, which has the property:

$$\text{vec} \bar{\mathbf{V}}_n = \text{vec} \bar{\mathbf{V}}_{n-1}^T, \quad (39)$$

helps showing that only a transposition and the simple rearrangement of the elements in a new matrix are necessary. The multiple Khatri-Rao matrix corresponding to this matricisation is $\bar{\mathbf{Z}}_{-n} \equiv \mathbf{A}_{n-1} \odot \dots \odot \mathbf{A}_1 \odot \mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1}$. This expedient is used in the N -way toolbox (Andersson and Bro, 2000) and can be particularly advantageous for large arrays. Unfortunately, it seems to be available only when either the columns or the rows of the matricised array interested by similar products refer to only one mode. Therefore, the same procedure is not applicable for reducing the time expense for the off-diagonal blocks of $\mathbf{J}^T \mathbf{J}$ or to compute the nonlinear part of the Hessian \mathbf{Q} (cf. section 2.3).

2.2.4 $\mathbf{J} \mathbf{v}$

The vector $\mathbf{c} = \mathbf{J} \mathbf{v}$ can also be more easily calculated partitioning \mathbf{v} and \mathbf{J} in N blocks relative to the different modes:

$$\mathbf{c} = [\mathbf{J}_1 \quad \dots \quad \mathbf{J}_N] \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_N \end{bmatrix} = \sum_{n=1}^N \mathbf{J}_n \mathbf{v}_n. \quad (40)$$

By rearranging the elements of \mathbf{v}_n in a matrix \mathbf{V}_n of dimensions $F \times I_n$ according to the relation $\text{vec} \mathbf{V}_n = \mathbf{v}_n$, one can calculate \mathbf{c} as:

$$\mathbf{c} = -\sum_{n=1}^N \mathbf{M}_n (\mathbf{I}_{I_n} \otimes \mathbf{Z}_{-n}) \mathbf{v} = -\sum_{n=1}^N \mathbf{M}_n (\mathbf{I}_{I_n} \otimes \mathbf{Z}_{-n}) \text{vec} \mathbf{V}_n = -\sum_{n=1}^N \mathbf{M}_n \text{vec}(\mathbf{Z}_{-n} \mathbf{V}_n). \quad (41)$$

The vectors $\mathbf{M}_n \text{vec}(\mathbf{Z}_{-n} \mathbf{V}_n)$ can be computed without any actual permutation if one considers that

$$\begin{aligned}
\mathbf{c}_n &\equiv -\mathbf{M}_n \text{vec}(\mathbf{Z}_{-n} \mathbf{V}_n) = -\mathbf{M}_n \left(\mathbf{V}_n^T \odot \mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \odot \dots \odot \mathbf{A}_1 \right) \mathbf{1}_F = \\
&= -\left(\mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{V}_n^T \odot \mathbf{A}_{n-1} \odot \dots \odot \mathbf{A}_1 \right) \mathbf{1}_F = \\
&= -\mathbf{A}_1 \left(\mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{V}_n^T \odot \mathbf{A}_{n-1} \odot \dots \odot \mathbf{A}_2 \right)^T
\end{aligned} \tag{42}$$

If one uses (42), the computational cost of this product is of $2NM$ for each \mathbf{c}_n and $(N-1)M$ for their summation. Nonetheless, this workload can be reduced by considering again the chain rule for matrix functions, similarly to what was done for the product $\mathbf{J}^T \mathbf{v}$. As shown in Appendix 2, \mathbf{c} can be calculated as:

$$\mathbf{c} = -\text{vec} \left\{ \mathbf{V}_1^T \mathbf{Z}^T + \mathbf{A}_1 \left[\sum_{n=2}^N \left(\mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{V}_n^T \odot \mathbf{A}_{n-1} \odot \dots \odot \mathbf{A}_2 \right)^T \right] \right\} \tag{43}$$

Like for $\mathbf{J}^T \mathbf{v}$, the computational advantage of using (43) instead of (42) is expected to grow with the ratio between the largest dimension of the array (here taken equal to I_1) and the rank of the model F . Finally, should \mathbf{WJv} be computed, the product should be performed as $\mathbf{W}(\mathbf{Jv})$, and, in case of diagonal \mathbf{W} , as $\mathbf{w} * (\mathbf{Jv})$, where \mathbf{w} is the vector of the diagonal elements of \mathbf{W} . Again, it is worth mentioning that, if many values in the array are set the missing, it could be beneficial to store the vector \mathbf{w} as sparse.

2.3 Hessian

One of the reasons why Gauss-Newton is preferred to the Newton method is that the computation of the exact second derivatives implies an additional (typically large) workload. In theory, though, the exact Hessian represents an advantage when the residuals are large or when the cost function is particularly nonlinear (Gill et al., 1986; Björck, 1996). One advantage of the PARAFAC model is that its multilinear structure makes the second order term \mathbf{Q} extremely sparse and computationally relatively inexpensive.

From eq. (8b), its element $q_{kk'}$ is calculated as

$$q_{kk'} = \sum_{m=1}^M r_m \frac{\partial^2 r_m}{\partial p_k \partial p_{k'}} = \mathbf{r}^T \left(\frac{\partial^2 \mathbf{r}}{\partial p_k \partial p_{k'}} \right) = \mathbf{r}^T \frac{\partial \mathbf{j}_k}{\partial p_{k'}}. \tag{44}$$

But, from equation (17), and considering $p_k \equiv a_{i_n, f}^{(n)}$ and $p_{k'} \equiv a_{i_{n'}, f'}^{(n')}$,

$$\begin{aligned}
\frac{\partial \mathbf{j}_k}{\partial p_{k'}} &= \frac{\partial}{\partial a_{i_{n'}, f'}^{(n')}} \left(-\mathbf{a}_{N, f} \odot \dots \odot \mathbf{a}_{n+1, f} \odot \mathbf{e}_{i_n} \odot \mathbf{a}_{n+1, f} \odot \dots \odot \mathbf{a}_{1, f} \right) = \\
&= \delta_{ff'} (1 - \delta_{nn'}) \left(-\mathbf{a}_{N, f} \odot \dots \odot \mathbf{a}_{n+1, f} \odot \mathbf{e}_{i_n} \odot \mathbf{a}_{n+1, f} \odot \dots \odot \mathbf{a}_{n'+1, f} \odot \mathbf{e}_{i_{n'}} \odot \mathbf{a}_{n'+1, f} \odot \dots \odot \mathbf{a}_{1, f} \right)
\end{aligned} \tag{45}$$

Thus, $q_{kk'}$ is not zero only when p_k and $p_{k'}$ belong to different loading matrices and to the same factor f . Moreover, \mathbf{Q} can be partitioned in N^2 blocks as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{0} & \mathbf{Q}_{21}^T & \dots & \mathbf{Q}_{N1}^T \\ \mathbf{Q}_{21} & \mathbf{0} & \dots & \mathbf{Q}_{N2}^T \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Q}_{N1} & \mathbf{Q}_{N2} & \dots & \mathbf{0} \end{bmatrix} \tag{46}$$

and all the elements of the partition $\mathbf{Q}_{nn'}$ can be obtained in one operation using Lemma 4 and equations (44) and (45):

$$\mathbf{U}^{(I_n I_{n'} \times F)} = \mathbf{R}_{nn'} \mathbf{Z}_{-nn'} \tag{47}$$

In order to see how to rearrange the elements of $\mathbf{U}^{(I_n I_{n'} \times F)}$ to obtain $\mathbf{Q}_{nn'}$, one should further partition the latter in $I_n I_{n'}$ blocks of size $F \times F$, each relative to the i_n -th and $i_{n'}$ -th rows of \mathbf{A}_n and $\mathbf{A}_{n'}$. Since the second derivatives are zero if $f \neq f'$, all such blocks are diagonal. It is easy to demonstrate that the non-zero elements of $\mathbf{Q}_{nn'}$ have the same indexes of the non-zeros of the matrix $(\mathbf{1}_{I_n} \mathbf{1}_{I_{n'}})^T \otimes \mathbf{I}_F$. Storing such indexes allows for the rapid formation of the sparse matrix $\mathbf{Q}_{nn'}$. Furthermore, exploiting the symmetry of \mathbf{Q} , only half of the indexes and of its nonzero elements need be stored, with additional savings in terms of memory consumption.

Table 1. Tested Routines and design factors

<i>Vector/Matrix</i>	<i>Routines (function name)</i>	<i>Factor</i>	<i>Level</i>
J	<ul style="list-style-type: none"> - Index (J_index)¹ - Sparse (J_sparse)¹ - Sparse + Index (J_sparse_part)² 	Array Size	25×25×25 50×50×50 100×100×100
		Rank	3, 6, 9
J^TJ, J^TWJ	<ul style="list-style-type: none"> - Element-wise ($JTJ_el, JTWJ_el$) - Sparse ($JTJ_sparse, JTWJ_sparse$) - Full ($JTJ_full, JTWJ_full$) 	Array Size	25×25×25 50×50×50 100×100×100
		Rank	3, 6, 9
H	<ul style="list-style-type: none"> - Index (H_index) - Sparse (H_sparse) 	Array Size	25×25×25 50×50×50 100×100×100
		Rank	3, 6, 9
J^Tv, Jv	<ul style="list-style-type: none"> - Full (JTv_full, Jv_full) - Chain Rule (JTv_chain, Jv_chain) 	Array Size	80×80×80 140×140×140 200×200×200
		Rank	4,8,12,..., 48

¹ used for all dimensions and ranks apart from 100×100×100 and model rank 6 and 9

² used for arrays of dimensions 100×100×100 and model rank 6.

The additional computational cost for the nonlinear part of the Hessian is $2 \sum_{\substack{n,n' \\ n \neq n'}} MF = (N-1)NMF$. As can be seen,

for the three-way case, **Q** is about as expensive to compute in terms of number of operations as the gradient, which suggest the feasibility of Newton algorithms for fitting PARAFAC for problems where the Hessian can be stored.

3. Experimental

The purpose of this work is to investigate the speed of different routines for computing the Jacobian, some of its products and the Hessian of a PARAFAC model and not to evaluate the convergence rate or the performances of the associated algorithms. Thus, the different procedures have been tested on arrays and loading matrices of random numbers and appropriate size without any specific N -linear structure. The tested routines and the corresponding outputs, the design factors and their levels are shown in Table 1. All the routines are only outlined here for reasons of space. They have all been made available for download at www.models.kvl.dk/source. All the simulations have been performed on the same dedicated machine, mounting a Pentium 4, 2.8 GHz, 1GB RAM operating under Windows XP Professional and using MATLAB 7.0.1.

The size of the Jacobian impedes the use of full matrices for its storage even for relatively small arrays and simple models; for example already for a $50 \times 50 \times 50$ array and a rank three PARAFAC model, the full Jacobian requires approximately 429MB in double precision. The same matrix stored as sparse requires slightly less than 13MB. Thus, the only feasible implementation to compute **J** is based on sparse matrices. In order to reduce the computation time, the indexes for the non-zero elements of the Jacobian are computed separately by J_index and passed to the main routine (J_sparse). Note however, that this solution is feasible only up to a point since the storage of the indexes requires an amount of memory on the same order of magnitude as the Jacobian itself, even if they are stored as unsigned long integers (i.e. if eight bytes are used to index each element). Therefore, for larger problems the indexes have to be reckoned at every iteration with a considerable increase in the computational load. The function J_sparse_part is used for this purpose and was employed to compute **J** for the $100 \times 100 \times 100$ arrays with model rank six. Even this procedure, though, is unfeasible when the problem is larger. For instance, at rank nine the Jacobian for the $100 \times 100 \times 100$ takes up 309MB in double precision even if stored as sparse and could not be calculated in a reasonable time.

The element-wise routines to compute **J^TJ** and **J^TWJ** (i.e. the functions JTJ_el and $JTWJ_el$, respectively) are built as to yield optimal performances in MATLAB (to the author's knowledge), minimising the number of operations and indexing. They do not employ external compiled code. JTJ_sparse and $JTWJ_sparse$, which explicitly form **J**, are

comprised of three steps. In the first, the indexes for nonzero elements of the Jacobian are computed; in the second, the Jacobian is formed and in the third, the actual product is calculated. Since the indexes need be reckoned only once for each fitting procedure, the corresponding time expense is not considered in the computational cost for these two routines apart for the case of $100 \times 100 \times 100$ arrays with model rank six, in which case J_sparse_part was used. Finally, the JTJ_full and $JTWJ_full$ use equations (19) and (25) to obtain $\mathbf{J}^T \mathbf{J}$ and equations (27), (28) and (29) to calculate $\mathbf{J}^T \mathbf{W} \mathbf{J}$. All the routines for computing $\mathbf{J}^T \mathbf{J}$ and $\mathbf{J}^T \mathbf{W} \mathbf{J}$ exploit the symmetry of these matrices (i.e. only the upper block-triangular part is calculated).

The computation of the nonlinear part \mathbf{Q} of the Hessian matrix comprises two steps: in the first, the indexes for the non-zero elements are reckoned (Q_index); in the second, these indexes are used to build \mathbf{Q} (Q_sparse). As for the Jacobian, the indexes would need to be computed only once and their reckoning is not included in the time expense for the computation of \mathbf{Q} . Because of the symmetry of \mathbf{Q} , only one half of the indexes need be stored and Q_sparse returns only one half of the elements in a sparse matrix $\mathbf{Q}_{\frac{1}{2}}$ that contains the upper triangular part of \mathbf{Q} . Hence, differently from \mathbf{J} , the storage for the indexes and the actual matrix \mathbf{Q} is typically rather limited and hardly a problem. For example, if the array has identical size I in all the modes and a rank F model is considered, $\mathbf{Q}_{\frac{1}{2}}$ has $N(N-1)I^2F$ nonzero elements. The full Hessian is formed as $\mathbf{H} = \mathbf{J}^T \mathbf{J} + \mathbf{Q}_{\frac{1}{2}} + \mathbf{Q}_{\frac{1}{2}}^T$ (or $\mathbf{H} = \mathbf{J}^T \mathbf{W} \mathbf{J} + \mathbf{Q}_{\frac{1}{2}} + \mathbf{Q}_{\frac{1}{2}}^T$ in the weighted least squares case).

Table 2. Computational steps for the considered optimisation algorithms. The functions JTv_chain , Jv_chain , JTJ_full and H_sparse have been used in the tests to compute $\mathbf{J}^T \mathbf{v}$, $\mathbf{J} \mathbf{v}$, $\mathbf{J}^T \mathbf{J}$ and $\mathbf{Q}_{\frac{1}{2}}$ respectively.

Algorithm	# L^a	Operation				Normal equations ^c
		$\mathbf{J}^T \mathbf{v}$	$\mathbf{J} \mathbf{v}$	$\mathbf{J}^T \mathbf{J}$	\mathbf{H}^b	
ALS ^d	2	✓				
Gauss-Newton	1	✓		✓		✓
Newton	1	✓		✓	✓	✓
CG soft line search	2	✓	✓			
CG exact line search	0	✓	✓			

^a Number of loss function evaluations per iteration; one loss function evaluation includes the computation of $\mathbf{R}_1 = \mathbf{X}_1 - \mathbf{A}_1 \mathbf{Z}_{-1}^T$ and of $\|\mathbf{R}_1\|_F^2$.

^b Includes the computation of $\mathbf{Q}_{\frac{1}{2}}$ and of $\mathbf{H} = \mathbf{J}^T \mathbf{J} + \mathbf{Q}_{\frac{1}{2}} + \mathbf{Q}_{\frac{1}{2}}^T$.

^c Normal equations entail the computation of $\mathbf{J}^T \mathbf{J} + \lambda \mathbf{D}$ (or $\mathbf{H} + \lambda \mathbf{D}$) and the solution of the linear system (11).

^d The product $\mathbf{J}^T \mathbf{v}$ with the routine JTv_chain is considered representative of the computational complexity of an ALS step apart from the loss function evaluation(s).

^e The time expense for \mathbf{R} is added twice to account for the soft line search (Harshman, 1970).

Table 3. Array and problem size to evaluate effect of order on different algorithms

Order	Size	N. of elements	N. of parameters			
			Rank	2	4	6
3	168	4741632		1008	2016	3024
4	47	4879681		376	752	1128
5	22	5153632		220	440	660
6	13	4826809		156	312	468
7	9	4782969		126	252	378

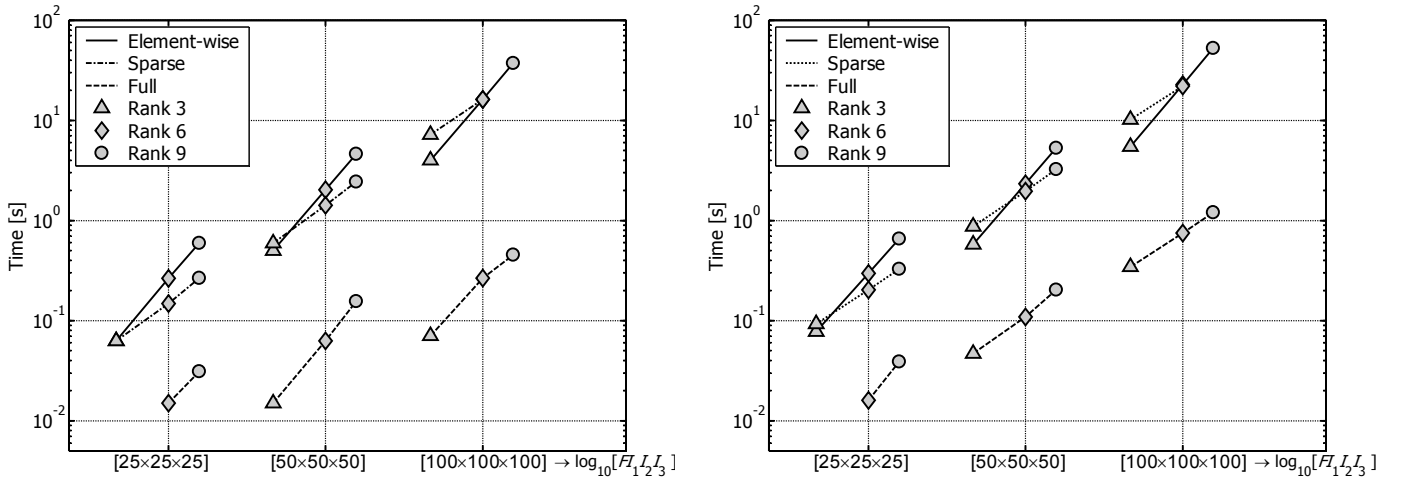


Figure 1. Median of the time consumption for the products $\mathbf{J}^T \mathbf{J}$ (left) and $\mathbf{J}^T \mathbf{W} \mathbf{J}$ (right). Note that for the $25 \times 25 \times 25$ / rank three case the ‘full’ routine is too fast and the corresponding time is zero. Per contra, the Jacobian for the $100 \times 100 \times 100$ case with rank nine was too large and the computation of $\mathbf{J}^T \mathbf{J}$ exceeded the available memory.

The conjugate gradients method, for which the products $\mathbf{J}^T \mathbf{v}$ and $\mathbf{J} \mathbf{v}$ are required, is suited for large problems, whereby the Hessian is too large to be stored. Therefore, the tests have been extended to cover cases with up approximately 2×10^4 estimated parameters. Two implementations are tested for each product: the ‘full’ routines (JTv_full and Jv_full) use equations (31) and (43) for $\mathbf{J}^T \mathbf{v}$ and $\mathbf{J} \mathbf{v}$ respectively, while the ‘chain-rule’ routines (JTv_chain and Jv_chain) use equations (35) and (45) for the same purpose.

The expense per iteration of the different fitting methods is evaluated in terms of computation time. The operations required by each algorithm (ALS, Newton, Gauss-Newton and CG) are listed in Table 2.

The cost of a single standard ALS step is well approximated by the cost of computing $\mathbf{J}^T \mathbf{v}$ plus that of one loss function evaluation (namely, the updated loading matrix at each sub-step of ALS is computed as $\mathbf{A}_n = \mathbf{X}_n \mathbf{Z}_{-n} \mathbf{E}_{-n}^+$, where \mathbf{E}_{-n}^+ denotes the pseudo inverse of \mathbf{E}_{-n} (Bro, 1998), and the cost of computing \mathbf{E}_{-n}^+ can often be neglected). However, it is possible to reduce the cost of an ALS iteration using a formula identical to the one based on the chain rule (cf. Appendix 3) and, in most ALS implementations, a simple line search procedure is added that accelerates convergence and requires an additional loss function evaluation (Harshman, 1970; Bro, 1998). Thus, the cost of an ALS iteration can be taken as equal to that of JTv_chain plus the cost of computing of the loss function twice.

Two variants of a nonlinear CG iteration are investigated: the first includes the cost of a simple line search (i.e., of an additional loss function evaluation, as implemented in the ME), whereas the second uses the exact line search procedure as described in Appendix 4 (*PARAFAC_exact_linesearch.m*) and does not require any loss function

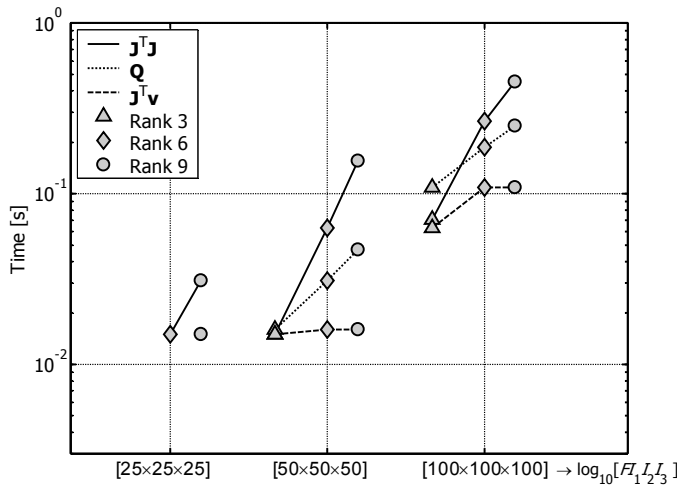


Figure 2. Median of time expense for the computation of the nonlinear part of the Hessian \mathbf{Q} . It can be observed that, as F increases, the contribution of the cross-product $\mathbf{J}^T \mathbf{J}$ to the time expense for the calculation of \mathbf{H} grows larger. The median time consumption for the $25 \times 25 \times 25$ arrays is for $\mathbf{J}^T \mathbf{v}$, for \mathbf{Q} at ranks 3 and 6 and for $\mathbf{J}^T \mathbf{J}$ at rank 3 are zero.

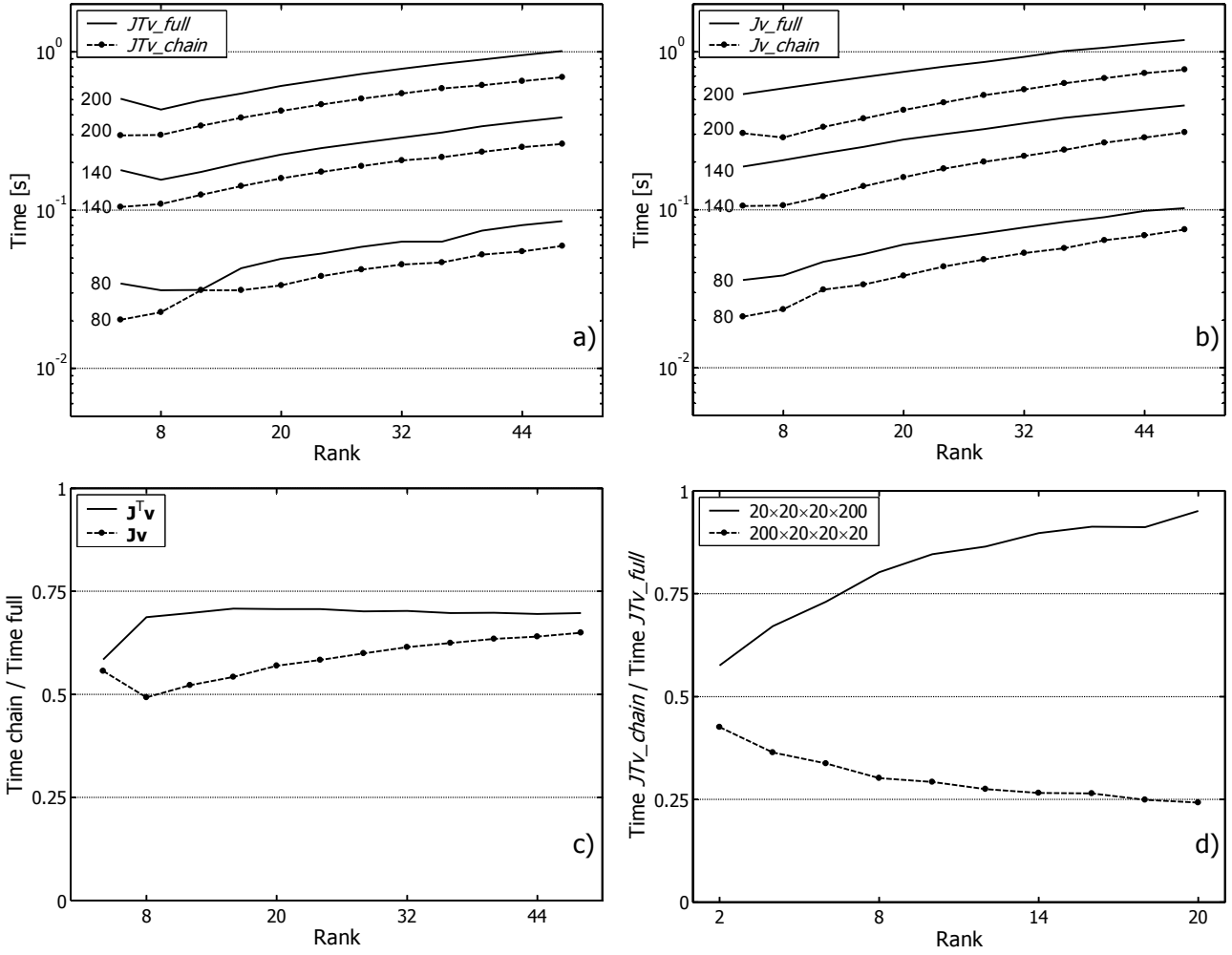


Figure 3. Computation time for the two products a) $J^T v$ and b) $J v$. The number on the right of the single curves indicates the size of the array. c) ratio between the computation time of the full routines and on chain rule based ones for a $200 \times 200 \times 200$ array. d) ratio between the computation times of full and chain rule based functions when the chain-rule reduction is applied (dashed line) or not (solid line) to the largest mode of a 4-way array.

evaluation (Paatero, 1999).

Finally, the number of modes may have a great influence on the time expense for the single algorithm since the number of array elements (which has the largest influence on the time expense for ALS and CG methods) grows with the product of the sizes of the different modes whereas the number of model parameters grows only with their sum. To test this aspect, arrays from order three to seven were formed as to have a number of array elements as close as possible to 9^7 (corresponding approximately to 36.5MB in double precision) and single iterations of the different algorithms were performed. The tested conditions are shown in Table 3.

3.1 Results and discussion

Figure 1. reports the computation time in seconds for the products $J^T J$ and $J^T W J$. It can be observed that JTJ_{full} and $JTWJ_{full}$ are at least one order of magnitude faster than both the sparse and the element-wise implementation. At the same time, JTJ_{sparse} and $JTWJ_{sparse}$ are faster than the element-wise routines for smaller arrays and for higher values of model's rank. On the other hand, the increase in speed of JTJ_{sparse} is not large enough to justify its preference over the JTJ_{el} , and from a practical perspective, its lower cost can hardly counterpoise the fact that for large arrays (or high ranks) J cannot be formed in the first place. It is worth remarking that this ratio between the performances may change when many values in the array are set to missing. Since missing values in a weighted least

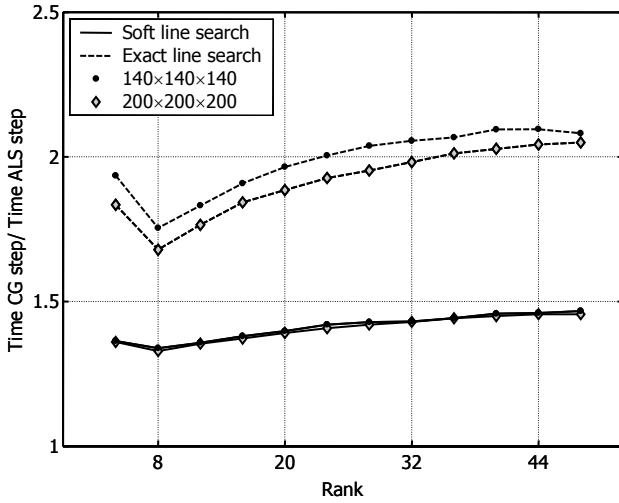


Figure 4. Ratio of cost per iteration between a CG step with soft and exact line search and an ALS step.

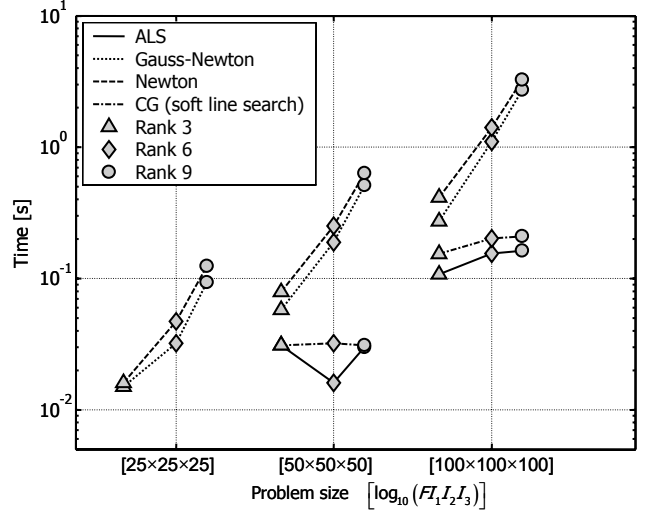


Figure 5. Computation cost of ALS, Gauss-Newton, Newton and Conjugate Gradients with soft line search. For the $25 \times 25 \times 25$ arrays ALS and CG iterations were too fast and the recorded time was zero.

squares approach are given zero weights, the corresponding rows of the Jacobian need not be filled and a sparse \mathbf{J} may be of manageable size and may turn out to be faster even than JTJ_{full} (Tomasi and Bro, 2005). A possibility that has not been investigated yet and will be the subject of future research is to store the \mathbf{W} as sparse and employ JTJ_{full} .

Figure 2 reports the computation time for \mathbf{Q} compared to that of JTJ_{full} and of JTv_{full} . It is apparent that the reckoning of the nonlinear part of the Hessian matrix is not particularly expensive. In fact, it is only slightly more expensive than a standard ALS step and as expensive as the computation of $\mathbf{J}^T \mathbf{J}$. Since the Cholesky decomposition of the full \mathbf{H} and of $\mathbf{J}^T \mathbf{J}$ are identical, the Newton method seems feasible whenever the Gauss-Newton algorithm is.

Figure 3a and b show the performances of the routines for computing $\mathbf{J}^T \mathbf{v}$ and \mathbf{Jv} . JTv_{chain} and Jv_{chain} require approximately 30% less time than JTv_{full} and Jv_{full} for all the tested conditions.

As mentioned in section 2.2, the computational advantage of JTv_{chain} and Jv_{chain} is expected to grow with the ratio between I_n and F . This is clearly visible for \mathbf{Jv} (Figure 3c), for which the gain decreases from approximately 50% at rank 8 to 30% at rank 48. For JTv_{chain} , the worsening is not visible thanks to the use of sparse identity matrices (and thus sparse Khatri-Rao products) for equation (33). The density of $\tilde{\mathbf{Z}}_{-n}$ decreases with the rank and compensates for the smaller ratio between size of the array and rank. When the array size in the different modes varies, it is important that the chain rule is employed to reduce the largest mode. This is clearly visible in Figure 3d: since the current implementation of JTv_{chain} and Jv_{chain} apply the reduction automatically to the first mode, regardless of its being the largest, when the largest mode is the first, the reduction in computational expense is dramatic and JTv_{chain} may require 25%-40% of the time of JTv_{full} . The ratio between Jv_{full} and Jv_{chain} is also around 0.25, although it does not decrease with the rank as for $\mathbf{J}^T \mathbf{v}$. This may be even more interesting when fitting a PARAFAC model to spectral data, whereby the size in one of the modes may be in the order of hundreds and the rank is limited to a few units.

A carefully implemented CG iteration includes one $\mathbf{J}^T \mathbf{v}$ and one \mathbf{Jv} product (Björck, 1996) and can greatly benefit from the use of formulae that do not require the explicit computation of the Jacobian. Paatero (Paatero, 1999) suggests that using sub-expressions (i.e. the chain rule) and sparse matrices yield CG iterations that are approximately three times as slow as an ALS step in the least squares case. However, the fact that the same formulae based on the chain rule can be used in ALS to reduce the cost of a single iteration and the additional loss function evaluation used for the line search can affect this ratio. Figure 4 shows the ratio between the time expense of an ALS iteration based on (A3.6) and of a CG step with either soft (Paatero, 1999) or exact line search (cf. Appendix 4). As can be seen, a CG iteration with exact line search is approximately twice as expensive as a standard ALS iteration, whereas a CG with soft line search corresponds to only 1.3 to 1.5 ALS iterations. Moreover, for three-way arrays, it may not be worthwhile to implement line search procedures that exceed few loss function evaluations because the soft line search becomes more expensive than the exact one when the loss function is evaluated more than three times (not shown). Figure 4 also shows that exact line search becomes more efficient as the I_n/F ratio increases, which also explains why the time consumption is lower for a larger array.

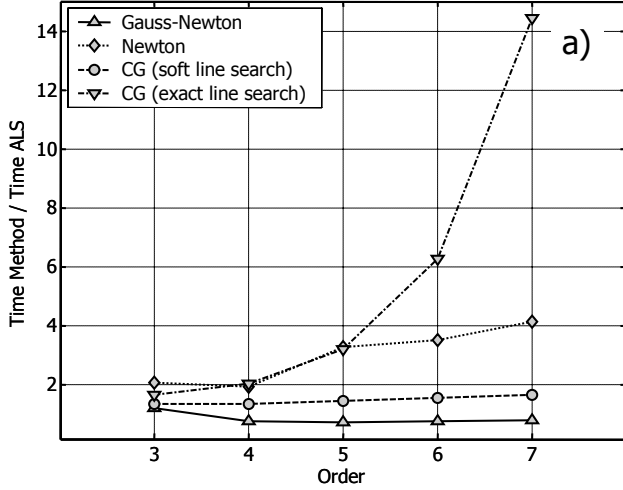


Figure 6. Median of the time consumption (per iteration) for four different algorithms made 1 the time consumption for ALS. (a) Model rank 2, (b) rank 4 and (c) rank 6.

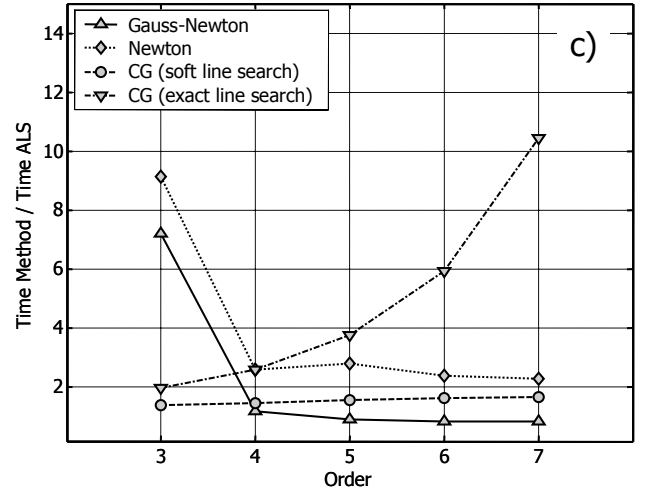
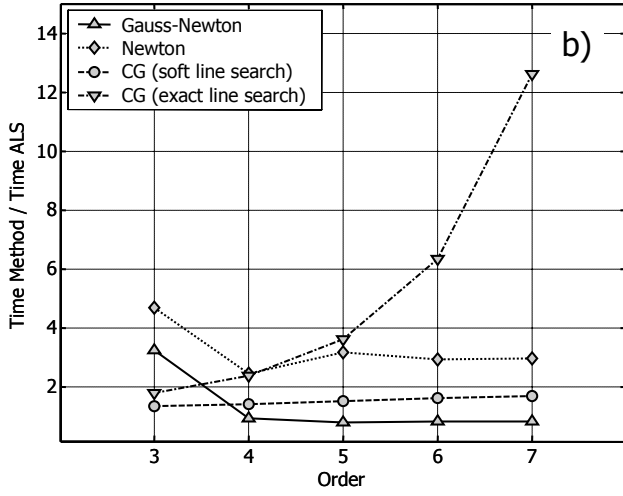


Figure 5 shows the iteration cost for the four considered algorithms on three way arrays. It is apparent that the cost per iteration for the Gauss-Newton and Newton approaches grows more rapidly with respect to F than for ALS and CG. This is hardly unexpected, because the number of operations for the Cholesky decomposition grows with F^3 , whereas for ALS (and CG) the cost per iteration grows linearly with F . Moreover, Newton steps are only slightly more costly than Gauss-Newton ones for three-way arrays, which suggests that an hybrid Gauss-Newton/Newton algorithm could be a good option for three-way problems with large residuals (Madsen et al., 2004). On the other hand, the speed of the linear CG step (i.e., without a line search step) indicates that, if the number of CG iterations is kept within few units truncated Newton methods may be the optimal choice for large problems.

Finally, Figure 6 shows the behaviour of the different algorithms as the order of the array increases. For higher order arrays (i.e., of order at least four, although this may increase for larger values of F) Gauss-Newton iterations are approximately as expensive as ALS ones. If one takes into account that the former method requires only a fraction of the iterations of the latter (particularly for highly collinear underlying components (Tomasi and Bro, 2006; Hopke et al., 1998)), the advantage of Gauss-Newton can be of more than one order of magnitude in terms of time consumption. From Figures 6a and b one can also infer that Gauss-Newton should be the algorithm of choice also for large arrays when the rank of the fitted model is small and $\mathbf{J}^T \mathbf{J}$ can be stored. Finally, the performance of the CG algorithm with exact line search deteriorates significantly as the order increases because of the computation of the coefficients of the polynomial used to solve the exact line search problem. In particular, in the procedure illustrated in Appendix 4, the number of Khatri-Rao products of $N-1$ terms necessary for the polynomial's coefficients grows exponentially with the order of the array. This nonlinear behaviour is clearly visible in the computation time and suggests that the proposed exact line search procedure is not feasible for higher order arrays. To verify the cost of the presented exact line search procedure compared to other methods, an algorithm that finds the coefficients by fitting a polynomial of order $2N$ to a sufficient number of data points (i.e. requiring $2N+1$ function evaluations) was developed. The latter method required less time than the proposed procedure for arrays of order higher than the fifth (results not shown).

4. Conclusions

The problem of fitting a PARAFAC model would greatly benefit from more efficient algorithms. In this paper, general formulae have been developed for the computation of several matrices and vectors that are required by most nonlinear curve fitting algorithms. Moreover, the results obtained with regards to the computation of such matrices and vectors have been systematised and expressed in terms of the Khatri-Rao product. The properties of the latter operation allow for many algebraic simplifications in the mathematical expressions and for a straightforward generalisation of existing algorithms and formulae.

The tests conducted in the course of this work also indicate that there is no particular reason for preferring ALS to, e.g., a Gauss-Newton (or Newton) algorithm. In fact, these methods appear to be complementary and one should be preferred to the other depending on the problem at hand. Hence, the Gauss-Newton (or Newton) method should be used when the number of estimated parameters is small and the number of elements in the array is large, as would happen for example for arrays with order higher than three and if the model rank is not exceedingly high. Likewise, the single iteration of a Conjugate Gradient algorithm appears to be only slightly more expensive than an ALS one. Thus, under the condition that an effective line search procedure is employed and a good preconditioning is found, nonlinear PCG could prove to be an effective method to fit PARAFAC models.

Whether the use of exact line search also leads to a faster convergence of the CG algorithm and the possible reduction in the number of iterations is sufficient to counterbalance the additional cost per iteration is to be further investigated. A similar observation can be made with respect to implementing exact line search in ALS in substitution of the simple scheme based on the iteration number currently used in, e.g., the N -way toolbox (Andersson and Bro, 2000). Some preliminary tests on difficult problems (Hopke et al., 1998) suggest that the reduction in the number of iterations using exact line search in ALS is not enough to compensate for the higher cost per iteration (results not shown). Again, these results need to be further validated before any conclusion can be drawn on the matter. In any case, exact line search seems unfeasible for higher order arrays, although it may be useful to exploit the number of common sub-terms in the 2^{N-1} Khatri-Rao products of $N-1$ terms necessary for to compute the coefficient of the polynomial used to solve the exact line search problem.

Reference List

MATLAB 6.5, 2002 The Mathworks.

Andersson CA, Bro R (2000) The N-way Toolbox for MATLAB. *Chemometrics Intell. Lab. Syst.* 52:1-4

Andersson GG, Dable BK, Booksh KS (1999) Weighted parallel factor analysis for calibration of HPLC-UV/Vis spectrometers in the presence of Beer's law deviations. *Chemometr. Intell. Lab. Syst.* 49:195-213

Björck Å (1996) *Numerical Methods for Least Squares Problems*, First edn. Society for Industrial and Applied Mathematics, Philadelphia (PA), USA

Bro, R. (1998) *Multi-way Analysis in the Food Industry. Models, Algorithms, and Applications*, Ph.D. dissertation, University of Amsterdam, Amsterdam, Netherlands.

Bro R, Sidiropoulos ND, Smilde AK (2002) Maximum likelihood fitting using ordinary least squares algorithms. *J. of Chemometrics* 16:387-400

Carroll JD, Chang JJ (1970) Analysis of Individual Differences in Multidimensional Scaling Via An N-Way Generalization of Eckart-Young Decomposition. *Psychometrika* 35:283-319

Faber NM, Bro R, Hopke PK (2003) Recent developments in CANDECOMP/PARAFAC algorithms: a critical review. *Chemometr. Intell. Lab. Syst.* 65:119-137

- Franc, A. (1992) *Etude algebrique des multitableaux: apports de l'algebre tensorielle*, Ph.D. dissertation, Université Montpellier II, Montpellier, France.
- Frandsen PE, Jonasson K, Nielsen HB, Tingleff O (2004) *Unconstrained Optimization*, 3rd edn. Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark
- Gill PE, Murray W, Wright MH (1986) *Practical Optimization*. Academic Press, London, UK
- Golub GH, Van Loan CF (1996) *Matrix Computations*, 3 edn. The Johns Hopkins University Press, Baltimore (MD), USA
- Harshman RA (1970) Foundations of the PARAFAC procedure: Models and conditions for an 'explanatory' multi-modal factor analysis. *UCLA working papers in phonetics* 16:1-84
- Hopke PK, Paatero P, Jia H, Ross RT, Harshman RA (1998) Three-way (PARAFAC) factor analysis: examination and comparison of alternative computational methods as applied to ill-conditioned data. *Chemometr. Intell. Lab. Syst.* 43:25-42
- Kapteyn A, Neudecker H, Wansbeek T (1986) An approach to n-mode components analysis. *Psychometrika* 51:269-275
- Kiers HAL (1998) A three-step algorithm for CANDECOMP/PARAFAC analysis of large data sets with multicollinearity. *J. of Chemometrics* 12:155-171
- Kiers HAL (2000) Towards a standardized notation and terminology in multiway analysis. *J. of Chemometrics* 14:105-122
- Liu XQ, Sidiropoulos ND (2001) Cramer-Rao lower bounds for low-rank decomposition of multidimensional arrays. *Ieee Transactions on Signal Processing* 49:2074-2086
- Madsen K, Nielsen HB, Tingleff O (2004) *Methods for non-linear least squares problems*, 2nd edn. Dept. Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark
- Magnus JR, Neudecker H (1999) *Matrix differential calculus with applications in statistics and econometrics Rev. ed.* John Wiley & Sons Ltd, Chichester, UK
- Mitchell BC, Burdick DS (1994) Slowly Converging Parafac Sequences - Swamps and 2-Factor Degeneracies. *J. of Chemometrics* 8:155-168
- Paatero P (1999) The multilinear engine - A table-driven, least squares program for solving multilinear problems, including the n-way parallel factor analysis model. *Journal of Computational and Graphical Statistics* 8:854-888
- Paatero P (1997) A weighted non-negative least squares algorithm for three-way 'PARAFAC' factor analysis. *Chemometr. Intell. Lab. Syst.* 38:223-242
- Rao CR, Mitra S (1971) *Generalized inverse of matrices and its applications*. John Wiley & Sons Ltd, Chichester, UK
- Schott JR (1997) *Matrix Analysis for Statistics*. John Wiley & Sons Ltd, Chichester, UK
- Smilde AK, Bro R, Geladi P (2004) *Multi-way analysis Applications in the chemical sciences*, 1st edn. John Wiley & Sons Ltd, Chichester, UK
- Tomasi G, Bro R (2005) PARAFAC and missing values. *Chemometrics Intell. Lab. Syst.* 75:163-180
- Tomasi G, Bro R (2006) A comparison of algorithms for fitting the PARAFAC model. *Comput. Stat. Data Anal.* 50:1700-1734

Vega-Montoto L, Wentzell PD (2003) Maximum likelihood parallel factor analysis (MLPARAFAC). J. of Chemometrics 17:237-253

Wang JH, Hopke PK (2001) Equation-oriented system: an efficient programming approach to solve multilinear and polynomial equations by the conjugate gradient algorithm. Chemometr. Intell. Lab. Syst. 55:13-22

Appendix 1. Some relevant formulae for the Khatri-Rao product

Let \mathbf{A} and \mathbf{B} be two matrices of size $I \times F$ and $J \times F$ respectively, a well known property of the column-wise Khatri-Rao product is (Bro, 1998):

$$(\mathbf{A} \odot \mathbf{B})^T (\mathbf{A} \odot \mathbf{B}) = \mathbf{A}^T \mathbf{A} * \mathbf{B}^T \mathbf{B}. \quad (\text{A.1})$$

Owing to the associativity of the Khatri-Rao product, equation (A.1) readily extends to products of an arbitrary number of matrices (Carroll and Chang, 1970; Bro, 1998). Consequently, for a set of N matrices \mathbf{A}_n for $n = 1 \dots N$:

$$\begin{aligned} \mathbf{E} &= (\mathbf{A}_1 \odot \dots \odot \mathbf{A}_N)^T (\mathbf{A}_1 \odot \dots \odot \mathbf{A}_N) = \\ &= [\mathbf{A}_1 \odot (\mathbf{A}_2 \odot \dots \odot \mathbf{A}_N)]^T [\mathbf{A}_1 \odot (\mathbf{A}_2 \odot \dots \odot \mathbf{A}_N)] = \\ &= \mathbf{A}_1^T \mathbf{A}_1 * (\mathbf{A}_2 \odot \dots \odot \mathbf{A}_N)^T (\mathbf{A}_2 \odot \dots \odot \mathbf{A}_N) = \mathbf{A}_1^T \mathbf{A}_1 * \dots * \mathbf{A}_N^T \mathbf{A}_N \end{aligned} \quad (\text{A.2})$$

It is also straightforward that for two matrices \mathbf{C} and \mathbf{D} of the same size of \mathbf{A} and \mathbf{B} respectively

$$(\mathbf{A} \odot \mathbf{B})^T (\mathbf{C} \odot \mathbf{D}) = \mathbf{A}^T \mathbf{C} * \mathbf{B}^T \mathbf{D}.$$

The following results pertaining to the Khatri-Rao and Kronecker products will also be useful.

Lemma 1. Let $\mathbf{A} \in \mathbb{R}^{I \times F}$, $\mathbf{A}_2 \in \mathbb{R}^{I \times F}$, $\mathbf{B}_1 \in \mathbb{R}^{J \times F}$ and $\mathbf{B}_2 \in \mathbb{R}^{J \times F}$, then the following hold

$$\mathbf{H} = (\mathbf{A}_1 \otimes \mathbf{B}_1) * (\mathbf{A}_2 \otimes \mathbf{B}_2) = (\mathbf{A}_1 * \mathbf{A}_2) \otimes (\mathbf{B}_1 * \mathbf{B}_2) \quad (\text{A.3})$$

$$\mathbf{H} = (\mathbf{A}_1 \odot \mathbf{B}_1) * (\mathbf{A}_2 \odot \mathbf{B}_2) = (\mathbf{A}_1 * \mathbf{A}_2) \odot (\mathbf{B}_1 * \mathbf{B}_2) \quad (\text{A.4})$$

Proof. Defining four diagonal matrices $\mathbf{D}_{\text{vec } \mathbf{A}_1}$, $\mathbf{D}_{\text{vec } \mathbf{B}_1}$, $\mathbf{D}_{\text{vec } \mathbf{A}_2}$ and $\mathbf{D}_{\text{vec } \mathbf{B}_2}$ holding the elements of the vectors $\text{vec } \mathbf{A}_1$, $\text{vec } \mathbf{B}_1$, $\text{vec } \mathbf{A}_2$ and $\text{vec } \mathbf{B}_2$ respectively, and using the relation between the vec operator and the Hadamard product: $\text{vec}(\mathbf{A} * \mathbf{B}) = \mathbf{D}_{\text{vec } \mathbf{A}} (\text{vec } \mathbf{B}) = (\text{vec } \mathbf{A}) \mathbf{D}_{\text{vec } \mathbf{B}}$, one obtains that

$$\begin{aligned} \text{vec } \mathbf{H} &= \text{vec}[(\mathbf{A}_1 \otimes \mathbf{B}_1) * (\mathbf{A}_2 \otimes \mathbf{B}_2)] = (\mathbf{D}_{\text{vec } \mathbf{A}_1} \otimes \mathbf{D}_{\text{vec } \mathbf{B}_1}) (\text{vec } \mathbf{A}_2 \otimes \text{vec } \mathbf{B}_2) = \\ &= [(\mathbf{D}_{\text{vec } \mathbf{A}_1}) \text{vec } \mathbf{A}_2] \otimes [(\mathbf{D}_{\text{vec } \mathbf{B}_1}) \text{vec } \mathbf{B}_2] = [\text{vec}(\mathbf{A}_1 * \mathbf{A}_2)] \otimes [\text{vec}(\mathbf{B}_1 * \mathbf{B}_2)] = \\ &= \text{vec}[(\mathbf{A}_1 * \mathbf{A}_2) \otimes (\mathbf{B}_1 * \mathbf{B}_2)] \end{aligned}$$

(A.4) also holds because the Khatri-Rao product is a selection of the columns of the Kronecker product. ■

Note also that, because of the associativity of the Kronecker and Khatri-Rao products, *Lemma 1* holds for the products of any number of suitably sized matrices. Thus, for instance, given \mathbf{C}_1 and \mathbf{C}_2 of size $K \times F$:

$$\begin{aligned} (\mathbf{A}_1 \odot \mathbf{B}_1 \odot \mathbf{C}_1) * (\mathbf{A}_2 \odot \mathbf{B}_2 \odot \mathbf{C}_2) &= [(\mathbf{A}_1 \odot \mathbf{B}_1) \odot \mathbf{C}_1] * [(\mathbf{A}_2 \odot \mathbf{B}_2) \odot \mathbf{C}_2] = \\ &= [(\mathbf{A}_1 \odot \mathbf{B}_1) * (\mathbf{A}_2 \odot \mathbf{B}_2)] \odot (\mathbf{C}_1 * \mathbf{C}_2) = (\mathbf{A}_1 * \mathbf{A}_2) \odot (\mathbf{B}_1 * \mathbf{B}_2) \odot (\mathbf{C}_1 * \mathbf{C}_2) \end{aligned}$$

Two special cases are used throughout this work: for any two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^I$

$$\mathbf{a}^T \odot \mathbf{b}^T = \mathbf{a}^T * \mathbf{b}^T; \quad (\text{A.5})$$

and, for any two vectors $\mathbf{a} \in \mathbb{R}^I$ and $\mathbf{b} \in \mathbb{R}^J$, it holds

$$\mathbf{a}\mathbf{b}^T = \mathbf{a} \otimes \mathbf{b}^T = (\mathbf{a} * \mathbf{1}_I) \otimes (\mathbf{1}_J^T * \mathbf{b}^T) = (\mathbf{a} \otimes \mathbf{1}_J^T) * (\mathbf{1}_I \otimes \mathbf{b}^T) = \mathbf{a} \mathbf{1}_J^T * \mathbf{1}_I \mathbf{b}^T \quad (\text{A.6})$$

Lemma 2. Let $\mathbf{A}_n \in \mathbb{R}^{I_n \times F}$ for $n=1, \dots, N$ be a set of N matrices and define the matrices $\mathbf{N}_1 = \mathbf{e}_{i_n} \mathbf{1}_F^T$, $\mathbf{N}'_1 = \mathbf{e}_{i'_n} \mathbf{1}_F^T$ and $\mathbf{N}_2 = \mathbf{e}_{i'_n} \mathbf{1}_F^T$, where \mathbf{e}_{i_n} , $\mathbf{e}_{i'_n}$ are respectively the i_n -th and i'_n -th columns of an $I_n \times I_n$ identity matrix and $\mathbf{e}_{i'_n}$ is the i'_n -th column of an $I_{n'} \times I_{n'}$ identity matrix; then

if $n \neq n'$:

$$\begin{aligned} \mathbf{H} &= (\mathbf{A}_1 \odot \dots \odot \mathbf{A}_{n-1} \odot \mathbf{N}_1 \odot \mathbf{A}_{n+1} \dots \odot \mathbf{A}_N)^T (\mathbf{A}_1 \odot \dots \odot \mathbf{A}_{n'-1} \odot \mathbf{N}_2 \odot \mathbf{A}_{n'+1} \dots \odot \mathbf{A}_N) = \\ &= \mathbf{A}_1^T \mathbf{A}_1 * \dots * \mathbf{A}_{n-1}^T \mathbf{A}_{n-1} * \mathbf{A}_{n+1}^T \mathbf{A}_{n+1} * \dots * \mathbf{A}_{n'-1}^T \mathbf{A}_{n'-1} * \mathbf{A}_{n'+1}^T \mathbf{A}_{n'+1} * \dots * \mathbf{A}_N^T \mathbf{A}_N * \bar{\mathbf{a}}_{n', i'_n}^T \bar{\mathbf{a}}_{n, i_n} =, \\ &= \mathbf{E}_{-nn'} * \bar{\mathbf{a}}_{n', i'_n}^T \bar{\mathbf{a}}_{n, i_n} \end{aligned} \quad (\text{A.7})$$

if $n = n'$:

$$\begin{aligned} \mathbf{H} &= (\mathbf{A}_1 \odot \dots \odot \mathbf{A}_{n-1} \odot \mathbf{N}_1 \odot \mathbf{A}_{n+1} \dots \odot \mathbf{A}_N)^T (\mathbf{A}_1 \odot \dots \odot \mathbf{A}_{n-1} \odot \mathbf{N}'_1 \odot \mathbf{A}_{n+1} \dots \odot \mathbf{A}_N) = \\ &= \delta_{i_n i'_n} \mathbf{A}_1^T \mathbf{A}_1 * \dots * \mathbf{A}_{n-1}^T \mathbf{A}_{n-1} * \mathbf{A}_{n+1}^T \mathbf{A}_{n+1} * \dots * \mathbf{A}_N^T \mathbf{A}_N = \delta_{i_n i'_n} \mathbf{E}_{-n} \end{aligned} \quad (\text{A.8})$$

Proof. Owing to the associativity of Khatri-Rao product, case a) reduces to

$$\begin{aligned} \mathbf{H} &= \mathbf{A}_1^T \mathbf{A}_1 * \dots * \mathbf{A}_{n-1}^T \mathbf{A}_{n-1} * \mathbf{N}_1^T \mathbf{A}_n * \mathbf{A}_{n+1}^T \mathbf{A}_{n+1} * \dots * \mathbf{A}_{n'-1}^T \mathbf{A}_{n'-1} * \mathbf{A}_{n'+1}^T \mathbf{N}_2 * \mathbf{A}_{n'+1}^T \mathbf{A}_{n'+1} * \dots * \mathbf{A}_N^T \mathbf{A}_N = \\ &= \mathbf{A}_1^T \mathbf{A}_1 * \dots * \mathbf{1}_F \bar{\mathbf{a}}_{i_n} * \dots * \bar{\mathbf{a}}_{i'_n}^T \mathbf{1}_F^T * \dots * \mathbf{A}_N^T \mathbf{A}_N \end{aligned}$$

where the second equality holds because $\mathbf{N}_1^T \mathbf{A}_n = \mathbf{1}_F \mathbf{e}_{i_n}^T \mathbf{A}_n = \mathbf{1}_F \bar{\mathbf{a}}_{i_n}$ and $\mathbf{A}_{n'}^T \mathbf{N}_2 = \mathbf{A}_{n'}^T \mathbf{e}_{i'_n} \mathbf{1}_F^T = \bar{\mathbf{a}}_{i'_n}^T \mathbf{1}_F^T$.

Equation (A.7) follows immediately since the Hadamard product is commutative and using (A.6).

b) can be demonstrated in the same way using the relation $\mathbf{N}_1^T \mathbf{N}'_1 = \mathbf{1}_F \mathbf{e}_{i_n}^T \mathbf{e}_{i'_n} \mathbf{1}_F^T = \delta_{i_n i'_n} \mathbf{1}_F \mathbf{1}_F^T$ and the fact that $\mathbf{1}_F \mathbf{1}_F^T * \mathbf{M} = \mathbf{M}$ (Magnus and Neudecker, 1999). ■

Lemma 3. Given a set of N matrices $\{\mathbf{A}_n | \mathbf{A}_n \in \mathbb{R}^{I_n \times F}, n=1, \dots, N\}$, for any $n, n'=1, 2, \dots, N$ and $n \neq n'$:

$$\mathbf{Z} = \mathbf{M}_n (\mathbf{A}_n \odot \mathbf{Z}_{-n}) \quad (\text{A.9})$$

$$\mathbf{Z} = \mathbf{M}_{nn'} (\mathbf{A}_{n'} \odot \mathbf{A}_n \odot \mathbf{Z}_{-nn'}), \quad (\text{A.10})$$

where \mathbf{M}_n and $\mathbf{M}_{nn'}$ are suitably defined commutation matrices.

Proof. Only a) will be proven here, as the proof for b) is very similar.

For $n=1, 2, \dots, N$, let $\mathbf{X}_n = \mathbf{b}_n (\mathbf{b}_N \odot \dots \odot \mathbf{b}_{n+1} \odot \mathbf{b}_{n-1} \dots \odot \mathbf{b}_1)^T$ where \mathbf{b}_n denotes a vector of length I_n , and define the commutation matrix \mathbf{M}_n such that $\text{vec } \mathbf{X}_1 = \mathbf{M}_n \text{vec } \mathbf{X}_n^T$. Owing to the relation: $\text{vec } \mathbf{a}\mathbf{b}^T = \mathbf{b} \otimes \mathbf{a} = \mathbf{b} \odot \mathbf{a}$ (Magnus and Neudecker, 1999), it holds that

$$\begin{aligned} \mathbf{b}_N \odot \dots \odot \mathbf{b}_1 &= \mathbf{b}_1 (\mathbf{b}_N \odot \dots \odot \mathbf{b}_2)^T = \text{vec } \mathbf{X}_1 = \mathbf{M}_n \text{vec } \mathbf{X}_n^T = \mathbf{M}_n \text{vec } (\mathbf{b}_N \odot \dots \odot \mathbf{b}_{n+1} \odot \mathbf{b}_{n-1} \dots \odot \mathbf{b}_1) \mathbf{b}_n^T = \\ &= \mathbf{M}_n [\mathbf{b}_n \odot \mathbf{b}_N \odot \dots \odot \mathbf{b}_{n+1} \odot \mathbf{b}_{n-1} \dots \odot \mathbf{b}_1]. \end{aligned}$$

Thus, partitioning \mathbf{Z} and \mathbf{Z}_{-n} as

$$\begin{aligned} \mathbf{Z} &= [\mathbf{z}_1 \quad \dots \quad \mathbf{z}_F] = [\mathbf{a}_{N,1} \odot \dots \odot \mathbf{a}_{1,1} \quad \dots \quad \mathbf{a}_{N,1} \odot \dots \odot \mathbf{a}_{1,1}] \\ \mathbf{Z}_{-n} &= [\mathbf{z}_{-n,1} \quad \dots \quad \mathbf{z}_{-n,F}] = [\mathbf{a}_{N,1} \odot \dots \odot \mathbf{a}_{n+1,1} \odot \mathbf{a}_{n-1,1} \dots \odot \mathbf{a}_{1,1} \quad \dots \quad \mathbf{a}_{N,F} \odot \dots \odot \mathbf{a}_{n+1,F} \odot \mathbf{a}_{n-1,F} \dots \odot \mathbf{a}_{1,F}], \end{aligned}$$

one yields

$$\begin{aligned}\mathbf{Z} &= \left[\mathbf{a}_{N,1} \odot \dots \odot \mathbf{a}_{1,1} \mid \dots \mid \mathbf{a}_{N,F} \odot \dots \odot \mathbf{a}_{1,F} \right] = \left[\mathbf{M}_n(\mathbf{a}_{n,1} \odot \mathbf{z}_{-n,1}) \mid \dots \mid \mathbf{M}_n(\mathbf{a}_{n,F} \odot \mathbf{z}_{-n,F}) \right] = \\ &= \mathbf{M}_n \left[\mathbf{a}_{n,1} \odot \mathbf{z}_{-n,1} \mid \mathbf{a}_{n,2} \odot \mathbf{z}_{-n,2} \mid \dots \mid \mathbf{a}_{n,F} \odot \mathbf{z}_{-n,F} \right] = \mathbf{M}_n(\mathbf{A}_n \odot \mathbf{Z}_{-n})\end{aligned}$$

■

Lemma 4. Let $\underline{\mathbf{X}}$ be an N -way array of dimensions $I_1 \times \dots \times I_N$ and $\mathbf{x} \equiv \text{vec } \mathbf{X}_1$, given a set of matrices $\{\mathbf{A}_n \mid \mathbf{A}_n \in \mathbb{R}^{I_n \times F}, n=1,2,\dots,N\}$ and two matrices $\mathbf{N}_i = \mathbf{e}_i \mathbf{1}_F^T$ and $\mathbf{N}_j = \mathbf{e}_j \mathbf{1}_F^T$ with $n \neq n'$, where \mathbf{e}_i denotes the i -th column of the $I_n \times I_n$ identity matrix and \mathbf{e}_j is the j -th column of the $I_{n'} \times I_{n'}$ identity matrix. Then:

$$(\mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{N}_i \odot \mathbf{A}_{n-1} \dots \odot \mathbf{A}_{n'+1} \odot \mathbf{N}_j \odot \mathbf{A}_{n'-1} \dots \odot \mathbf{A}_1)^T \mathbf{x} = \mathbf{Z}_{-nn'}^T \mathbf{X}_{nn'}^T (\mathbf{e}_i \odot \mathbf{e}_j). \quad (\text{A.11})$$

Proof. Using Lemma 3:

$$\begin{aligned}\mathbf{u} &= (\mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{N}_i \odot \mathbf{A}_{n-1} \dots \odot \mathbf{A}_{n'+1} \odot \mathbf{N}_j \odot \mathbf{A}_{n'-1} \dots \odot \mathbf{A}_1)^T \mathbf{x} = \\ &= \left[\mathbf{M}_{nn'}(\mathbf{N}_j \odot \mathbf{N}_i \odot \mathbf{Z}_{-nn'}) \right]^T \mathbf{x} = (\mathbf{N}_j \odot \mathbf{N}_i \odot \mathbf{Z}_{-nn'})^T \mathbf{M}_{nn'}^T \text{vec } \mathbf{X}_1 = (\mathbf{N}_j \odot \mathbf{N}_i \odot \mathbf{Z}_{-nn'})^T \text{vec } \mathbf{X}_{nn'}^T.\end{aligned}$$

Since $(\mathbf{N}_i \odot \mathbf{B}) = [\mathbf{e}_i \odot \mathbf{b}_1 \ \dots \ \mathbf{e}_i \odot \mathbf{b}_F] = \mathbf{e}_i \otimes \mathbf{B}$ for any matrix \mathbf{B} with F columns and using the relation $\text{vec } \mathbf{ABC} = (\mathbf{C}^T \otimes \mathbf{A}) \text{vec } \mathbf{B}$ (Magnus and Neudecker, 1999), it follows that:

$$\mathbf{u} = \left[(\mathbf{e}_j \odot \mathbf{e}_i) \otimes \mathbf{Z}_{-nn'} \right]^T \text{vec } \mathbf{X}_{nn'}^T = \mathbf{Z}_{-nn'}^T \mathbf{X}_{nn'}^T (\mathbf{e}_j \odot \mathbf{e}_i)$$

■

Lemma 5. Let $\mathbf{A} \in \mathbb{R}^{I \times F}$ and $\mathbf{D}_{\mathbf{w}}$ an $IJ \times IJ$ diagonal matrix holding the elements of the vector $\mathbf{w} = \text{vec } \mathbf{W}$ where $\mathbf{W} \in \mathbb{R}^{I \times J}$. Define also two matrices $\mathbf{N}_j = \mathbf{e}_j \mathbf{1}_F^T$ and $\mathbf{N}_{j'} = \mathbf{e}_{j'} \mathbf{1}_F^T$, where \mathbf{e}_j and $\mathbf{e}_{j'}$ are the j -th and j' -th columns of the $J \times J$ identity matrix. Then

$$(\mathbf{N}_j \odot \mathbf{A})^T \mathbf{D}_{\mathbf{w}} (\mathbf{N}_{j'} \odot \mathbf{A}) = \delta_{jj'} \mathbf{A}^T \mathbf{D}_{\mathbf{w}_j} \mathbf{A} \quad (\text{A.12})$$

where $\mathbf{D}_{\mathbf{w}_j}$ is a diagonal matrix holding the elements of the j -th column of \mathbf{W} .

Proof. First, note that the left hand side of (A.12) is the equation of a rank IJ PARAFAC model of a three-way array $\underline{\mathbf{H}} \in \mathbb{R}^{F \times F \times 1}$ and having loading matrices $(\mathbf{N}_j \odot \mathbf{A})^T$, $(\mathbf{N}_{j'} \odot \mathbf{A})^T$ and \mathbf{w}^T . Thus, one can write:

$$\mathbf{H}_1 = (\mathbf{N}_j \odot \mathbf{A})^T \left[\mathbf{w}^T \odot (\mathbf{N}_{j'} \odot \mathbf{A})^T \right]^T,$$

or, equivalently:

$$\text{vec } \mathbf{H}_1 = \text{vec } \mathbf{H}_3^T = \left[(\mathbf{N}_j \odot \mathbf{A})^T \odot (\mathbf{N}_{j'} \odot \mathbf{A})^T \right] \mathbf{w},$$

Element $h_{jj'}$ of $\mathbf{H} \equiv \mathbf{H}_1$ is equal to

$$h_{jj'} = \mathbf{e}_{j'}^T \mathbf{H} \mathbf{e}_{j'} = \text{vec}(\mathbf{e}_{j'}^T \mathbf{H} \mathbf{e}_{j'}) = (\mathbf{e}_{j'}^T \otimes \mathbf{e}_{j'}^T) \mathbf{H} = (\mathbf{e}_{j'}^T \otimes \mathbf{e}_{j'}^T) \left[(\mathbf{N}_j \odot \mathbf{A})^T \odot (\mathbf{N}_{j'} \odot \mathbf{A})^T \right] \mathbf{w}.$$

Considering that $(\mathbf{S} \otimes \mathbf{T})(\mathbf{U} \odot \mathbf{V}) = \mathbf{SU} \odot \mathbf{TV}$ for \mathbf{S} , \mathbf{T} , \mathbf{U} and \mathbf{V} of appropriate size (Smilde et al., 2004), one obtains

$$\begin{aligned}h_{jj'} &= (\mathbf{e}_{j'}^T \otimes \mathbf{e}_{j'}^T) \left[(\mathbf{N}_j \odot \mathbf{A})^T \odot (\mathbf{N}_{j'} \odot \mathbf{A})^T \right] \mathbf{w} = \left\{ \left[\mathbf{e}_{j'}^T (\mathbf{N}_j \odot \mathbf{A})^T \right] \odot \left[\mathbf{e}_{j'}^T (\mathbf{N}_{j'} \odot \mathbf{A})^T \right] \right\} \mathbf{w} = \\ &= \left\{ \left[(\mathbf{N}_j \odot \mathbf{A}) \mathbf{e}_{j'} \right]^T \odot \left[(\mathbf{N}_{j'} \odot \mathbf{A}) \mathbf{e}_{j'} \right]^T \right\} \mathbf{w}.\end{aligned}$$

Since $(\mathbf{N}_j \odot \mathbf{A}) \mathbf{e}_{j'} = \mathbf{e}_j \otimes \mathbf{a}_{j'}$ and $(\mathbf{N}_{j'} \odot \mathbf{A}) \mathbf{e}_{j'} = \mathbf{e}_{j'} \otimes \mathbf{a}_{j'}$:

$$\begin{aligned}
h_{j'f'} &= \left[(\mathbf{e}_j^T \otimes \mathbf{a}_{f'}^T) \odot (\mathbf{e}_{j'}^T \otimes \mathbf{a}_f^T) \right] \mathbf{w} = \left[(\mathbf{e}_j^T \otimes \mathbf{a}_{f'}^T) * (\mathbf{e}_{j'}^T \otimes \mathbf{a}_f^T) \right] \mathbf{w} = \left[(\mathbf{e}_j^T * \mathbf{e}_{j'}^T) \otimes (\mathbf{a}_{f'}^T * \mathbf{a}_f^T) \right] \mathbf{w} = \\
&= \delta_{jj'} (\mathbf{a}_{f'}^T * \mathbf{a}_f^T) \mathbf{W} \mathbf{e}_j = \delta_{jj'} (\mathbf{a}_{f'}^T \odot \mathbf{a}_f^T) \mathbf{w}_j = \delta_{jj'} \mathbf{a}_{f'}^T \mathbf{D}_{\mathbf{w}_j} \mathbf{a}_f
\end{aligned}$$

where the last equality holds because $(\mathbf{a}_{f'}^T \odot \mathbf{a}_f^T) \mathbf{w}_j$ is the equation of a rank I PARAFAC model of a $1 \times 1 \times 1$ array. Equation (A.12) follows immediately. \blacksquare

Lemma 6. Let $\mathbf{A} \in \mathbb{R}^{I \times F}$, $\mathbf{B} \in \mathbb{R}^{J \times F}$, $\mathbf{C} \in \mathbb{R}^{K \times F}$ and $\underline{\mathbf{W}} \in \mathbb{R}^{I \times J \times K}$ and let $\mathbf{D}_{\mathbf{w}}$ an $IJK \times IJK$ diagonal matrix holding the elements of the vector $\mathbf{w} = \text{vec } \mathbf{W}_1$. Furthermore, define two matrices $\mathbf{N}_i = \mathbf{e}_i \mathbf{1}_F^T$ and $\mathbf{N}_j = \mathbf{e}_j \mathbf{1}_F^T$, where \mathbf{e}_i denotes the i -th column of the $I \times I$ identity matrix and \mathbf{e}_j denotes the j -th column of the $J \times J$ identity matrix. Then:

$$(\mathbf{C} \odot \mathbf{N}_j \odot \mathbf{A})^T \mathbf{D}_{\mathbf{w}} (\mathbf{N}_k \odot \mathbf{B} \odot \mathbf{A}) = \bar{\mathbf{b}}_j^T \bar{\mathbf{c}}_k^T * (\mathbf{A}^T \mathbf{D}_{\mathbf{w}_{(j,k)}} \mathbf{A}), \quad (\text{A.13})$$

where $\mathbf{D}_{\mathbf{w}_{(j,k)}}$ is a diagonal matrix holding the elements of $\mathbf{w}_{(j,k)}$, which is the vector of the elements of $\underline{\mathbf{W}}$ obtained by fixing the indexes in the second and third mode to j and k respectively.

Proof. First note that the left hand side of (A.13) is the equation of a rank IJK PARAFAC model relative to one slice of the three-way array $\underline{\mathbf{H}} \in \mathbb{R}^{F \times F \times 1}$. Its loading matrices are: $(\mathbf{C} \odot \mathbf{N}_j \odot \mathbf{A})^T$, $(\mathbf{N}_k \odot \mathbf{B} \odot \mathbf{A})^T$, and \mathbf{w} . Therefore, one can write:

$$\mathbf{H}_1 = (\mathbf{C} \odot \mathbf{N}_j \odot \mathbf{A})^T \left[\mathbf{w}^T \odot (\mathbf{N}_k \odot \mathbf{B} \odot \mathbf{A})^T \right]^T,$$

or, equivalently,

$$\text{vec } \mathbf{H}_1 = \text{vec } \mathbf{H}_3^T = \left[(\mathbf{C} \odot \mathbf{N}_j \odot \mathbf{A})^T \odot (\mathbf{N}_k \odot \mathbf{B} \odot \mathbf{A})^T \right] \mathbf{w}$$

Proceeding as in *Lemma 5*, element $h_{j'f'}$ of $\mathbf{H} \equiv \mathbf{H}_1$ is equal to

$$\begin{aligned}
h_{j'f'} &= (\mathbf{e}_{f'}^T \otimes \mathbf{e}_j^T) \left[(\mathbf{C} \odot \mathbf{N}_j \odot \mathbf{A})^T \odot (\mathbf{N}_k \odot \mathbf{B} \odot \mathbf{A})^T \right] \mathbf{w} = \\
&= \left[(\mathbf{c}_{f'} \odot \mathbf{e}_j \odot \mathbf{a})^T \odot (\mathbf{e}_k \odot \mathbf{b}_f \odot \mathbf{a})^T \right] \mathbf{w}.
\end{aligned}$$

Hence, using *Lemma 1* :

$$\begin{aligned}
h_{j'f'} &= \left[(\mathbf{c}_{f'} \odot \mathbf{e}_j \odot \mathbf{a}_{f'})^T * (\mathbf{e}_k \odot \mathbf{b}_f \odot \mathbf{a}_{f'})^T \right] \mathbf{w} = \left[(\mathbf{c}_{f'} \odot \mathbf{e}_j \odot \mathbf{a}_{f'}) * (\mathbf{e}_k \odot \mathbf{b}_f \odot \mathbf{a}_{f'}) \right]^T \mathbf{w} = \\
&= \left[(\mathbf{c}_{f'} * \mathbf{e}_k) \odot (\mathbf{e}_j * \mathbf{b}_f) \odot (\mathbf{a}_{f'} * \mathbf{a}_f) \right]^T \mathbf{w} = b_{jf} c_{kf'} \left[(\mathbf{e}_k \odot \mathbf{e}_j) \odot (\mathbf{a}_{f'} * \mathbf{a}_f) \right]^T \mathbf{w} = \\
&= b_{jf} c_{kf'} \left[(\mathbf{e}_k \odot \mathbf{e}_j)^T \otimes (\mathbf{a}_{f'}^T \odot \mathbf{a}_f^T) \right] \text{vec } \mathbf{W}_1 = b_{jf} c_{kf'} (\mathbf{a}_{f'}^T \odot \mathbf{a}_f^T) \mathbf{W}_1 (\mathbf{e}_k \odot \mathbf{e}_j) = \\
&= b_{jf} c_{kf'} (\mathbf{a}_{f'}^T \odot \mathbf{a}_f^T) \mathbf{w}_{(j,k)} = b_{jf} c_{kf'} \mathbf{a}_f \mathbf{D}_{\mathbf{w}_{(j,k)}} \mathbf{a}_{f'}.
\end{aligned}$$

Equation (A.13) follows immediately. \blacksquare

Lemmas 5 and 6 encompass the general case as the following lemma shows.

Lemma 7. Let $\{\mathbf{A}_n | \mathbf{A}_n \in \mathbb{R}^{I_n \times F}, n=1,2,\dots,N\}$, $\underline{\mathbf{W}} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ and $\mathbf{D}_{\mathbf{w}}$ an $\left(\prod_{n=1 \dots N} I_n \right) \times \left(\prod_{d=1 \dots N} I_n \right)$ diagonal matrix holding the elements of the vector $\mathbf{w} = \text{vec } \mathbf{W}_1$. Define also matrices $\mathbf{N}_i = \mathbf{e}_i \mathbf{1}_F^T$, $\mathbf{N}_{i'} = \mathbf{e}_{i'} \mathbf{1}_F^T$, $\mathbf{N}_j = \mathbf{e}_j \mathbf{1}_F^T$ where \mathbf{e}_i and $\mathbf{e}_{i'}$ are the i -th and i' -th columns of the $I_n \times I_n$ identity matrix and \mathbf{e}_j is the j -th column of the $I_n \times I_n$ identity matrix. For any $n, n' = 1, \dots, N$,

if $n \neq n'$, then

$$\begin{aligned}\mathbf{H} &= (\mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{N}_i \odot \mathbf{A}_{n-1} \dots \odot \mathbf{A}_1)^T \mathbf{D}_{\mathbf{w}} (\mathbf{A}_N \odot \dots \odot \mathbf{A}_{n'+1} \odot \mathbf{N}_j \odot \mathbf{A}_{n'-1} \dots \odot \mathbf{A}_1) = \\ &= \bar{\mathbf{a}}_{n,i}^T \bar{\mathbf{a}}_{n',j}^* \left(\mathbf{Z}_{-nn'}^T \mathbf{D}_{\bar{\mathbf{w}}_{(i,j)}} \mathbf{Z}_{-nn'} \right)\end{aligned}\quad (\text{A.14})$$

if $n = n'$, then

$$\begin{aligned}\mathbf{H} &= (\mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{N}_i \odot \mathbf{A}_{n-1} \dots \odot \mathbf{A}_1)^T \mathbf{D}_{\mathbf{w}} (\mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{N}_{i'} \odot \mathbf{A}_{n-1} \dots \odot \mathbf{A}_1) = \\ &= \delta_{ii'} \mathbf{Z}_{-nn'}^T \mathbf{D}_{\bar{\mathbf{w}}_i} \mathbf{Z}_{-nn'}\end{aligned}\quad (\text{A.15})$$

where $\mathbf{w}_{(i,j)} \equiv (\mathbf{e}_j \otimes \mathbf{e}_i)^T \mathbf{W}_{nn'}$ (i.e. vectorised sub-array obtained by fixing the indexes in the n -th and n' -th mode to i and j respectively), $\bar{\mathbf{w}}_i$ is the i -th row of \mathbf{W}_n , and $\mathbf{D}_{\bar{\mathbf{w}}_{(i,j)}}$ and $\mathbf{D}_{\bar{\mathbf{w}}_i}$ are the diagonal matrices holding the elements of $\bar{\mathbf{w}}_i$ and $\bar{\mathbf{w}}_{(i,j)}$.

Proof. To prove a), since

$$\mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{N}_i \odot \mathbf{A}_{n-1} \dots \odot \mathbf{A}_1 = \mathbf{M}_{nn'} (\mathbf{A}_{n'} \odot \mathbf{N}_i \odot \mathbf{Z}_{-nn'})$$

and

$$\mathbf{A}_N \odot \dots \odot \mathbf{A}_{n'+1} \odot \mathbf{N}_j \odot \mathbf{A}_{n'-1} \dots \odot \mathbf{A}_1 = \mathbf{M}_{nn'} (\mathbf{N}_j \odot \mathbf{A}_n \odot \mathbf{Z}_{-nn'})$$

where $\mathbf{M}_{nn'}$ is a commutation matrix of appropriate size, then

$$\begin{aligned}\mathbf{H} &= [\mathbf{M}_{nn'} (\mathbf{A}_{n'} \odot \mathbf{N}_i \odot \mathbf{Z}_{-nn'})]^T \mathbf{D}_{\mathbf{w}} \mathbf{M}_{nn'} (\mathbf{N}_j \odot \mathbf{A}_n \odot \mathbf{Z}_{-nn'}) = \\ &= (\mathbf{A}_{n'} \odot \mathbf{N}_i \odot \mathbf{Z}_{-nn'})^T \mathbf{M}_{nn'}^T \mathbf{D}_{\mathbf{w}} \mathbf{M}_{nn'} (\mathbf{N}_j \odot \mathbf{A}_n \odot \mathbf{Z}_{-nn'}),\end{aligned}$$

which is the case treated in *Lemma 6* because the matrix $\mathbf{D}'_{\mathbf{w}} \equiv \mathbf{M}_{nn'}^T \mathbf{D}_{\mathbf{w}} \mathbf{M}_{nn'}$ is also diagonal and holds the element of the vector $\mathbf{w}' \equiv \text{vec } \mathbf{W}_{nn'}^T$.

b) is proved in a similar way. Thus, since:

$$\mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{N}_i \odot \mathbf{A}_{n-1} \dots \odot \mathbf{A}_1 = \mathbf{M}_n (\mathbf{N}_i \odot \mathbf{Z}_{-n})$$

and

$$\mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{N}_{i'} \odot \mathbf{A}_{n-1} \dots \odot \mathbf{A}_1 = \mathbf{M}_n (\mathbf{N}_{i'} \odot \mathbf{Z}_{-n}),$$

where \mathbf{M}_n is a suitable commutation matrix, then

$$\mathbf{H} = [\mathbf{M}_n (\mathbf{N}_i \odot \mathbf{Z}_{-n})]^T \mathbf{D}_{\mathbf{w}} \mathbf{M}_n (\mathbf{N}_{i'} \odot \mathbf{Z}_{-n}) = (\mathbf{N}_i \odot \mathbf{Z}_{-n})^T \mathbf{M}_n^T \mathbf{D}_{\mathbf{w}} \mathbf{M}_n (\mathbf{N}_{i'} \odot \mathbf{Z}_{-n}).$$

where $\mathbf{D}'_{\mathbf{w}} \equiv \mathbf{M}_n^T \mathbf{D}_{\mathbf{w}} \mathbf{M}_n$ is again diagonal and holds the element of the vector $\mathbf{w}' = \text{vec } \mathbf{W}_n^T$. Thus one obtains the case treated in *Lemma 5*. \blacksquare

Lemma 8. Let $\mathbf{B} \in \mathbb{R}^{J \times F}$ and $\mathbf{C} \in \mathbb{R}^{K \times F}$, then the Jacobian matrix \mathbf{J} of the function $\mathbf{F}(\mathbf{B}, \mathbf{C}) = \mathbf{C} \odot \mathbf{B}$ is:

$$\mathbf{J} = [\mathbf{J}_{\mathbf{B}} | \mathbf{J}_{\mathbf{C}}] = [\mathbf{M}_1 (\mathbf{I}_J \otimes (\mathbf{I}_F \odot \mathbf{C})) | \mathbf{M}_2 (\mathbf{I}_K \otimes (\mathbf{I}_F \odot \mathbf{B}))], \quad (\text{A.16})$$

where \mathbf{M}_1 and \mathbf{M}_2 are denote appropriate commutation matrices for a $K \times J \times F$ array.

Proof. Note first that

$$\mathbf{Z} \equiv \mathbf{C} \odot \mathbf{B} = (\mathbf{C} \odot \mathbf{B}) \mathbf{I}_F.$$

But $(\mathbf{C} \odot \mathbf{B}) \mathbf{I}_F$ is the equation of a matricised PARAFAC model for a $J \times K \times F$ array. Consequently,

$$\text{vec } \mathbf{Z} = \text{vec} [(\mathbf{C} \odot \mathbf{B}) \mathbf{I}_F] = \text{vec} [\mathbf{B} (\mathbf{I}_F \odot \mathbf{C})^T].$$

It is then apparent that the Jacobian of $\mathbf{C} \odot \mathbf{B}$ is the same as that of a PARAFAC model having as loading matrices $\mathbf{A}_1 = \mathbf{B}, \mathbf{A}_2 = \mathbf{C}$ and $\mathbf{A}_3 = \mathbf{I}_F$. Equation (A.16) follows immediately. ■

Appendix 2. Computation of $\tilde{\mathbf{J}}_{2...N}^T (\tilde{\mathbf{J}}_{\mathbf{Z}_{1:}}^T \mathbf{v})$ and $\tilde{\mathbf{J}}_{\mathbf{Z}_{1:}}^T (\tilde{\mathbf{J}}_{2...N} \mathbf{v})$

The matrices and vectors required by the product $\tilde{\mathbf{J}}_{2...N}^T (\tilde{\mathbf{J}}_{\mathbf{Z}_{1:}}^T \mathbf{v})$ are:

$$\tilde{\mathbf{J}}_{\mathbf{Z}_{1:}}^T = (\mathbf{I}_{I_1} \otimes \mathbf{A}_1) \quad (\text{Magnus and Neudecker, 1999}) \quad (\text{A2.1})$$

$$\tilde{\mathbf{J}}_{2...N} = \mathbf{K}_{FL-1} \left[\tilde{\mathbf{M}}_1 \left(\mathbf{I}_{I_2} \otimes (\mathbf{I}_F \odot \mathbf{Z}_{-\{1,2\}}) \right) \right] \cdots \left[\tilde{\mathbf{M}}_{N-1} \left(\mathbf{I}_{I_N} \otimes (\mathbf{I}_F \odot \mathbf{Z}_{-\{1,N\}}) \right) \right] \quad (\text{A2.2})$$

$$\mathbf{v} \equiv \text{vec } \mathbf{V}_1 \quad (\text{A2.3})$$

where (A2.2) is obtained from *Lemma 8* and the fact that $d(\text{vec } \mathbf{X}^T) = \mathbf{K}_{np}^T d(\text{vec } \mathbf{X})$ for any \mathbf{X} matrix of size $n \times p$. \mathbf{V}_1 denotes the matricised $I_1 \times I_2 \times \dots \times I_N$ array $\underline{\mathbf{V}}$. Combining equations (A2.1) and (A2.3), one obtains

$$\tilde{\mathbf{J}}_{\mathbf{Z}_{1:}}^T \mathbf{v} = (\mathbf{I}_{I_1} \otimes \mathbf{A}_1^T) \text{vec } \mathbf{V}_1 = \text{vec } \mathbf{A}_1^T \mathbf{V}_1 \equiv \text{vec } \tilde{\mathbf{V}}_1, \quad (\text{A2.4})$$

where the array $\tilde{\mathbf{V}}$ has size $F \times I_2 \times \dots \times I_N$. Hence, because of the partitioning of $\tilde{\mathbf{J}}_{2...N}$, the computation of $\mathbf{u} \equiv \tilde{\mathbf{J}}_{2...N}^T (\tilde{\mathbf{J}}_{\mathbf{Z}_{1:}}^T \mathbf{v})$ can be written as:

$$\begin{aligned} \mathbf{J}_{2...N}^T \tilde{\mathbf{v}} &= \left(\left[\tilde{\mathbf{M}}_1 \left(\mathbf{I}_{I_2} \otimes (\mathbf{I}_F \odot \mathbf{Z}_{-\{1,2\}}) \right) \right] \cdots \left[\tilde{\mathbf{M}}_{N-1} \left(\mathbf{I}_{I_N} \otimes (\mathbf{I}_F \odot \mathbf{Z}_{-\{1,N\}}) \right) \right] \right)^T \mathbf{K}_{FL-1}^T \text{vec } \tilde{\mathbf{V}}_1 = \\ &= \left(\left[\tilde{\mathbf{M}}_1 \left(\mathbf{I}_{I_2} \otimes (\mathbf{I}_F \odot \mathbf{Z}_{-\{1,2\}}) \right) \right] \cdots \left[\tilde{\mathbf{M}}_{N-1} \left(\mathbf{I}_{I_N} \otimes (\mathbf{I}_F \odot \mathbf{Z}_{-\{1,N\}}) \right) \right] \right)^T \text{vec } \tilde{\mathbf{V}}_1^T = \\ &= \begin{bmatrix} \left(\tilde{\mathbf{M}}_1 \left(\mathbf{I}_{I_2} \otimes (\mathbf{I}_F \odot \mathbf{Z}_{-\{1,2\}}) \right) \right)^T \text{vec } \tilde{\mathbf{V}}_1^T \\ \vdots \\ \left(\tilde{\mathbf{M}}_{N-1} \left(\mathbf{I}_{I_N} \otimes (\mathbf{I}_F \odot \mathbf{Z}_{-\{1,N\}}) \right) \right)^T \text{vec } \tilde{\mathbf{V}}_1^T \end{bmatrix}. \end{aligned} \quad (\text{A2.5})$$

Thus, the partition \mathbf{u}_n relative to the n -th mode is:

$$\mathbf{u}_n = \left[\mathbf{I}_{I_n} \otimes (\mathbf{I}_F \odot \mathbf{Z}_{-\{1,n\}}) \right]^T \tilde{\mathbf{M}}_{n-1}^T \text{vec } \tilde{\mathbf{V}}_1^T \quad (\text{A2.6})$$

It can be easily verified that the matrix $\tilde{\mathbf{M}}_{n-1}^T$ performs the following operation:

$$\tilde{\mathbf{M}}_{n-1}^T \text{vec } \tilde{\mathbf{V}}^{(I_{-1} \times F)} = \text{vec } \tilde{\mathbf{V}}^{(I_{-\{1,n\}} F \times I_n)}. \quad (\text{A2.7})$$

Hence,

$$\mathbf{u}_n = \left[\mathbf{I}_{I_n}^T \otimes (\mathbf{I}_F \odot \mathbf{Z}_{-\{1,n\}})^T \right] \text{vec } \tilde{\mathbf{V}}^{(I_{-\{1,n\}} F \times I_n)} = \text{vec} \left[\left(\mathbf{I}_F \odot \mathbf{Z}_{-\{1,n\}} \right)^T \tilde{\mathbf{V}}^{(I_{-\{1,n\}} F \times I_n)} \right]. \quad (\text{A2.8})$$

A simple reordering of the terms of the Khatri-Rao product and a rows-permutation of $\tilde{\mathbf{V}}^{(I_{-\{1,n\}} F \times I_n)}$ allow for a more convenient notation. Therefore, (A2.8) can be written as

$$\mathbf{u}_n = \text{vec} \left[\left(\mathbf{Z}_{-\{1,n\}} \odot \mathbf{I}_F \right)^T \tilde{\mathbf{V}}_n^T \right]. \quad (\text{A2.9})$$

The product $\tilde{\mathbf{J}}_{\mathbf{Z}_{1:}}^T (\tilde{\mathbf{J}}_{2...N} \mathbf{v})$ allows similar simplifications. The matrix $\tilde{\mathbf{J}}_{2...N}$ and the vector \mathbf{v} can be partitioned in $N-1$ blocks, each referring to the parameters in one loading matrix. Thus,

$$\tilde{\mathbf{J}}_{2\dots N} \mathbf{v} = \sum_{n=2}^N \tilde{\mathbf{J}}_n \mathbf{v}_n = \mathbf{K}_{FL-1} \sum_{n=2}^N \tilde{\mathbf{M}}_{n-1} \left(\mathbf{I}_{I_n} \otimes \left(\mathbf{I}_F \odot \mathbf{Z}_{-\{1,n\}} \right) \right) \mathbf{v}_n \quad (\text{A2.10})$$

where the matrices $\tilde{\mathbf{M}}_{n-1}$ are multi-way commutation matrices defined on an $I_2 \times \dots \times I_N \times F$ array according to *Lemma 8*. If, a matrix \mathbf{V}_n of dimensions $F \times I_n$ such that $\text{vec } \mathbf{V}_n = \mathbf{v}_n$, equation (A2.10) can be written as:

$$\begin{aligned} \tilde{\mathbf{J}}_{2\dots N} \mathbf{v} &= \mathbf{K}_{FL-1} \sum_{n=2}^N \tilde{\mathbf{M}}_{n-1} \left(\mathbf{I}_{I_n} \otimes \left(\mathbf{I}_F \odot \mathbf{Z}_{-\{1,n\}} \right) \right) \text{vec } \mathbf{V}_n = \\ &= \mathbf{K}_{FL-1} \sum_{n=2}^N \tilde{\mathbf{M}}_{n-1} \text{vec} \left[\left(\mathbf{I}_F \odot \mathbf{Z}_{-\{1,n\}} \right) \mathbf{V}_n \right] \equiv \mathbf{K}_{FL-1} \sum_{n=2}^N \tilde{\mathbf{M}}_{n-1} \text{vec } \tilde{\mathbf{V}}_n \end{aligned} \quad (\text{A2.11})$$

where $\text{vec } \tilde{\mathbf{V}}_n \equiv \text{vec} \left[\left(\mathbf{I}_F \odot \mathbf{Z}_{-\{1,n\}} \right) \mathbf{V}_n \right]$. The commutation matrix can be removed by an appropriate permutation of the terms of the Khatri-Rao product:

$$\begin{aligned} \tilde{\mathbf{M}}_{n-1} \text{vec } \tilde{\mathbf{V}}_n &= \tilde{\mathbf{M}}_{n-1} \left(\mathbf{V}_n^T \odot \mathbf{I}_F \odot \mathbf{Z}_{-\{1,n\}} \right) \mathbf{1}_F = \\ &= \left(\mathbf{I}_F \odot \mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{V}_n^T \odot \mathbf{A}_{n-1} \dots \odot \mathbf{A}_2 \right) \mathbf{1}_F \end{aligned} \quad (\text{A2.12})$$

Therefore:

$$\begin{aligned} \sum_{n=2}^N \tilde{\mathbf{M}}_{n-1} \text{vec } \tilde{\mathbf{V}}_n &= \sum_{n=2}^N \left(\mathbf{I}_F \odot \mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{V}_n^T \odot \mathbf{A}_{n-1} \dots \odot \mathbf{A}_2 \right) \mathbf{1}_F = \\ &= \left[\mathbf{I}_F \odot \sum_{n=2}^N \left(\mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{V}_n^T \odot \mathbf{A}_{n-1} \dots \odot \mathbf{A}_2 \right) \right] \mathbf{1}_F = \\ &= \text{vec} \left\{ \left[\sum_{n=2}^N \left(\mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{V}_n^T \odot \mathbf{A}_{n-1} \dots \odot \mathbf{A}_2 \right) \right] \mathbf{I}_F \right\} = \\ &= \text{vec} \left[\sum_{n=2}^N \left(\mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{V}_n^T \odot \mathbf{A}_{n-1} \dots \odot \mathbf{A}_2 \right) \right] \equiv \\ &\equiv \text{vec } \tilde{\mathbf{V}}^{(L-1 \times F)} \end{aligned} \quad (\text{A2.13})$$

Finally, since $\mathbf{K}_{FL-1} \text{vec } \tilde{\mathbf{V}}^{(L-1 \times F)} = \text{vec } \tilde{\mathbf{V}}^{(F \times L-1)}$, one can write

$$\tilde{\mathbf{J}}_{\mathbf{Z}^T} \left(\tilde{\mathbf{J}}_{2\dots N} \mathbf{v} \right) = \left(\mathbf{I}_{L-1} \otimes \mathbf{A}_1 \right) \text{vec } \tilde{\mathbf{V}}^{(F \times L-1)} = \text{vec } \mathbf{A}_1 \tilde{\mathbf{V}}^{(F \times L-1)} \quad (\text{A2.14})$$

In terms of number of operations, the computational advantage is of the same magnitude as for $\mathbf{J}^T \mathbf{v}$.

Appendix 3. Fast ALS iteration

For an N -way array $\underline{\mathbf{X}}$ of size $I_1 \times \dots \times I_N$, an iteration of the ALS algorithm for PARAFAC can be synthesised as

$$\mathbf{A}_n^{s+1} = \mathbf{X}_n \left(\mathbf{Z}_{-n}^s \right)^{+T} \quad \text{for } n = 1 \dots N, \quad (\text{A3.1})$$

where $\mathbf{Z}_{-n}^s = \mathbf{A}_N^{s+1} \odot \dots \odot \mathbf{A}_{n+1}^{s+1} \odot \mathbf{A}_{n-1}^s \dots \odot \mathbf{A}_1^s$, $s = 0, 1, \dots$ denotes the iteration number, $+$ is the Moore-Penrose inverse and the set \mathbf{A}_n^0 for $n = 2, \dots, N$ are some initial estimates of the last $N-1$ loading matrices. Using a well-known property of Moore-Penrose inverse (Schott, 1997), equation (A3.1) is typically solved as (Harshman, 1970):

$$\mathbf{A}_n^{s+1} = \mathbf{X}_n \mathbf{Z}_{-n}^s \left(\mathbf{E}_{-n}^s \right)^+ \quad \text{for } n = 1 \dots N, \quad (\text{A3.2})$$

where $\mathbf{E}_{-n}^s \equiv \left(\mathbf{Z}_{-n}^s \right)^T \mathbf{Z}_{-n}^s$. The following lemma, which yields expressions similar to those obtained using the chain rule for the $\mathbf{J}^T \mathbf{v}$ product, can reduce the computation load by a factor proportional to the ratio between the size of $\underline{\mathbf{X}}$ in the largest mode and F .

Lemma 9. Given an N -way array $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ and the set of matrices $\left\{ \mathbf{A}_n \mid \mathbf{A}_n \in \mathbb{R}^{I_n \times F}, n = 1, 2, \dots, N \right\}$, the following holds:

$$\mathbf{X}_n \mathbf{Z}_{-n} = \tilde{\mathbf{X}}^{(I_n \times I_{-\{1,n\}} F)} (\mathbf{I}_F \odot \mathbf{Z}_{-\{1,n\}}) \quad \text{for } n = 2 \dots N, \quad (\text{A3.3})$$

where $\tilde{\mathbf{X}}^{(F \times I_2 \dots I_N)} \equiv \mathbf{A}_1^T \mathbf{X}_1$ and \mathbf{I}_F is an $F \times F$ identity matrix.

Proof. Let \mathbf{Y} be an $(N-1)$ -way array of size $I_1 \times \dots \times I_{n-1} \times I_{n+1} \times \dots \times I_N$ and \mathbf{K}_n denote the commutation matrix $\mathbf{K}_{I_1(I_2 \dots I_{n-1} I_{n+1} \dots I_N)}$. By definition, \mathbf{K}_n performs the operation:

$$\mathbf{Y}^{(I_1 \dots I_{n-1} I_{n+1} \dots I_N \times 1)} = \text{vec } \mathbf{Y}^{(I_1 \times I_2 \dots I_{n-1} I_{n+1} \dots I_N)} = \mathbf{K}_n \text{vec } \mathbf{Y}^{(I_2 \dots I_{n-1} I_{n+1} \dots I_N \times I_1)} = \mathbf{K}_n \mathbf{Y}^{(I_2 \dots I_{n-1} I_{n+1} \dots I_N I_1 \times 1)}.$$

Thus, it follows immediately that

$$\mathbf{X}_n^T \equiv \mathbf{X}^{(I_1 \dots I_{n-1} I_{n+1} \dots I_N \times I_n)} = \mathbf{K}_n \mathbf{X}^{(I_2 \dots I_{n-1} I_{n+1} \dots I_N I_1 \times I_n)}. \quad (\text{A3.4})$$

Proceeding as in Lemma 3, it can also be demonstrated that:

$$\mathbf{Z}_{-n} = \mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \dots \odot \mathbf{A}_1 = \mathbf{K}_n (\mathbf{A}_1 \odot \mathbf{Z}_{-\{1,n\}}). \quad (\text{A3.5})$$

From, (A3.4) and (A3.5), it follows that

$$\mathbf{X}_n \mathbf{Z}_{-n} = \mathbf{X}_n \mathbf{K}_n (\mathbf{A}_1 \odot \mathbf{Z}_{-\{1,n\}}) = \mathbf{X}^{(I_n \times I_{-\{1,n\}} I_1)} (\mathbf{A}_1 \mathbf{I}_F \odot \mathbf{I}_{I_{-\{1,n\}}} \mathbf{Z}_{-\{1,n\}}) = \mathbf{X}^{(I_n \times I_{-\{1,n\}} I_1)} (\mathbf{A}_1 \otimes \mathbf{I}_{I_{-\{1,n\}}}) (\mathbf{I}_F \odot \mathbf{Z}_{-\{1,n\}}),$$

where $\mathbf{X}^{(I_n \times I_{-\{1,n\}} I_1)} \equiv \mathbf{X}^{(I_n \times I_2 \dots I_{n-1} I_{n+1} \dots I_N I_1)}$, and

$$\begin{aligned} \text{vec} \left[\mathbf{X}^{(I_n \times I_{-\{1,n\}} I_1)} (\mathbf{A}_1 \otimes \mathbf{I}_{I_{-\{1,n\}}}) \right] &= (\mathbf{A}_1^T \otimes \mathbf{I}_{I_{-\{1,n\}}}) \text{vec } \mathbf{X}^{(I_n \times I_{-\{1,n\}} I_1)} = (\mathbf{A}_1^T \otimes \mathbf{I}_{I_{-1}}) \text{vec } \mathbf{X}^{(I_n \times I_{-\{1,n\}} I_1)} \\ &= (\mathbf{A}_1^T \otimes \mathbf{I}_{I_{-1}}) \text{vec } \mathbf{X}^{(I_n \times I_{-\{1,n\}} I_1)} = (\mathbf{A}_1^T \otimes \mathbf{I}_{I_{-1}}) \text{vec } \mathbf{X}^{(I_n I_{-\{1,n\}} \times I_1)} = \text{vec} \left(\mathbf{X}^{(I_n I_{-\{1,n\}} \times I_1)} \mathbf{A}_1 \right) \equiv \text{vec } \tilde{\mathbf{X}}^{(I_n I_{-\{1,n\}} \times F)}. \end{aligned} \quad \blacksquare$$

Using (A3.3), the ALS iteration becomes

$$\begin{aligned} \mathbf{A}_1^{s+1} &= \mathbf{X}_1 \mathbf{Z}_{-1}^s (\mathbf{E}_{-1}^s)^+ \\ \mathbf{A}_n^{s+1} &= \left[\tilde{\mathbf{X}}^{(I_n \times I_{-\{1,n\}} F)} (\mathbf{I}_F \odot \mathbf{Z}_{-\{1,n\}}^s) \right] (\mathbf{E}_{-1}^s)^+ \quad \text{for } n \geq 2, \end{aligned} \quad (\text{A3.6})$$

where $\tilde{\mathbf{X}}^{(I_{-1} \times F)} = \mathbf{X}_1^T \mathbf{A}_1^{s+1}$. It can easily be verified that (A3.6) essentially corresponds to (A2.9) apart from the computation of $(\mathbf{E}_{-1}^s)^+$ and the different matricisation of $\tilde{\mathbf{X}}$.

Appendix 4. Exact line search

The univariate problem of finding the step length in a specific direction that minimises the residuals between a PARAFAC model and an N -way array can be expressed as

$$\alpha^* = \arg \min_{\alpha} \left\| \text{vec } \mathbf{X}_1 - [(\mathbf{A}_N + \alpha \Delta \mathbf{A}_N) \odot \dots \odot (\mathbf{A}_1 + \alpha \Delta \mathbf{A}_1)] \mathbf{I}_F \right\|_2^2, \quad (\text{A4.1})$$

where α is the sought step length.

Owing to the distributivity of the Khatri-Rao product, it holds that

$$\begin{aligned} (\mathbf{A}_N + \alpha \Delta \mathbf{A}_N) \odot \dots \odot (\mathbf{A}_1 + \alpha \Delta \mathbf{A}_1) &= \mathbf{Z} + \alpha (\Delta \mathbf{A}_N \odot \mathbf{Z}_{-N} + \dots + \mathbf{Z}_{-1} \odot \Delta \mathbf{A}_1) + \\ &+ \alpha^2 (\Delta \mathbf{A}_N \odot \Delta \mathbf{A}_{N-1} \odot \mathbf{Z}_{-\{N,N-1\}} + \dots + \mathbf{Z}_{-\{1,2\}} \odot \Delta \mathbf{A}_2 \odot \Delta \mathbf{A}_1) + \dots + \alpha^N \Delta \mathbf{A}_N \odot \dots \odot \Delta \mathbf{A}_1 \end{aligned} \quad (\text{A4.2})$$

Thus let \mathbf{T}_m define the sum of the Khatri-Rao products including m Δ -terms:

$$\mathbf{T}_0 = \mathbf{Z} \quad (\text{A4.3})$$

$$\mathbf{T}_1 = \sum_{n=1}^N \mathbf{M}_n (\Delta \mathbf{A}_n \odot \mathbf{Z}_{-n}) \quad (\text{A4.3b})$$

$$\mathbf{T}_m = \sum_{\mathcal{C}_m} \mathbf{M}_{\{n_1, \dots, n_m\}} (\Delta \mathbf{A}_{n_m} \odot \dots \odot \Delta \mathbf{A}_{n_1} \odot \mathbf{Z}_{-\{n_1, \dots, n_m\}}) \quad (\text{A4.3c})$$

$$\mathbf{T}_N = \Delta \mathbf{A}_N \odot \dots \odot \Delta \mathbf{A}_1, \quad (\text{A4.3d})$$

where $\mathcal{C}_m = \{\{n_1, n_2, \dots, n_m\} \mid n_1, n_2, \dots, n_m \in \{1, 2, \dots, N\}, n_1 \neq n_2 \dots \neq n_m\}$.

Then, problem (A4.1) can be written as:

$$\alpha^* = \arg \min_{\alpha} \left\| \text{vec } \mathbf{X}_1 - \left(\sum_{m=0}^N \alpha^m \mathbf{T}_m \right) \mathbf{1}_F \right\|_2^2 = \arg \min_{\alpha} \left\| \text{vec } \mathbf{R}_1 - \left(\sum_{m=1}^N \alpha^m \mathbf{T}_m \right) \mathbf{1}_F \right\|_2^2, \quad (\text{A4.4})$$

or, equivalently,

$$\alpha^* = \arg \min_{\alpha} \left(\mathbf{r}^T \mathbf{r} + \sum_{m=1}^N \alpha^m \mathbf{r}^T \mathbf{t}_m + \sum_{m=1}^N \sum_{m'=1}^N \alpha^{m+m'} \mathbf{t}_m^T \mathbf{t}_{m'} \right), \quad (\text{A4.5})$$

where $\mathbf{t}_m = \mathbf{T}_m \mathbf{1}_F$. Therefore, one can see that solving (A4.1) corresponds to finding the real root that minimises a $2N$ -th degree polynomial. The i -th degree polynomial coefficient is the sum of the elements of the matrix $[\text{vec } \mathbf{R}_1 \quad \mathbf{t}_1 \quad \dots \quad \mathbf{t}_N]^T [\text{vec } \mathbf{R}_1 \quad \mathbf{t}_1 \quad \dots \quad \mathbf{t}_N] \equiv \mathbf{U}$ whose sum of row and column indexes is equal to $i + 2$.

Unfortunately, each \mathbf{t}_m entails the summation of $\binom{N}{m}$ Khatri-Rao products of N terms (whose cost is approximately equivalent to that of the same number of function evaluations) and the explicit computation of \mathbf{U} costs approximately $N^2 M$ operations. Thus, it is easy to verify that computing the coefficients for (A4.5) is somewhat more costly than $\sum_{m=1}^N \binom{N}{m} = 2^N - 1$ function evaluations. In other words, while there is little point for a three-way problem in allowing more than seven function evaluations for the line search because then it would be possible to fit a sixth degree polynomial that gives the exact solution, exact line search computed in this fashion clearly becomes unfeasible already for arrays of order four.

However, great savings can be attained by noticing that:

$$\mathbf{t}_m = \begin{cases} \text{vec} [\mathbf{A}_1 \mathbf{B}_m^T + \Delta \mathbf{A}_1 \mathbf{B}_{m-1}^T] & \text{for } m = 1 \dots N-1 \\ \text{vec} \Delta \mathbf{A}_1 \mathbf{B}_{N-1}^T & \text{for } m = N \end{cases}, \quad (\text{A4.6})$$

where \mathbf{B}_m denotes the sum of the Khatri-Rao products of all the possible combinations of $N - m - 1$ loading matrices and m Δ -terms for the last $N - 1$ modes. That is,

$$\mathbf{B}_m = \sum_{c=1}^{\binom{N}{m}} \mathbf{M}_{\{n_1, n_2, \dots, n_m\}_c} (\Delta \mathbf{A}_{n_m} \odot \Delta \mathbf{A}_{n_{m-1}} \dots \odot \mathbf{A}_{n_1} \odot \mathbf{Z}_{-\{1, n_1, n_2, \dots, n_m\}_i}) \quad (\text{A4.7})$$

where $\{n_1, n_2, \dots, n_m\}_c \subseteq \{2, \dots, N\}$ denotes the c -th subset with m Δ -terms, $\binom{N}{m}$ is the number of such combinations and \mathbf{M} denotes a commutation matrix for an $N - 1$ array. For example, for a four-way array:

$$\mathbf{B}_2 = (\Delta \mathbf{A}_4 \odot \Delta \mathbf{A}_3 \odot \mathbf{A}_2 + \Delta \mathbf{A}_4 \odot \mathbf{A}_3 \odot \Delta \mathbf{A}_2 + \mathbf{A}_4 \odot \Delta \mathbf{A}_3 \odot \Delta \mathbf{A}_2).$$

Owing to the recursive nature of (A4.6), only 2^{N-1} Khatri-Rao products of $N - 1$ terms need be computed and the equivalent of $2N - 1$ function evaluations (namely products of an $I_1 \times F$ matrix and with an $F \times I_{-1}$ one) is required.

In spite of the great savings implied by (A4.6), the computation of \mathbf{U} requires the storage of all \mathbf{t}_m 's and may be too expensive for larger arrays. However, since $(\text{vec } \mathbf{R}_1)^T \text{vec} (\mathbf{A}_1 \mathbf{B}_m^T) = \text{tr} (\mathbf{R}_1 \mathbf{B}_m \mathbf{A}_1^T) = \text{tr} (\mathbf{A}_1^T \mathbf{R}_1 \mathbf{B}_m)$, whereby $\text{tr}(\mathbf{X})$ denotes the trace of \mathbf{X} (Magnus and Neudecker, 1999), the elements of the first row and column of \mathbf{U} for the least squares case can be computed as

$$u_{m1} = u_{1m} = \text{tr}(\mathbf{A}_1^T \tilde{\mathbf{R}}_m) + \text{tr}(\Delta \mathbf{A}_1^T \tilde{\mathbf{R}}_{m-1}) \quad (\text{A4.8})$$

where $\tilde{\mathbf{R}}_m \equiv \mathbf{R}_1 \mathbf{B}_m$.

The elements of rows and columns other than the first can be calculated in a similar way, but this would imply computing the cross products of all the possible combinations of the \mathbf{B}_m matrices and their storage (note that (A4.8) requires the storage of only one). Again, this can be problematic and expensive for larger arrays. On the other hand, property (A.2) can be used to obtain the cross products between any two such matrices rather cheaply and with the sole additional storage of $3N$ matrices of size $F \times F$. Let

$$\mathbf{E}_{\{n_1, \dots, n_m\}} \equiv \mathbf{A}_{n_1}^T \mathbf{A}_{n_1} * \dots * \mathbf{A}_{n_m}^T \mathbf{A}_{n_m}$$

$$\hat{\mathbf{E}}_{\{n_1, \dots, n_m\}} \equiv \Delta \mathbf{A}_{n_1}^T \Delta \mathbf{A}_{n_1} * \dots * \Delta \mathbf{A}_{n_m}^T \Delta \mathbf{A}_{n_m}$$

and

$$\mathbf{E}_{\Delta \{n_1, \dots, n_m\}} \equiv \Delta \mathbf{A}_{n_1}^T \mathbf{A}_{n_1} * \dots * \Delta \mathbf{A}_{n_m}^T \mathbf{A}_{n_m}.$$

Then, any product $\mathbf{B}_m^T \mathbf{B}_{m'}$ can be written as

$$\mathbf{B}_m^T \mathbf{B}_{m'} = \sum \mathbf{E}_{\mathcal{A}_1} * \mathbf{E}_{\Delta \mathcal{A}_2}^T * \mathbf{E}_{\Delta \mathcal{A}_3} * \hat{\mathbf{E}}_{\mathcal{A}_4} \quad (\text{A4.9})$$

where $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$, and \mathcal{A}_4 are proper subsets of $\{2, \dots, N\}$ such that $\bigcup_i \mathcal{A}_i = \{2, \dots, N\}$ and $\mathcal{A}_i \cap \mathcal{A}_{i'} = \emptyset$ for any $i \neq i'$.

The number of summations to calculate all the elements in \mathbf{U} that do not belong to the first row or column is approximately equal to the number of combinations for a set of $2(N-1)$ elements. Hence, the number of operations required is in the order of $NF^2 2^{2(N-1)}$ and obviously almost negligible compared to the computation of the \mathbf{B}_m 's and of $\tilde{\mathbf{R}}_m$.

Paper IV

Correlation Optimized Warping and Dynamic Time
Warping as preprocessing methods for chroma-
tographic data

G. Tomasi, F. van den Berg and C. A. Andersson

Journal of Chemometrics, 18, 2004, 231-241.

Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data

Giorgio Tomasi*, Frans van den Berg and Claus Andersson

The Royal Veterinary and Agricultural University (KVL), Department of Food Science, Food Technology, Rolighedsvej 30, DK-1958 Frederiksberg C, Denmark

Received 16 May 2003; Revised 23 March 2004; Accepted 16 April 2004

Two different algorithms for time-alignment as a preprocessing step in linear factor models are studied. Correlation optimized warping and dynamic time warping are both presented in the literature as methods that can eliminate shift-related artifacts from measurements by correcting a sample vector towards a reference. In this study both the theoretical properties and the practical implications of using signal warping as preprocessing for chromatographic data are investigated. The connection between the two algorithms is also discussed. The findings are illustrated by means of a case study of principal component analysis on a real data set, including manifest retention time artifacts, of extracts from coffee samples stored under different packaging conditions for varying storage times. We concluded that for the data presented here dynamic time warping with rigid slope constraints and correlation optimized warping are superior to unconstrained dynamic time warping; both considerably simplify interpretation of the factor model results. Unconstrained dynamic time warping was found to be too flexible for this chromatographic data set, resulting in an over-compensation of the observed shifts and suggesting the unsuitability of this preprocessing method for this type of signals. Copyright © 2004 John Wiley & Sons, Ltd.

KEYWORDS: DTW; COW; warping; retention time shift; PCA

1. INTRODUCTION

'Shift' is a common occurrence in chemistry. Many analytical techniques yield data where the same phenomena may yield variations at different positions (e.g. retention times in a chromatogram, wavelengths in NIR spectroscopy due to temperature influences) or may have different 'durations' depending on the specific analytical conditions. Analogously, the measurements for the single samples can have different time scales or axes, or the sample vectors may have different lengths (e.g. different batch lengths in industrial processes).

Warping is one of the numerous pretreatment methods that have been proposed to correct for shifts, conditioning data for multilinear models like PCA, PLS or PARAFAC for exploratory purposes as well as quantitative determination by alignment of the shifted variables [1]. As will be discussed later on, if data are not brought to a form where the observed variables of the samples under analysis express similar attributes, the required assumption for using bi- and multi-

linear modeling, namely that like variables represent similar phenomena in all samples, is violated.

In this paper, a general framework will be given to employ the two warping algorithms on chromatographic data, and their connection is illustrated using a data set from a food research experiment as a case study. The subject of this research was the effect of different packaging conditions on changes in ground coffee composition during storage over several weeks. The time-shift problem in the chromatograms is depicted in Figure 1. The problem results from a clear deterioration in the columns separation performance over time. An additional difficulty is the confounding of the column performance with storage time, the factor of primary interest in this experiment.

2. THEORY

Two different warping algorithms have received much attention in recent years for the alignment of time trajectories, chromatographic profiles and spectra [2–4]. The first method, termed dynamic time warping (DTW), was initially devised for aligning frequency spectra of words pronounced by different speakers for recognition purposes [5,6]. The more recent approach for aligning signals, termed correlation

*Correspondence to: G. Tomasi, The Royal Veterinary and Agricultural University (KVL), Department of Food Science, Food Technology, Rolighedsvej 30, DK-1958 Frederiksberg C, Denmark. E-mail: gt@kvl.dk

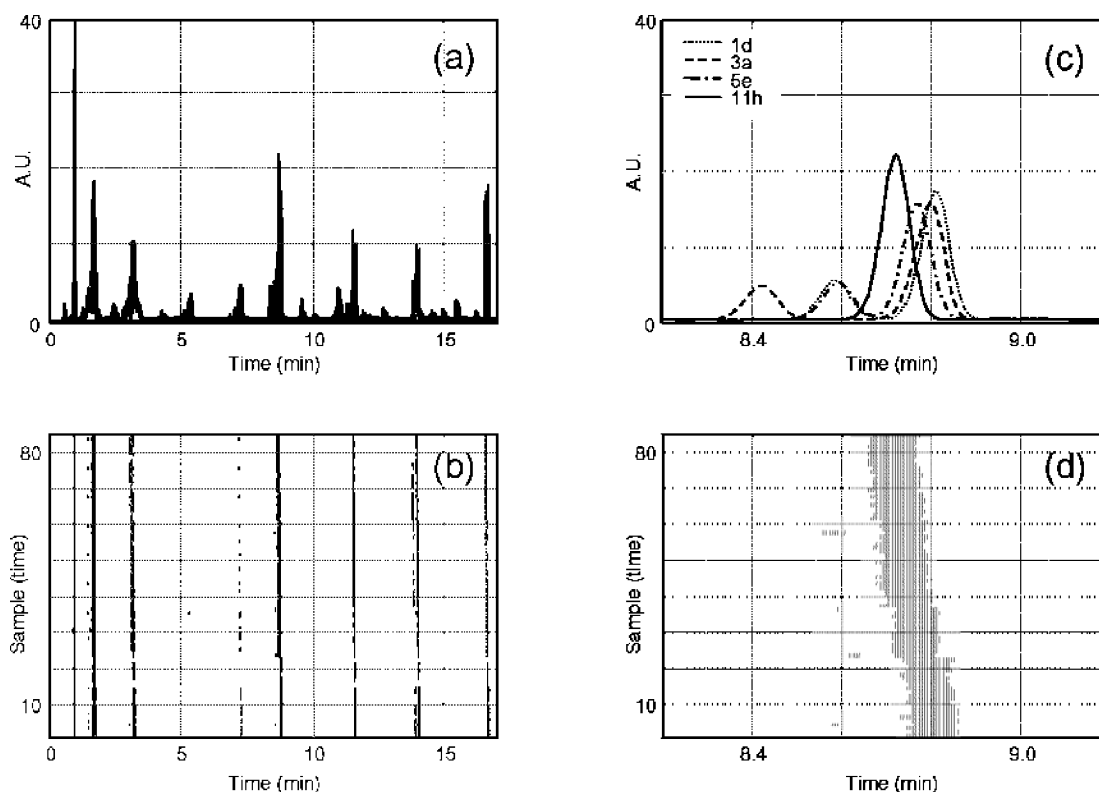


Figure 1. The shift problem as observed in the chromatograms of coffee extracts: (a) raw chromatograms for four selected samples 4(1d), 15(3a), 33(5e) and 84(11h); (b) top-plot of the whole data set (cut-off 5 a.u.; note sample order equals analysis order); (c) detail plot for one peak area in raw data; (d) top-plot for selected peak area (notice the systematic retention-time reduction for the main peak original found at 8.8 minutes observed in all samples). This trend is observed for all the common components in the data set.

optimized warping (COW), was proposed in 1998 as a means to correct chromatograms for shifts in the time axis prior to multivariate modeling [7].

The features of both warping algorithms will be studied in relation to PCA bilinear modeling for exploratory data analysis [8–10]. In PCA, the number of retained components expresses the complexity of the observed variations in data and may be regarded as the number of independent phenomena that cause the observed variations among objects and variables, e.g. the chemical rank of the data set in case of NIR spectroscopy. The first principal components are the ones that capture the boldest patterns; henceforth, more and more refined information may be captured by successive principal components. However, aside from the mathematical aspects of rank there are more practical implications [10]. Data sets often require preprocessing before the actual modeling step. This preprocessing effectively reduces the rank, leading to more parsimonious, more robust and better interpretable models. In NIR spectroscopy removing, for example, baseline offsets or using derivative spectra is common practice, thereby eliminating what are considered instrumental artifacts, usually of no interest to the experimenter. If these artifacts are not removed beforehand, they may form a relevant pattern in the data, frequently obscuring the important information. Time-shifts as discussed in this paper form another category of artifacts and the warping algorithms proposed can be seen as preprocessing steps applicable to factor models such as PCA.

2.1. Nomenclature and terminology

In the present work, the focus is on correcting non-analytical changes in the time-axis of samples by optimizing similarity with a reference sample. Hence, two measurements of similar nature are involved in each alignment operation in the warping procedures: the 'reference' and the 'sample' (designated by the letters 'r' and 's', respectively). The direction along which the warping is performed is simply referred to as *time*.

Throughout this work, italics are used for scalars (e.g. m) and bold for vectors (e.g. \mathbf{m}). N and M indicate the vector lengths for reference and sample, respectively. The m th element ($m = 1, 2, \dots, M$) in the *time* mode of the sample is designated by $s(m)$. Vectors \mathbf{n} and \mathbf{m} will be used to denote element indexing in reference and sample. For example, if $\mathbf{n} = [2 \ 3 \ 4]$, $\mathbf{r}(\mathbf{n}) = \mathbf{r}([2 \ 3 \ 4])$ selects those elements from the reference indexed by the entries in \mathbf{n} . A special reservation is made for $\mathbf{s}\{\mathbf{n}\}$ where the braces indicate entries in the sample vector estimated by interpolation, corresponding to matching reference points with index \mathbf{n} .

2.2. Correlation optimized warping

To correct for misalignments or shifts in discrete data signals, a procedure called COW was introduced by Nielsen *et al.* [7]. It is a piecewise or segmented data preprocessing method (operating on one sample record at a time) aimed at aligning a sample data vector towards a reference vector by allowing limited changes in segments lengths on the sample vector. The ratio between the number of points in the

penultimate segment. Since not restricted by the optimization space for the algorithm as defined here, the user could select segment length and slack so that border points can pass over each other (make loops or knots in the time line). This undesirable situation has to be prevented by putting additional restrictions into the algorithm, effectively eliminating these potential knot-points during the optimization search.

In accordance with the DP principles, the optimum for the last border positioning in step I equals the global optimum and the optimal path is found by determining the global optimum from the last local optimum and all its predecessors. In this implementation a simple summation of the local measures of correlation is used, but alternatives are again possible (e.g. the product or a weighted sum). Once known, tracking back the optimal path, positioning all the borders at the right position and finding warped/aligned sample segments by linear interpolation will reconstruct the best-matching preprocessed sample signal for the predetermined set of parameters I and t .

Notice that the degree of flexibility increases for segment border points in the middle of the data vector compared to the edges. The total flexibility of the border positions in the example of Figure 2 shows the binomial-like structure 1–3–9–3–1 (1 for the two boundary points, 9 for the center border). This observation, together with algorithm parameters segment length i and slack parameter t , determines the corrective power of the COW preprocessing.

2.3. Dynamic time warping

Dynamic time warping 'nonlinearly warps the two trajectories in such a way that similar events are aligned and a minimum distance between them is obtained' [7]. The algorithm was first presented by Sakoe and Chiba [12] and further developed in numerous papers [5,6,13–15]. In recent years it has found application in chromatography [2–4], in batch process monitoring [16–20] and in gene expression studies [21,22]. The general algorithm is described in great detail in these publications and will be only briefly outlined in this section. Much focus will be put on the constraints and the synchronization step, which are essential to illustrate the connection between DTW and COW and critical to yielding a meaningful alignment of chromatographic profiles.

2.3.1. The algorithm

The warping path F is a mapping of the sample and the reference time axes on a common time axis

$$F = \langle [m(k), n(k)] | k = 1, \dots, K \rangle \quad (2)$$

where K is the length of the common time axis. The k th element of F , $[m(k), n(k)]$, contains the indexes for the sample and the reference at the k th point on the common (warped) time axis. Figure 3 shows an example mapping grid and illustrates the concept of the warping path.

The global optimization problem in DTW can be written as follows [6]:

$$\underset{F}{\operatorname{argmin}} D(F) = \frac{\sum_{k=1}^K d_{rs}[m(k), n(k)] w(k)}{\sum_{k=1}^K w(k)} \quad (3)$$

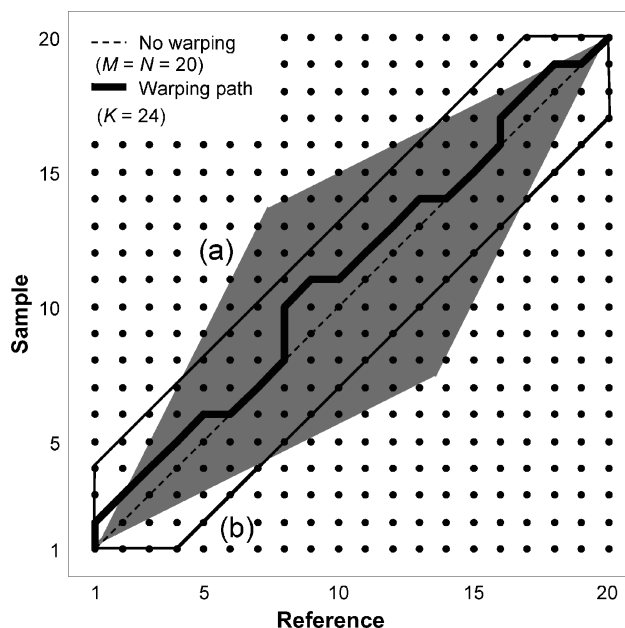


Figure 3. Mapping grid with nomenclature, the abscissa is the time axis for the reference, the ordinate is the time axis for the sample. The solid line is the warping path, i.e. the common time axis k . Feasible areas related to local continuity constraints for table $T^{(3,1)}$ (a) and band constraint with $A = 4$ (b).

where $d_{rs}[m(k), n(k)]$ is a dissimilarity measure, e.g. the squared Euclidean distance between $r[n(k)]$ and $s[m(k)]$, and $w(k)$ are suitable weights.

In order for problem (3) to be solved correctly, $m(k)$ and $n(k)$ must not decrease along the common time axis to avoid creating knots in the time axis and trivial solutions as $F = \langle (1, 1), (M, N) \rangle$ must be prevented. Therefore (3) is subjected to the so-called 'local continuity constraints' of the form:

$$\begin{aligned} 0 &\leq m(k+1) - m(k) \leq a \\ 0 &\leq n(k+1) - n(k) \leq b \end{aligned} \quad (4)$$

where a and b are integers and positive. For example, for $a = b = 1$, there are only three feasible predecessors for $[m(k), n(k)]$: $[m(k) - 1, n(k)]$, $[m(k) - 1, n(k) - 1]$ and $[m(k), n(k) - 1]$.

Typically, the end points for F are fixed and equal to $(1, 1)$ and (M, N) , i.e. the initial (and the final) entries in reference and sample are constrained to be the same on the common time axis as well.

Moreover, if no weights are used, the solution to problem (3) is biased towards shorter paths, which involve the summation of fewer terms [13,22]. The main purpose of weights is thus to remove such a bias, rendering the optimal distance (and consequently the reconstructed F) independent of the length of the warping path [20]. Not all weighting schemes fulfill this purpose [13], and although some bias may be desirable to decrease the occurrence of extreme warps [17,20], it also raises problems in establishing which optimization criterion is actually employed. Furthermore, stricter local continuity constraints (described in the next section) can be used to avoid excessive corrections in a more straightforward fashion.

The adopted weighting scheme was initially proposed in Sakoe and Chiba [6] and uses as weight for point $[m(k+1), n(k+1)]$:

$$w(k+1) = m(k+1) - m(k) + n(k+1) - n(k) \quad (5)$$

The sum of such weights over any warping path going from (1,1) to (M,N) is equal to $M+N$ and is thus independent of K .

If reference and sample are treated equally, with respect to both weights and local continuity constraints, the warping is defined as symmetric [13]. The asymmetric option (e.g. if $a \neq b$) will not be described, as the principals are identical [13].

Under these restrictions, problem (3) can be solved using dynamic programming. Assuming that the global distances up to the allowed predecessors to (m,n) have been computed, the optimal global distance to point, (m,n) is the sum of $d_{rs}(m,n)$, multiplied by a suitable weight depending on the local path, and the minimum among the global distances to any of the allowed predecessors.

Identifying the global optimum for the warping path is thus transformed in an efficient iterative procedure divided into a forward step and a backward step:

- (1) Starting from point (1,1) and according to (4) construct the mapping grid $D(M \times N)$, in which element d_{mn} is the optimal accumulated distance up to point (m,n) (the forward step).
- (2) Find the optimal warping path by tracing backwards [i.e. from $(m(K), n(K)) = (M, N)$], once again in accordance with (4), the minimal accumulated distance up to point (1,1) (the backward step).

2.3.2. Constraints

More elaborate constraints have been devised for the DTW problem than the essential ones mentioned thus far. Their use may yield warping paths more in agreement with *a priori* knowledge that may be available for the problem at hand, avoiding unfeasible compressions or expansions of reference or sample signal. As will be shown in the experimental section, such constraints are necessary to successfully apply dynamic time warping to the chromatographic data used for this application.

For a generalization of the local continuity constraints, it is necessary to introduce the concept of transition: an elementary transition describes the single advancement in the common time axis (corresponding to the single arrow in the graphs of Table 1).

The warping path can be described by a sequence of elementary transitions going from one end point to the other and its local behavior can be restricted by allowing only certain sequences of elementary transitions. Such a local series is referred to as 'rule'. The collection of rules used is named 'lookup table' and denoted by $T^{(x,y)}$, x being the largest block distance covered by any of the rules in the table and y the maximum number of horizontal/vertical consecutive transitions allowed for by the table. For example, the local continuity constraints of equation (4) with $a = b = 1$ can be translated into the lookup table:

$$T^{(2,\infty)} \equiv \begin{cases} t_1 = (1, 1) \\ t_2 = (1, 0) \\ t_3 = (0, 1) \end{cases} \quad (6)$$

Table 1. Examples of different lookup tables with complex transition rules. The arrows represent the single elementary transition. They point in the direction of the common time axis and start from a legal predecessor to the element in the mapping grid toward which they point

Constraint ^(A)		Graph
Name	Slope: Min – Max	
$T^{(2,\infty)}$	0 – ∞	
$T^{(3,1)}$	$\frac{1}{2}$ – 2	
$T^{(5,1)}$	$\frac{2}{3}$ – $\frac{3}{2}$	
$T^{(7,1)}_{COW}$	$\frac{2}{3}$ – $\frac{4}{3}$	

^AThe first index is the largest block distance covered by any step in the table, the second is the maximum number of horizontal/vertical transitions allowed by any step.

where the infinity indicates that this particular lookup table does not impose any restriction on the number of consecutive horizontal/vertical transitions and the two integers in parentheses are, respectively, the advancements in the sample and in the reference time axes corresponding to the elementary transition t_x .

Apart from $T^{(2,\infty)}$ (which is referred to in the remaining part of this work as the 'unconstrained DTW'), all lookup tables limit the grid points that can be reached given the end points [13]. The feasible part of the grid is typically delimited by four lines passing through either (1,1) or (M,N) and having slope equal to the minimum or the maximum slope allowed by the lookup table [13]. An example of the resulting lozenge is depicted in Figure 3.

The corrective power of the single lookup table is a function of the minimum and maximum slopes it allows for the warping path and the number of points spanned by rules in the lookup table. The closer the former are to M/N (hence 1 when sample and reference are of equal length), the more rigid are the constraints.

A second relevant type of constraint limits the feasible area to a band delimited by two lines of slope M/N . These lines pass at $|M - N| + A$ points from (M,N), where A is an arbitrary integer, defining the maximum compression/expansion in time-points of the sample and reference with respect to their original lengths [6]. Figure 3 shows the feasible area around the diagonal of the mapping grid. Although they prevent extreme behaviors of the warping path, band-constraints alone are not adequate for this purpose and additional local restrictions to the number of consecutive vertical or horizontal transitions are still required to prevent the optimal path from moving from the 'top' line to the 'bottom' line delimiting the search space [22].

2.3.3. Synchronization

The symmetric DTW algorithm yields a warped reference and a warped sample of identical length K (see Figure 3). If a warping correction takes place, K will be larger than either M or N . The extent of elongation is unpredictable until the warping process is finished and may vary from sample to sample. Therefore, an additional synchronization step rendering vectors of length N is required, if sample vectors are to be stacked for bilinear modeling. This synchronization step is not part of the original DTW algorithm, which merely used the optimal distance for classification purposes [6,22], and is necessarily asymmetric since horizontal vs vertical transitions (or reference vs sample time-points) are treated differently. To synchronize, one can take the average of the measurements at the different sample points forming a sequence of vertical transitions (e.g. reference point 8 in Figure 3). The rationale in this approach is that, by using the average, all the information in the sample time-points is taken into account. Alternatively, one could use an asymmetric warping algorithm that maps the sample time axis on the reference time axis. In this case, $n(k) = k$ and K would be equal to N . This choice may, however, lead to information loss and discontinuities in the warped sample because some points are ignored [20].

Furthermore, if slope constraints are imposed and the distinct points spanned by the rules comprising vertical or horizontal transitions are deemed as forming a segment in the sample, it is possible to use interpolation, analogously to COW. In this case, the series of distinct sample time-points determined by the optimal transition is interpolated to estimate a new series of points of length equal to the corresponding one in the reference. For example, if the series includes a horizontal transition followed by three diagonal ones (e.g. reference points 5 to 8 in Figure 3), four distinct points are involved for the reference, but only three for the sample. In order to yield the same length after the synchronization, the three points must be interpolated to four.

2.4. The connection between DTW and COW

Although COW and DTW are treated in the literature as two distinct solutions to the warping problem, there is a connection between the two that helps to shed some light on the success of COW applied to chromatographic data [1,7] opposed to the very poor results yielded by DTW on these data (see the Experimental section).

This link can be established from a combination of DTW-constraints and interpolation.

First of all, COW, expressed in a DTW framework, requires imposing the condition that rules including horizontal or vertical transitions (i.e. those correcting the shift) can be applied starting only at fixed points of the reference. Stated differently, given a lookup table of the form $T_{\text{COW}}^{(7,1)}$ in Table 1, only a limited set of points on the reference are candidate end points of a rule (namely, the gray and white dots in the graph).

Lookup tables of the form $T_{\text{COW}}^{(2i+y,y)}$ represent a further restriction to 'slope constraints' and each rule spans the same number of points i in the reference, whereas the number of distinct (i.e. not repeated) points in the sample can vary from

$i-y$ to $i+y$. Hence, i is equivalent to the segment length as was previously defined for the COW algorithm and y corresponds to the slack parameter.

Note that tables like $T_{\text{COW}}^{(2i+y,y)}$ alone are not sufficient to guarantee equivalence between COW and DTW: local continuity constraints only require that any subsection of the warping path complies with one of the rules in the lookup table and a sequence of, for example, 10 consecutive diagonal transitions would not violate any local constraint in $T_{\text{COW}}^{(7,1)}$. Thus, the constraint on the initial/final points for the rules is necessary. When such restrictions are applied, all points in the lozenge of Figure 3 remain feasible, but the allowed end points form $I+1$ vertical (one-point-wide) bands within the feasible area.

Under the above constraints the local distance at the end points for each rule in the DTW algorithm may be the correlation coefficient as in COW, although problem (3) should be changed from minimization to maximization.

Moreover, if interpolation is applied to the distinct sample points of each rule prior to the computation of the distance when their number differs from i , and afterwards in the synchronization step, one yields an algorithm that is almost identical to COW. Note that, because of this interpolation, the position of the vertical or horizontal transitions in the single rules of tables $T_{\text{COW}}^{(2i+y,y)}$ need not be uniquely defined to yield equivalence and in Table 1 it is set as the 'last' in the series only for simplicity.

To yield complete equivalence, one further constraint needs to be applied to the warping path because of the condition set in COW that the first point of one segment is adjacent to the last of the previous one. In the DTW context, this means that a diagonal transition with zero weight (to remove the influence of this transition from the optimal distance) connects the boundaries for the two segments.

Hence, COW may be regarded as a special case of DTW where additional constraints are added to reduce the search space for the optimal warping and to employ correlation coefficient as optimization criterion. The shape-preserving features of COW and the quality of the warping [1,7] thus appear linked to the relative rigidity of the slope constraints of the corresponding warping paths rather than the focus on correlation instead of the Euclidean distance.

Nielsen *et al.* [7] suggest that the segment length should be at least equal to the width of the smallest feature one wants to align (e.g. peaks in chromatography) and that lower values may result in the alignment of noise or other non-chemical information and alterations in peak shapes. Depending on the chromatographic analysis, this value may vary, but in the case study presented below it is approximately 30 points (same order as the largest shift observed). The corresponding slope constraints are significantly stricter than in standard DTW, where tables $T_{\text{COW}}^{(2,\infty)}$ and $T_{\text{COW}}^{(3,1)}$ are most often employed [4,16–18,20]. From this view point, unconstrained (or loosely constrained) DTW is expected to be too flexible and to deform peaks and features. Using rigid slope constraints may, however, prevent adequate correction at the two ends of the chromatogram because of the fixed end point assumption. Even though such restrictions on the path can be relaxed by modifying the algorithm [1,20], similar results may be obtained by appending to the sample (and/or

to the reference) a segment of suitable length containing only white noise that is removed again after warping.

DTW with rigid slope constraints (DTWc in the rest of the paper) and Euclidean distance presents a clear advantage over COW, as one can avoid setting restrictions on the rules' end points (i.e. fixed boundaries for the reference segments can be dropped) and the correction for shift is not bound to any specific position in the reference. However, this flexibility comes at the price of a considerably larger number of feasible end points (all those within the gray lozenge of Figure 3) and, consequently, computational cost. In this respect the COW algorithm has the advantage of being simple to implement, in general faster and less memory-demanding [7,20]. Furthermore, the additional flexibility granted by the DTWc may not be necessary, in which case COW would yield a perfectly acceptable solution.

3. DATA AND EXPERIMENTAL CONDITIONS

The data set is obtained by gas chromatographic (GC) analysis of extracts from ground coffee. The sample set is built up from material stored under different packaging conditions (combinations of gas headspace, vacuum and temperature) for different time durations. The instrumentation used was a GC Shimadzu 14A chromatograph (Shimadzu, Tokyo, Japan) with flame ionization detection (FID) and a Supelcowax 10 column: 30 m, 0.53 mm i.d., 1.0 μ m film (Supelco, Bellefonte, PA, USA). Temperatures in the system were 280°C at the injector, 260°C in the oven, and 260°C at the detector. Internal standard was Caprylic acid (C8). Each sample was extracted by ether, and the extract was injected into a GC carrier gas-flow for in-line thermally assisted methylation and the output from a FID was recorded for 17 min. The first 2000 points were used for warping and PCA ($N = M = 2000$ data points).

The separation of the gas sample into fatty acids of different sizes results in the lightest and most volatile molecules (<C5) to be detected within the first 4 min, whereas the larger and less volatile molecules (>C25) leave the system after 16 min. All chromatograms were individually baseline-corrected by subtracting the average signal for the first 120 s from the full chromatogram prior to modeling.

The typical time for a chromatographic column to wear out of course depends on the application and in particular on the number of analyses applied on the column material. However, a clear continuous drop in performance can be observed by visual inspection (see Figure 1), the incentive to the warping study described in this paper. Figure 1(a,c) shows four typical samples, Figure 1(b,d) shows an overview of the systematic variation of the entire set. The figure clearly shows the column material deteriorating over time, giving rise to shorter retention times. As mentioned previously, this column deterioration is confounded with the order of collecting samples. Because coffee extracts of this nature cannot be stored for a long time, randomization of GC analysis over the experimental condition storage time is not feasible (see below). Based on visual inspection of the 88 different samples, a total of four samples were considered outliers and were removed from the data set. From *a priori* knowledge about the sample set, chromatogram no. 9 (hence, a sample from the beginning of the measurement series) was chosen as reference for warping, since this sample contained the highest number of the chemical constituents compared with all the other samples collected at the start of the experiment. Objects in the sample series are labeled by two digits: numbers '1'–'11' indicating the different sampling times during the storage experiments, 'a'–'h' indicating the eight different packaging conditions. In this notation, the reference sample is labeled '2a'.

As the scope of this study is limited to the time warping as preprocessing, the finer details of the experiments are not given here.

4. RESULTS AND DISCUSSION

In this section the different effects of various DTW and COW implementations will be illustrated. A PCA model was fitted on the coffee extract GC data set without any shift-correcting pretreatment, meaning no warping or data mean centering. A detail of the loadings is shown in Figure 4(a), and a score plot for the first two PCs in Figure 5(a). The first principal component is identified as representing the average chromatogram. A moderate grouping according to packaging can be observed in the tendency of PC1 where the most expensive and best quality treatment (type 'g') is isolated

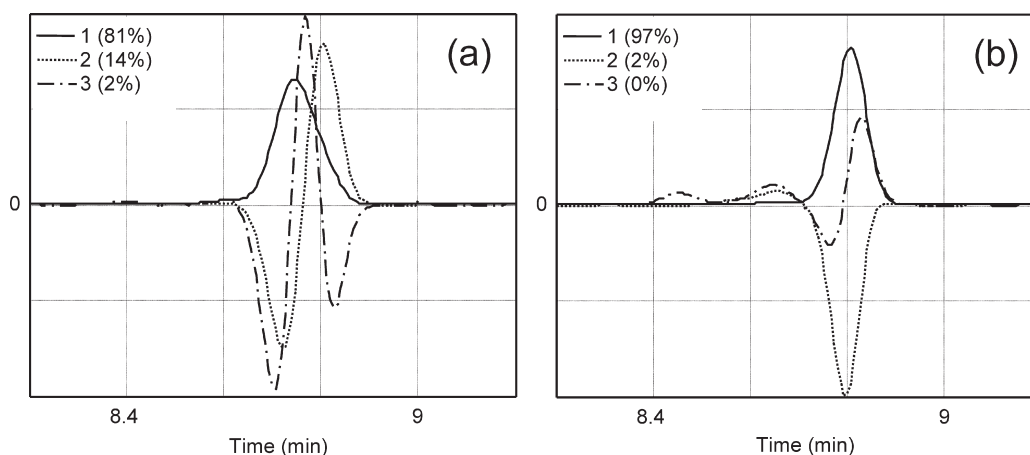


Figure 4. (a) PCA loading details raw coffee data; (b) PCA loading details after COW correction.

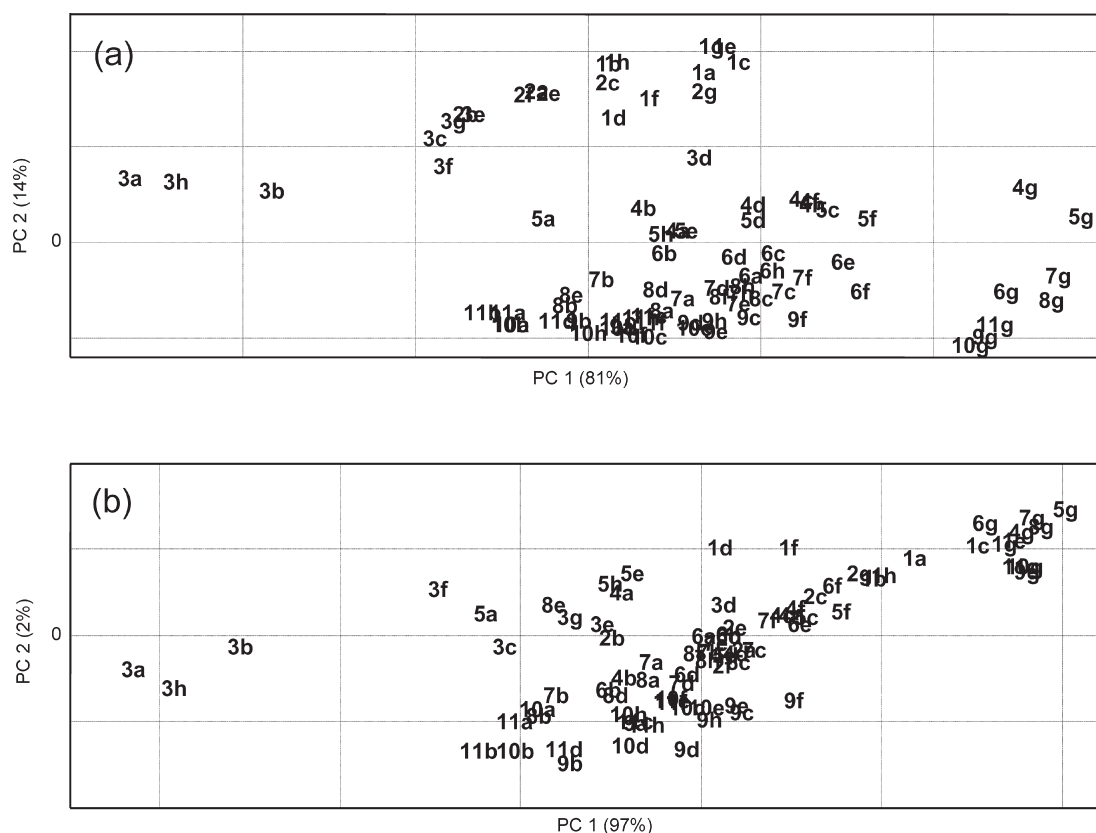


Figure 5. Score plot (a) raw coffee data (b) after COW correction.

from the others. It was anticipated during the setup of the experiment that this packaging type would show little to no effect over storage time. The second PC has a loading vector that resembles the first derivative of the loading vector for PC1. This 'Taylor series' development is characteristic for PCA modeling of data sets with shift phenomena [23]. As a consequence of the time-shifts in the chromatograms (see Figure 1), the PC1 vs PC2 score plot shows a typical 'horseshoe' configuration, visible in Figure 5(a). This shape is not related to the storage time and packaging 'g' is particularly suited to show this. Visual inspection of the set of 11 chromatograms from series 'g' is consistent with the observation that the curved shape in the score plot is almost exclusively caused by retention time shifts in the chromatograms, not by chemical changes. Moreover, what may appear as two large groupings (i.e. '1', '2' and '3' on one side and '4' to '11' on the other) is the primary consequence of the retention time shifts, following the curved pattern just described.

In conclusion, the PCA components for untreated data describe both chemical information and 'artifacts' due to the shift confounded with the experimental factor storage time (e.g. the location of the '3' samples in the score plot, in particular '3a', '3b' and '3h', is due to an altered composition of the extracts probably related to the fact that the samples in this series were stored in a freezer for several days up to analysis). Exploratory PCA modeling of the untreated (unwarped) data set is then clearly unfeasible.

In order to verify the change in the quality of the warping for DTW as the slope constraints become more and more strict, several lookup tables have been applied

[Figure 6(a, b, d)]. A locally unconstrained DTW algorithm using a $T^{(2,\infty)}$ lookup table was applied with band constraints. Band limitation A was set equal to 200, corresponding to a maximum $\pm 10\%$ allowed correction for the chromatograms, significantly larger than the observed 20–30 points maximum shift. For the synchronization step, averaging was used [20]. The results of this preprocessing were very unsatisfactory [Figure 6(a)], e.g. small peaks like those found in the region 8.4–8.6 min in Figure 1(c) disappear. They are completely merged with the bigger one found in every chromatogram in the range 8.7–8.8 min. The disappearance of the small peaks is not caused by the warping itself, but by the averaging in the synchronization step, which compresses all information occurring during long sequences of vertical transitions [4]. Synchronization is also responsible for a second type of artifact related to the peak height: when the sample peak is larger than the matching one in the reference, the former is cut in height. Conversely, when the sample peak is smaller, its top element is repeated until the two sides of the synchronized peak match those on the reference, resulting in a plateau. This behavior cannot be the consequence of the described warping procedure, which is symmetric. When the reference peak is smaller, its top is repeated until the sides overlap the larger sample peak. In the warping path this appears as a sequence of vertical transitions, for which corresponding sample points are successively averaged. Thus, after the synchronization the sample peak is still taller than the reference peak, but the computed average may still be significantly lower than the original peak height. Note that normalization of reference and sample vector to length one, as suggested in some

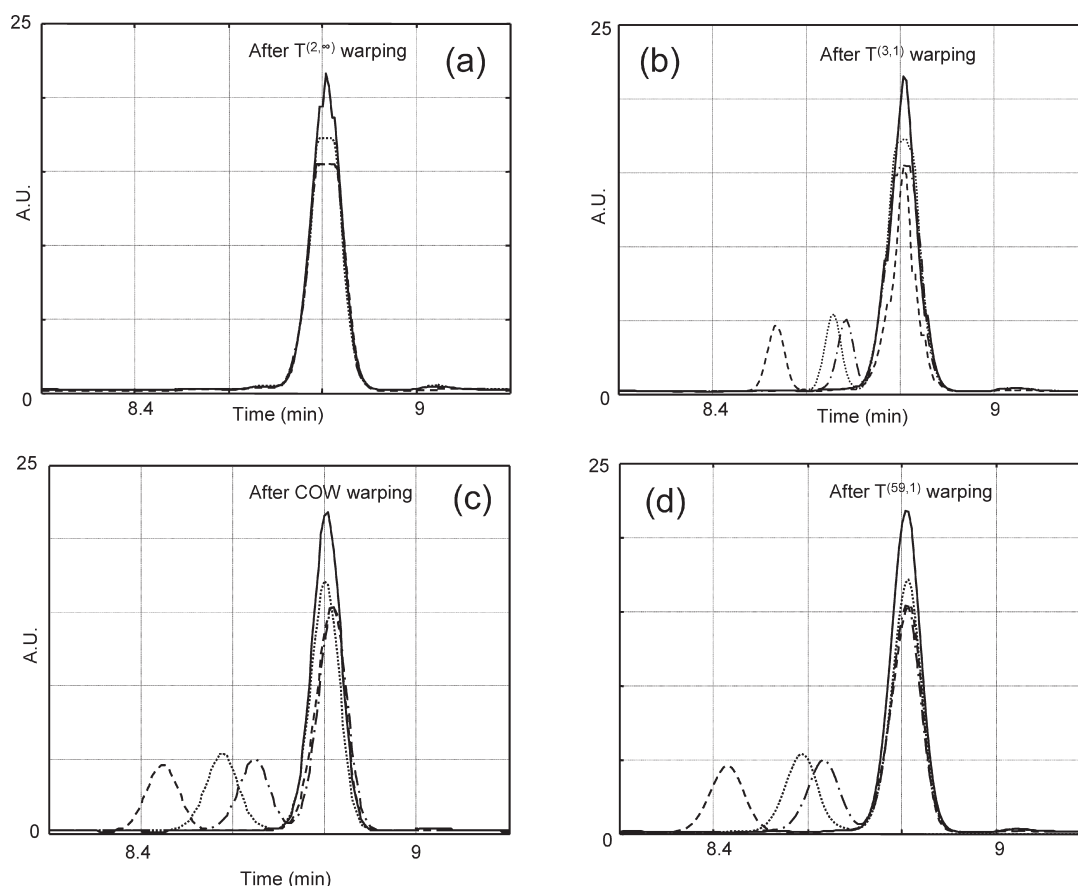


Figure 6. Signal detail reconstruction after (a) $T^{(2,\infty)}$ warping; (b) $T^{(3,1)}$ warping; (c) COW; (d) $T^{(59,1)}$ warping.

publications, does not remedy these errors. These artifacts were observed at various retention times for different peaks and substantially altered the chemical diversity present in the data, confirming the inadequacy of unconstrained DTW for the correct realignment of chromatograms. It should be emphasized that a different choice of reference (e.g. one that contains the extra peaks) is somewhat helpful in preserving the peaks like those around 8.3–8.6 min, but cannot be considered as a solution to make unconstrained DTW viable. In fact, an analysis of the shifts in the raw data indicates that the small peaks around 8.3–8.6 min are distinct and a $T^{(2,\infty)}$ warping aligns them as if they were one, which is evidently incorrect. It would be very difficult and time-consuming to find the references that contain all the peaks or even most of them. Moreover, using alternative references would not avoid the artifacts, but only show them at different positions in the chromatogram.

Figure 6(b) shows some details of the warped samples using lookup table $T^{(3,1)}$ with 10% band constraints and averaging. A clear reduction in the occurrence of artifacts, both the misalignments and peak-top kind, can be observed from this figure. In particular, the peaks at 8.4–8.6 are no longer merged with the larger one at 8.7–8.8, and are maintained separate. Nonetheless, their widths are modified beyond need as a result of the several vertical transitions (albeit alternated with diagonal ones because of the constraints) in the warping path. Repeated points (due to horizontal transitions) also modify the shape of the peaks,

but the effect can be partly reduced using interpolation in the synchronization step (not shown).

In any case, the quality of the alignment cannot be compared with the best one obtained with COW, using a segment length $i = 100$ time-points ($I = 20$ segments) and a slack of $t = 3$ time-points [Figure 6(c)]. The limited correcting flexibility and the interpolation still allowed the correct alignment of the peaks around 8.7–8.8 min while maintaining their original shape and the series of smaller peaks before this retention time are visible as individual entities and not deformed.

The results of DTW are comparable to those of COW only when very rigid constraints are used [Figure 6(d)]. As anticipated, the rules in the lookup table should be larger than the peaks and the best results were obtained with a $T^{(59,1)}$ warping (i.e. with rules spanning 30 points) and using interpolation for the synchronization step. The results do not completely overlap those of COW, as the optimization criterion is different (the squared Euclidean distance was used in DTW), but the discrepancies are marginal and the warped chromatograms are considerably better than those obtained with shorter rules. Note that the corrective power of $T_{\text{COW}}^{(203,3)}$ and $T^{(59,1)}$ is in practice identical. Shorter segments are required for DTWc because in this algorithm interpolation is not concurrent to the computation of the optimal local distances.

When rigid slope constraints were applied, a residual retention time shift, albeit limited to one single point (before or after the position of the matching peak in the reference) at

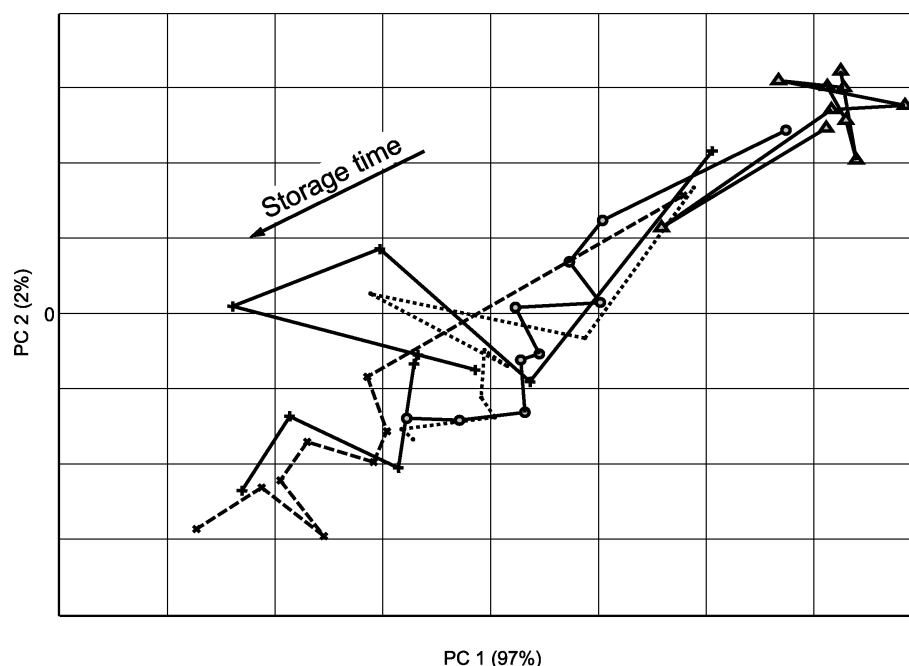


Figure 7. PCA score plots for coffee data set after $T^{(59,1)}$ warping and removal of storage time '3'. Clustering of packaging 'g' (\blacktriangle), and trends of packagings 'a' (+), 'b' (x), 'c' (\bullet), and 'h' (...). The scores relative to the other packagings were removed from the plot for clarity.

the peak's top, could be observed and could not be removed with either COW or DTWc, regardless of the choice of segment length and slack. This residual shift seems to be related to the slightly different peak shape due to noise and the widening associated with column deterioration.

Although the correction was satisfactory in the central part of the chromatograms with both COW and DTWc, as expected, the quality of the realignment degraded at the end of the chromatogram (starting approximately at 11–12 min). Appending a segment of 500 points with low intensity white noise to both sample and reference prior to warping and removing it afterwards effectively dealt with the problem without any modification to the warping algorithm.

From a thorough visual inspection of the warped chromatograms, none of the two algorithms distinctively emerged as better than the other.

The score plot in Figure 5(b) shows a storage-time trend in the first PC, with '1' and 'g' (the most expensive packaging) on the right and '11' on the left. The time-'3' packaging can be considered an outlier, probably the consequence of freezing the extracts before analysis. If this last category is removed, an even clearer storage time trend (particularly packagings 'a', 'b', 'c' 'd' and 'h') and clustering in packaging conditions (namely the already mentioned high-quality 'g') is found, revealing the different capabilities in preserving coffee volatiles (Figure 7). In particular, the second component is related to the ratio between low and high molecular weight fatty acids and the third component describes the 'extra' peaks like those found at 8.3 minutes. Such peaks are present in a relatively small number of samples and are likely indexes of chemical degradation of some of the fatty acids in the ground coffee. They precede all the 'standard' peaks by 0.2–0.5 min and are present only in a few samples of

four packaging types ('a', 'd', 'e' and 'h'), but show no evident trend over the entire set. DTWc and COW gave nearly identical results in the PCA analysis (not shown), confirming again the substantial equivalence of the two methods as preprocessing methods for bilinear modeling.

5. CONCLUSIONS

In this paper two different algorithms—correlation optimized warping and dynamic time warping—were studied as a preprocessing step in (bilinear) factor modeling. Both the theoretical properties and some practical implications were investigated on the basis of a case study of chromatographic data vectors with retention time artifacts and a connection between the two methods has been established. The most relevant conclusion is that time alignment corrections should be handled with great care. Simple correction schemes can lead to severe distortion of the signal, and unconstrained (or loosely constrained) DTW is clearly too flexible for the coffee case study and most likely for chromatographic data in general. More rigid settings for both methods were found to be successful as alignment operation, making an exploratory PCA analysis of the coffee data set much more interpretable.

More generally, it should be kept in mind that the original DTW was constructed for pattern recognition, thus with the aim of minimizing the distance between the profile and the possible match. In this context, it has been shown repeatedly that rigid slope constraints increase the error rate [6]. Conversely, when applied to chromatographic data (or for alignment of other time trajectories or even spectra), great attention should be paid to the original shape retaining as much as possible. Thus, slope and segment lengths (rules)

should be chosen so that the warping is as rigid as possible within the limits given by the problems in question.

This observation conflicts to some extent with earlier findings of other researchers on chromatographic data vectors [4]. However, in these earlier works an artificial sinusoidal baseline was introduced [7]. We believe that experience has shown that signal processing without proper de-trending can lead to unrealistic conclusions [24]. Therefore, we opted for studying and showing local alignment capabilities instead of more global behaviors.

Another non-trivial step is finding the correct reference vector. If no prior knowledge is available on the data set or experimental conditions, any vector should potentially be able to serve as such, possibly leading to very different solutions. It is also possible to generate or simulate a reference vector (e.g. the first PCA scaled loading-vector from untreated data), again leading to a distinct preprocessing.

For practical purposes, the COW algorithm and DTWc are relatively insensitive towards the parameters values. Any reasonable choice for segment length and slack will give an indication of the anticipated synchronization performance. Furthermore, due to the relatively small search space, a trial and error approach for finding the best settings is feasible even on a modest computer system. In this respect, COW may be more suitable than DTW, as the number of possible paths is much smaller and, consequently, the memory requirements are significantly reduced.

REFERENCES

1. Bylund D, Danielsson R, Malmquist G, Markides KE. Chromatographic alignment by warping and dynamic programming as a pre-processing tool for PARAFAC modelling of liquid chromatography-mass spectrometry data. *J. Chromatogr. A* 2002; **961**: 237–244.
2. Reiner E, Abbey LE, Moran TF, Papamichalis P, Shafer RW. Characterization of normal human cells by pyrolysis-gas-chromatography mass spectrometry. *Biomed. Mass Spectrom.* 1979; **6**: 491–498.
3. Wang CP, Isenhour TL. Time-warping algorithm applied to chromatographic peak matching gas-chromatography Fourier-transform infrared mass-spectrometry. *Anal. Chem.* 1987; **59**: 649–654.
4. Pravdova V, Walczak B, Massart DL. A comparison of two algorithms for warping of analytical signals. *Anal. Chim. Acta* 2002; **456**: 77–92.
5. Itakura F. Minimum prediction residual principle applied to speech recognition. *IEEE Trans. ASSP* 1975; **AS23**: 67–72.
6. Sakoe H, Chiba S. Dynamic-programming algorithm optimization for spoken word recognition. *IEEE Trans. ASSP* 1978; **26**: 43–49.
7. Nielsen NPV, Carstensen JM, Smedsgaard J. Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping. *J. Chromatogr. A* 1998; **805**: 17–35.
8. Jolliffe IT. *Principal Components Analysis* (2nd edn). Springer: New York, 2002.
9. Jackson JEA. *User's Guide to Principal Components*. Wiley-Interscience: New York, 1991.
10. Munck L, Norgaard L, Engelsen SB, Bro R, Andersson CA. Chemometrics in food science—a demonstration of the feasibility of a highly exploratory, inductive evaluation strategy of fundamental scientific significance. *Chemometr. Intell. Lab. Syst.* 1998; **44**: 31–60.
11. Hillier FS, Lieberman GJ. *Introduction to Operations Research* (7th edn). McGraw-Hill: New York, 2001.
12. Sakoe H, Chiba S. *Proceedings of the International Congress of Acoustics*, Budapest, 1971, paper 20 C13.
13. Myers C, Rabiner LR, Rosenberg AE. Performance trade-offs in dynamic time warping algorithms for isolated word recognition. *IEEE Trans. ASSP* 1980; **28**: 623–635.
14. Sakoe H. 2-Level Dp-matching—dynamic programming-based pattern-matching algorithm for connected word recognition. *IEEE Trans. ASSP* 1979; **27**: 588–595.
15. Myers CS, Rabiner LR. A level building dynamic time warping algorithm for connected word recognition. *IEEE Trans. ASSP* 1981; **29**: 284–297.
16. Kassidas A, MacGregor JF, Taylor PA. Synchronization of batch trajectories using dynamic time warping. *AIChE J.* 1998; **44**: 864–875.
17. Ramaker HJ, van Sprang ENM, Westerhuis JA, Smilde A. Dynamic Time Warping of spectroscopic BATCH data. *Anal. Chim. Acta* 2003; **498**: 133–153.
18. Kassidas A, Taylor PA, MacGregor JF. Off-line diagnosis of deterministic faults in continuous dynamic multivariable processes using speech recognition methods. *J. Process Control* 1998; **8**: 381–393.
19. Gollmer K, Posten C. Supervision of bioprocesses using a dynamic time warping algorithm. *Control Eng. Pract.* 1996; **4**: 1287–1295.
20. Kassidas A. Fault detection and diagnosis in dynamic multivariable chemical processes using speech recognition methods. Ph.D., Mc Master University, Hamilton, Ontario, Canada, 1997.
21. Aach J, Church GM. Aligning gene expression time series with time warping algorithms. *Bioinformatics* 2001; **17**: 495–508.
22. Kruskal JB, Liberman M. *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison* (Reissue). Center for the Study of Language and Information Publications, Leland Stanford University, Stanford, CA 1999; 125–161.
23. Wulfert F, Kok WT, Smilde AK. Influence of temperature on vibrational spectra and consequences for the predictive ability of multivariate models. *Anal. Chem.* 1998; **70**: 1761–1767.
24. Box GEP, Jenkins GM, Reinsel GC. *Time Series Analysis, Forecasting and Control* (3rd edn). Prentice-Hall: Upper Saddle River, NJ, 1994.

Paper V

Chemical fingerprinting of petroleum biomarkers using time warping and PCA

J. H. Christensen, G. Tomasi, A. B. Hansen

Environmental Science and Technology, 39, 2005, 255 - 260.

Chemical Fingerprinting of Petroleum Biomarkers Using Time Warping and PCA

JAN H. CHRISTENSEN,^{*,†,‡}
GIORGIO TOMASI,[#] AND
ASGER B. HANSEN[‡]

*Department of Environmental Chemistry and Microbiology,
National Environmental Research Institute,
Frederiksborgvej 399, P.O. Box 358, 4000 Roskilde, Denmark,
Department of Life Sciences and Chemistry, Roskilde
University, Universitetsvej 1, P.O. Box 260,
4000 Roskilde, Denmark, and Department of Food Science,
The Royal Veterinary & Agricultural University,
Frederiksberg C, Denmark*

A new method for chemical fingerprinting of petroleum biomarkers is described. The method consists of GC-MS analysis, preprocessing of GC-MS chromatograms, and principal component analysis (PCA) of selected regions. The preprocessing consists of baseline removal by derivatization, normalization, and alignment using correlation optimized warping. The method was applied to chromatograms of *m/z* 217 (tricyclic and tetracyclic steranes) of oil spill samples and source oils. Oil spill samples collected from the coastal environment in the weeks after the Baltic Carrier oil spill were clustered in principal components 1 to 4 with oil samples from the tank of the Baltic Carrier (source oil). The discriminative power of PCA was enhanced by deselecting the most uncertain variables or scaling them according to their uncertainty, using a weighted least squares criterion. The four principal components were interpreted as follows: boiling point range (PC1), clay content (PC2), carbon number distribution of sterols in the source rock (PC3), and thermal maturity of the oil (PC4). In summary, the method allows for analyses of chromatograms using a fast and objective procedure and with more comprehensive data usage compared to other fingerprinting methods.

Introduction

Chemical fingerprinting is a collection of techniques that trace the origin of a sample (e.g. pollutant) based on its chemical composition. In forensic oil spill identification and in geochemistry, petroleum biomarkers are widely used for this purpose (1–3). Oil contains a large number of biomarkers, of which terpanes and steranes are among the most abundant in crude oils. The relative content of biomarker compounds in source rocks, and hence crude oils, depends on source, maturation, and in-reservoir weathering and biodegradation processes (2). Furthermore, these compounds are recalcitrant when released to the environment following oil spills. Thus, they are useful for oil/oil and oil/source rock correlation purposes (4–6).

* Corresponding author phone: +45-46301200; fax: +45-46301114; e-mail: jch@dmu.dk.

† National Environmental Research Institute.

‡ Roskilde University.

The Royal Veterinary & Agricultural University.

Gas chromatography–mass spectrometry (GC-MS) is the standard method for the analysis of petroleum biomarkers (1–3). The associated chromatograms contain a considerable amount of information relevant to chemical fingerprinting but can be complex with peaks that coelute. Consequently, standard peak quantification procedures are associated with large variability and often fail to extract high quality data. Peak separation can be improved by using longer capillary columns or more sophisticated mass spectrometry methods; e.g. high-resolution GC-MS and GC-MS-MS have been found particularly useful for improved resolution and identification of biomarker compounds (7). However, such instrumentation is not widespread in the scientific community, and it is cumbersome to identify and quantify large numbers of peaks as a means to compare oil spill samples and source oils. Consequently, some chemical information is typically ignored, and chemical fingerprinting focuses on few descriptive variables, e.g. diagnostic ratios (3, 4, 8). Moreover, chromatographic data preprocessing, which includes peak identification, quantification, and quality control, is time-consuming and often requires subjective decisions.

Chemometric methods such as principal component analysis (PCA) provide useful tools for more extensive analyses of chromatographic data (5, 9, 10). However, when applied to quantitative data these methods are still affected by the implications described in the previous paragraph. Thus, our primary aim was to develop an objective method for chemical fingerprinting by avoiding initial peak identification, and quantification, and instead performing PCA on the digitized chromatograms. The most severe impediment to such an approach is the inevitable retention time shift caused largely by deterioration of the capillary column (11). The correlation optimized warping algorithm (COW) (12) has been successfully employed to realign chromatograms from GC-FID (13), HPLC (12), and LC-MS (14); here, it is combined with PCA into a method for chemical fingerprinting of petroleum biomarkers. PCA allows for chemical interpretation of the results, which is needed for confirming the observed correlation of oil samples.

The method consists of two parts: preprocessing and chemometric data analysis. Preprocessing comprises derivatization, normalization, and alignment (which includes selection of a target chromatogram, optimization of the warping parameters, and warping of the sample chromatograms). In the chemometric analysis, the data are first divided into a calibration set of source oils, a set of reference oils (the 'reference set'), and a test set containing spill samples; then, a principal component model is fitted to the calibration set and optimized on the basis of the reference set; finally the test set is projected on the model and the spill samples are matched to the source oils.

The method was applied to 101 chromatograms of *m/z* 217, which includes tricyclic and tetracyclic steranes (15) and other compounds yet unidentified (Figure 1). Tetracyclic steranes have been used frequently for chemical fingerprinting (4, 5), but many peaks coelute and hence only a fraction of these is commonly employed for forensic oil spill identification (3, 4, 8).

Methods and Materials

Experimental. The oil samples used in the analysis were all part of the oil database at the forensic oil spill laboratory, National Environmental Research Institute, DK. The database consists of crude oils, refined products, oil mixtures, and spill samples from oil spill cases during the last 10 years. A subgroup of these oils was dissolved separately in dichlo-

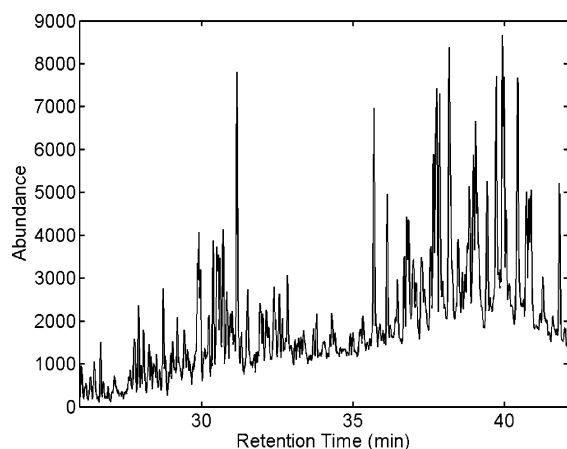


FIGURE 1. Chromatographic profile of m/z 217. Among others, this profile contains tricyclic steranes (eluting between 26 and 34 min) and tetracyclic steranes (eluting between 35 and 42 min).

romethane to a total oil concentration of approximately 2000 mg/L and stored at -20°C in airtight vials, prior to chemical analysis. Oil samples were analyzed on a HP-6890 GC interfaced to a HP-5973 quadrupole mass spectrometer (Agilent technologies) using electron ionization. Chromatographic conditions: $1\ \mu\text{L}$ aliquots were injected in splitless mode; carrier gas helium ($1.0\ \text{mL/min}$); inlet temperature 300°C ; temperature program: 35°C (1 min), 20°C/min to 150°C , 6°C/min to 315°C (15 min). Mass spectrometric conditions: transfer line and ion source temperature: 300°C and 280°C , respectively. Selected ion monitoring was used to analyze 48 mass fragments, divided into 6 groups containing 14–15 ions each as part of a larger fingerprinting study (10). Hence, a low sampling rate of 1.27 scans/s was obtained. In the analytical sequence, a blank and an in-laboratory reference oil were analyzed between every five oil samples. The reference oil was a 1:1 mixture of Brent crude oil (North Sea crude oil) and bunker oil from the Baltic Carrier. The reference samples were used to optimize the warping parameters and to calculate the relative analytical standard deviation used in the variable selection and in the weighted least squares approach.

Data. The 101 oil samples used in this study consisted of 16 weathered spill samples from the Baltic Carrier oil spill (March 29th, 2001, Denmark (16)), 51 source oils, including a sample from the tank of the Baltic Carrier (source oil), 18 replicate reference samples, 2 unrelated spill samples (analyzed in triplicate) from the Round-Robin oil spill exercise (17), and 10 replicate analyses of selected source oils, including 2 replicates of the Baltic Carrier source oil. The chromatograms comprised 2510 data points which after warping were reduced to 1231 (i.e. a retention window from 26 to 42 min) by omitting the parts without chemical information. The data matrix was divided into a calibration set of 61 chromatograms from 51 source oils and 10 replicates (61×1231), a reference set containing the 18 replicate reference samples (18×1231), and a test set comprised of 16 Baltic Carrier oil spill samples and 2 Round-Robin spill samples analyzed in triplicate (22×1231).

Preprocessing of Data. Preprocessing of the chromatographic data prior to PCA is necessary to remove variation unrelated to the chemical composition: time shifts, baselines, concentration effects, and sensitivity changes related to, for example, fragmentation in the ion source and mass selective detector. Three preprocessing methods are outlined in the following: derivatization, normalization, and alignment.

Baseline Removal by Derivatization. The baseline in Figure 1 is caused by features unrelated to the chemical composition and by coelution of compounds with a mass fragment of

217. It can negatively affect both warping (12, 13) and normalization and should be removed. Derivatization is one of several methods that can be employed, and it is often used in combination with smoothing (e.g. Savitsky-Golay), since it amplifies noise (18). To avoid unnecessary alterations of the original data, beside those introduced by the time warping, we decided to use the first derivative calculated numerically as the difference of consecutive points and not to smooth the derivatized chromatograms. Note that polynomial- or piecewise-linear baseline fits are inapplicable in this case, because many peaks are not baseline separated. For the same reasons, it is not viable to manually select points for the baseline fit, since this would increase the human intervention in the preprocessing.

Normalization. Chemical fingerprinting of biomarkers considers the relative distribution rather than concentrations of the single constituents, because these are typically unrelated to the oil source. The concentration information is affected by several parameters, including sampling, extraction, and cleanup procedures, all of which introduce uncertainty and variability in the data. Since the variation associated with total oil concentration is likely to mask the compositional information in a PCA, each chromatogram was normalized using eq 1

$$x_{nj}^N = x_{nj} \left(\sum_{j=1}^J x_{nj}^2 \right)^{-0.5} \quad (1)$$

where x_{nj} is the first derivative of the n th chromatogram at the j th retention time, J is the total number of retention times, and x_{nj}^N is the normalized data. The normalization also adjusts for sensitivity changes that are normally corrected by the use of internal or external standards (e.g. 5β -androstane). Note that the use of standards for normalization would be insufficient, because concentration effects are retained in the data. Furthermore, if normalization were applied before derivatization, the chromatograms would be scaled according to a value that is also unrelated to the chemical composition, thereby influencing the PCA in unpredictable ways. Conversely, normalization according to eq 1 may be affected by closure effects (19), in which case more complex normalization schemes (e.g. using only a limited set of retention times referring to specific peaks) should be adopted. However, this was unnecessary for this data set, and the possibility was not investigated.

Chromatographic Alignment by COW. Compared to other warping methods such as unconstrained dynamic time warping, COW is less flexible. This rigidity gives fewer artifacts and improves the quality of the alignment when applied to complex chromatographic data (13). With the COW procedure, a target chromatogram is selected and divided into segments, and then the optimal boundary positions for corresponding segments are determined separately for each of the remaining chromatograms (sample chromatograms). All combinations of segment boundaries are evaluated, and the one that maximizes the sum of the correlations between corresponding segments in a sample and the target chromatogram gives the optimal chromatographic alignment (12). When the segment length in a sample and the target chromatogram is different, the former is linearly interpolated to the same number of points as the latter. Consequently, the chromatograms are aligned along the time axis by local compression or expansion.

The number of data points each boundary is allowed to move is determined by the so-called slack parameter. Slack parameter, number of segments, and end point constraints (the position of the first and last point is identical in all chromatograms) determine the maximum local correction allowed. This is largest in the central part of a sample

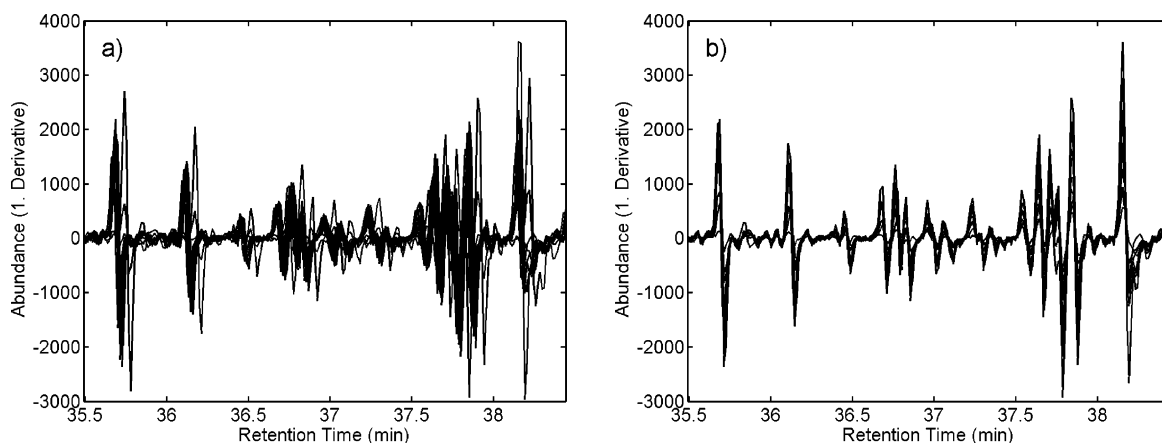


FIGURE 2. First derivative of a section of m/z 217 for five references and five source oils: (a) before warping and (b) after warping using COW with segment length of 175 data points and a slack of 3 points.

chromatogram, which contrasts experimental observations where the shift typically increases with retention time (14). The COW algorithm can be modified to account for this. However, this was not necessary, because the first and last parts of the chromatograms (about 250 and 1000 points) contained only noise and allowed for adequate corrections in the section retained for the PCA (12).

COW is robust with respect to the choice of segment length and slack (13), but some parameter optimization is necessary. The 18 replicate reference samples provided the means for optimization. Disregarding noise and without mean centering, the rank of a matrix consisting of perfectly aligned reference chromatograms (i.e. the number of nonzero singular values) is one. However, since noise is always present in experimental data, the optimal choice of warping parameters is the one that maximizes the first singular value (20).

Chemometric Data Analysis. The preprocessed data were analyzed by PCA, which is particularly useful when variables are highly correlated, as is the case for neighboring retention times in chromatography. The variation in the data matrix \mathbf{X} ($I \times J$) is described by the outer product of the scores matrix \mathbf{T} ($I \times K$) and the loading matrix \mathbf{P} ($J \times K$), where \mathbf{T} is column-wise orthogonal, \mathbf{P} is column-wise orthonormal, and the data matrix \mathbf{X} is typically column-wise centered. In eq 2 \mathbf{E} is the residual matrix.

$$\mathbf{X} = \mathbf{TP}^T + \mathbf{E} \quad (2)$$

The principal components (PCs) are linear combinations of the original variables and are ordered according to their explained variances. Consequently, PC1 represents the most prominent source of variation in the data, PC2 the second, and so forth. The number of components K expresses the complexity of the data and can be regarded as the number of independent underlying phenomena (20). There are several methods to establish the correct K , e.g. the broken-stick rule or cross-validation (20). However, a thorough analysis of scores and loadings (21) was particularly useful here.

Although preprocessing removes most of the variation unrelated to the chemical composition, the chromatograms still contain noise (see Figure 2), which contributes to the variance and thus affects the principal components. To reduce this influence and improve the ability of PCA to distinguish dissimilar oil samples (i.e. its discriminative power) two strategies were tested: variable selection and application of weighted least squares for fitting the PCA model (22).

In the first approach, an increasing number of variables was excluded from the model starting with the one with the largest uncertainty (relative standard deviation) and removing

one at a time until exhaustion. The relative standard deviation of each retention time was determined from the 18 replicate references (RSD_{ref}). The optimal number of retention times was established by taking advantage of the presence of replicate oil samples in the data set. The optimization was done by minimizing the variance of replicates for each oil sample s with respect to their average scores \bar{t}_s (eq 3a), compared to the variance explained by the model (eq 3b (20)), in the K -dimensional space spanned by the principal components

$$d_{Rep} = \sum_s (n_s - 1)^{-1} \sum_{i \in S_s} (\mathbf{t}_i - \bar{\mathbf{t}}_s)(\mathbf{t}_i - \bar{\mathbf{t}}_s)^T \quad (3a)$$

$$d_{All} = \sum_{i=1}^I \mathbf{t}_i \mathbf{t}_i^T \quad (3b)$$

$$r = \frac{d_{Rep}}{d_{All}} \quad (3c)$$

where S_s are the row indexes for the n_s replicates of oil sample s , \mathbf{t}_i is the i th row of \mathbf{T} , and r is dimensionless. If additional retention times describe noise rather than systematic variation, the variances of the replicates d_{Rep} are expected to increase because of overfitting. Thus, PCA with the highest discriminative power minimizes r (eq 3c), where d_{All} is the normalization factor that accounts for the increase in d_{Rep} due to the larger number of variables included in the model. Note that r depends on K , which depends on the variables included in the PCA. Three values of r were monitored: r_{Cal} , using the 10 replicate oil samples included in the calibration set, r_{Ref} for the reference set, and r_{Test} using the oil samples within the test set (i.e. spill samples). In r_{Ref} and r_{Test} , the scores were obtained by projection on the space spanned by \mathbf{P} .

In the second approach, the subjectivity was reduced by fitting the PCA according to a weighted least squares criterion (23), thus retaining all the variables in the data. We used the MILES-PCA algorithm (Maximum likelihood via Iterative Least squares EStimation) (23) for this purpose, using the inverse of RSD_{ref} as weights.

All data analyses were performed in MATLAB 6.5 (The MathWorks). The algorithms for COW and MILES-PCA can be downloaded from www.models.kvl.dk.

Results and Discussion

Retention Time Alignment. Combinations of segment lengths from 25 to 225 data points, with increments of 25 and slacks between 1 and 4 were tested to find the optimal

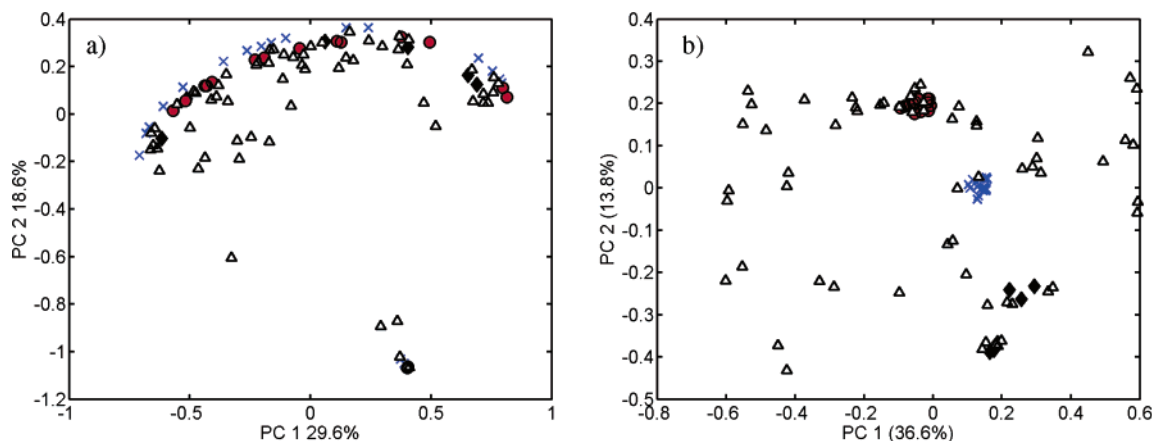


FIGURE 3. PCA score plots of PC1 vs PC2: (a) without alignment and (b) with alignment. Baltic Carrier oil spill samples ((red) ●), replicate references ((blue) ×), triplicate Round-Robin oil spill samples (◆), and oil samples in the calibration set (△).

warping parameters. A reference sample analyzed halfway through the analytical sequence was selected as the target chromatogram to reduce the need for correction. The explained variance of a one-component model, which is proportional to the first singular value, was calculated for the uncentered reference set. It increased from 59.1% (without warping) to above 94% for all tested segment lengths and slack of 2 to 4.

The maximum explained variance (96.2%) was attained for segments of 175 data points and a slack of 3. Figure 2 illustrates the effect of time warping on a section of the chromatograms (35.5–38.5 min). The improvement is evident, and the residual misalignment after warping is at most one point when considering the whole data set (101×1231). It has been suggested (12, 13) that optimal chromatographic alignment is achieved if the segment length is of the same order of magnitude as the peaks. Although for this data set the peak width was around 10 points, short segments (25 points or less) yielded poorer alignments. This appears to be caused by the low sampling rate and by the interpolation step inherent to the COW algorithm: the peaks may be shifted by a fraction of a point that depends on the ratio between slack and segment length. Thus longer segments, provided that enough flexibility is allowed for the bulk of the shift, may produce finer corrections that may improve the warping results. Increasing the sampling rate, e.g. by focusing on fewer masses in the GC-MS analysis, would allow for more refined corrections.

Chemometric Data Analyses. PCA was applied to the mean-centered calibration set with and without warping. Figure 3a shows the score plot of PC1 vs PC2 without warping. The 16 Baltic Carrier oil spill samples and the 18 replicate references form a pattern typical of situations where the retention time shift is the main cause of systematic variation. Hence, PCA describes the misalignment rather than the chemical composition of the oil samples. Figure 3b shows the score plot of PC1 vs PC2 after warping. The improvement in the discriminative power of the PCA is evident: both replicate references and Baltic Carrier oil spill samples are clustered, and the triplicate Baltic Carrier source oils fall within the cluster of the spill samples.

The explained variance of a four-component model before and after warping was almost constant (67.8% and 67.5%, respectively), while the amount of variation described by the individual components changed considerably (e.g. 29.6% to 36.6% for PC1). This was caused by the removal of the retention time shifts as a source of variation and demonstrates that the shift is confounded, not only with the first but also with subsequent components, which prevents any sensible interpretation of the PCA without alignment (19).

The clustering of references and of the Baltic Carrier spill samples holds for the first four components, whereas subsequent PCs to some extent describe noise and residual misalignment. This was confirmed by the loadings, where some peaks in PC5 and subsequent components are systematically shifted one point when compared to the mean of the 101 chromatograms, and from r_{Cal} and r_{Ref} , which increased sharply when more than five components were included in the model. Thus, the chemical information in additional components is confounded with the residual misalignment enhanced by the low sampling rate, and an upper limit to the number of significant components was observed. For the same reason, methods for establishing K such as cross-validation would not suggest the correct number of components, since the prediction error in validation would continue to decrease without a sharp bend.

The method was modified by removing uncertain variables in order to enhance the discriminative power of the PCA. The number of retention times included in the PCA was decreased from 1231 to 1 according to their RSD_{Ref} . r_{Cal} and r_{Ref} fluctuated when using few retention times, and a distinct minimum was attained at 838 variables for r_{Cal} and 351 variables for r_{Ref} . Retaining more than 838 variables led to a steady increase in both ratios, which suggested that additional variables did not contain useful information. Three models using 1231, 838, and 351 retention times were applied to the test set, which yielded r_{Test} values of 2.0×10^{-2} , 1.8×10^{-2} , and 1.3×10^{-2} . Likewise, the lowest distances between source oils and the corresponding spill samples were yielded using 351 retention times, which indicates that 838 variables were too many and the model was overfitting.

Nevertheless, the selection of variables is clearly not trivial, and it contradicts the aim of minimizing subjectivity in the data analysis. Fitting the PCA model according to a weighted least squares criterion represents an objective alternative. Figure 4 shows the score plot of PC1 vs PC2 using MILES-PCA, with four components and mean-centering (22). The value of r_{Test} is 1.6×10^{-2} which is slightly higher than for the optimal variable selection.

The results are encouraging since the variances of replicate oil samples are consistently reduced when using variable selection or a weighted least squares criterion compared to standard PCA. However, this information alone is not sufficient to determine which method is best for correlating oil spill samples and source oils. Further studies considering weathering processes and statistical testing will be necessary to establish whether these reductions are significant and to obtain some general guidelines. However, despite weathering processes (e.g. evaporation and water washing) for up to 14 days, the Baltic Carrier oil spill samples and the corresponding

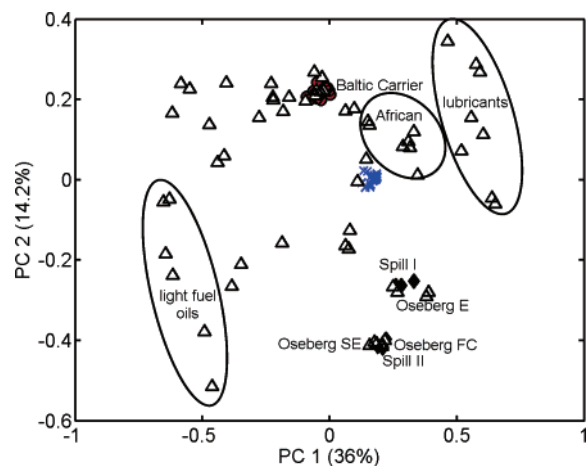


FIGURE 4. Weighted least squares PCA score plot of PC1 vs PC2. Baltic Carrier oil spill samples ((red) ●), replicate references ((blue) ×), triplicate Round-Robin oil spill samples (◆), and oil samples in the calibration set (△). The three replicate source oils from the tank of the Baltic Carrier fall within the cluster of spill samples.

source oil are clustered in PC1 through 4. Likewise, the Round-Robin spill samples, Spill I and Spill II, are grouped in the plot with the corresponding sources, Oseberg East (E) and Oseberg Field Centre (FC) (17). Note that, although Oseberg South East (SE) lies close to Oseberg E along PC1 and PC2, they are well separated along PC3 and PC4 (not shown).

Spill samples from real spill cases consist sometimes of a mixture of source oils, which considerably hinders the source correlation process. In this respect, it would be advantageous if a fingerprinting technique could also identify the constituent source oils of such mixtures. As previously stated, the reference oil is a 1:1 mixture of Baltic Carrier oil and a North Sea crude (Brent, U.K.). Brent crude was not analyzed in our study, but several other North Sea crudes were, e.g. the Oseberg source oils. Correspondingly, the replicate reference samples lie almost on a line connecting the clusters of Baltic Carrier samples and North Sea Crude oils in the four-dimensional space spanned by the retained principal components. Most of the variation between the two oil types is described by PC2, whereas they form a single cluster in PC3 and PC4.

Hence, we expect that another mixture of the same oils but in different proportions would lie on the same line. Likewise, if three sources were considered, all possible mixtures would lie on a plane and so forth.

Petroleum biomarkers provide detailed information on source, depositional environment, thermal maturity (2), and refining process. Numerous ratios of biomarker compounds have been used for forensic oil spill identification and oil exploration in geochemistry (2, 4, 8, 24). Consequently, chemical interpretation of the results facilitates the correlation between source oils and spill samples. Retention times that contribute the most to a PC are associated with large negative or positive coefficients in the corresponding loading.

Figure 5 shows part of the integral (cumulative sum) of the loading of PC2 and of the mean chromatogram of the 101 oil samples. The loading coefficients for diasteranes (DAS) are negative, whereas they are positive for the rearranged steranes (RS). Ratios of the type DAS/RS are commonly used to distinguish oil originating from source rocks with different clay content (2). Low DAS/RS indicate anoxic clay-poor, carbonate source rock, whereas high DAS/RS indicate source rocks containing abundant clays. Hence, PC2 can be interpreted as a source parameter where oil samples with positive PC2 (e.g. the Baltic Carrier oil) are derived from a source rock containing less clay than oil samples with negative PC2

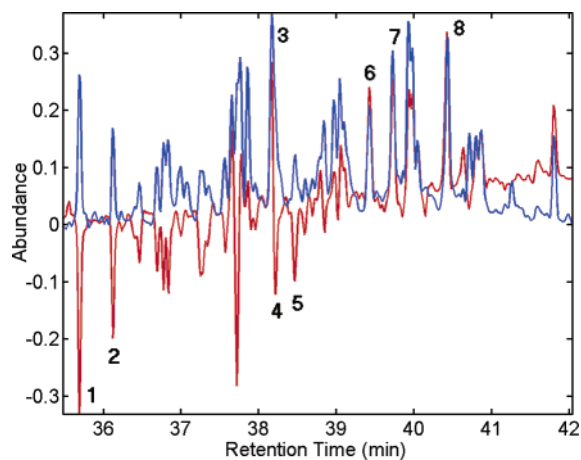


FIGURE 5. Integrated mean-chromatogram (blue) and integrated loadings of PC2 (red) for weighted least squares PCA. Compounds are tentatively identified from refs 2, 8, and 15. (1) 13 β ,17 α ,20S-cholestane (DAS), (2) 13 β ,17 α ,20R-cholestane (DAS), (3) 5 α ,14 α ,17 α ,20R-cholestane (RS), (4) 24-ethyl-13 β ,17 α ,20R-cholestane (DAS), (5) 24-ethyl-13 α ,17 β ,20S-cholestane (DAS), (6) 24-methyl-5 α ,14 α ,17 α ,20R-cholestane (RS), (7) 24-methyl-5 α ,14 α ,17 α ,20S-cholestane (RS), (8) 24-ethyl-5 α ,14 α ,17 α ,20R-cholestane (RS).

(e.g. North Sea crude oils). Consequently, it appears that the Baltic Carrier oil and North Sea crudes are distinguished by the clay content of their source rocks.

PC1 (not shown) describes the boiling point range, whereas tetracyclic steranes have positive coefficients, most of the lower-boiling-range compounds in the first peak-cluster (26 to 34 min) have negative ones. Correspondingly, lubricants have positive scores for this component, whereas the scores for light fuel oils are negative (see Figure 4). In PC3 (not shown) large positive coefficients of 5 α ,14 α ,17 α ,20R-cholestane (27 $\alpha\alpha$ R) and 24-methyl-5 α ,14 α ,17 α ,20R-cholestane (28 $\alpha\alpha$ R) compared to 24-ethyl-5 α ,14 α ,17 α ,20R-cholestane (29 $\alpha\alpha$ R) suggest that PC3 is a source parameter reflecting carbon number distribution of sterols in the organic matter of the source rock (2). Finally, PC4 (not shown) describes the thermal maturity of oils. The $\beta\beta$ -isomers of C₂₇ to C₂₉-regular steranes have large negative coefficients, whereas the $\alpha\alpha$ -isomers have positive ones. The ratio of C₂₉-regular steranes ($\beta\beta/(\beta\beta+\alpha\alpha)$) is a highly specific parameter for maturity and appears to be independent of source organic matter input (2). The $\beta\beta$ isomers have a higher thermal stability compared to the $\alpha\alpha$ isomers, thus the above ratio increases with thermal maturity. African crude oils (Gabon, Kote-Cameroon, Escravos-Nigeria) have positive scores for this component, whereas the scores for oils from e.g. the Middle East (Iran Light, Iran Heavy, Basrah-Iraq, Dubai-Iraq) and Russia (Romaskino, Ural) are negative. Consequently, the formers appear to have a lower thermal maturity. Analogous conclusions can be drawn from additional maturity parameters e.g. 20S/(20S+20R) C₂₉-regular steranes.

In summary, our method allows for analyses of chromatograms using a fast and highly objective procedure. Once the PCA model is constructed, the complete data analysis of a new oil sample (derivatization, normalization, alignment, and PCA) requires few seconds. Conversely, if PCA were performed on quantitative data, an analogous analysis would be considerably more time-consuming and less objective due to the quantification process. Furthermore, as long as the variation between oils in the calibration set is sufficient, the PCA can distinguish coeluting peaks. The same is far more difficult in standard quantification procedures. For example, peaks 3 and 4 in Figure 5 are highly overlapping in the mean chromatogram, but their loading coefficients have different signs. This is consistent with the interpretation

of PC2 that regular steranes have positive coefficients and diasteranes have negative ones.

Retention time shifts are not the sole effect of instrumental drift and deterioration of the GC column; peak shapes are also affected. When using quantitative data for PCA (relative peak areas or concentrations), these changes have a limited effect, because peak areas remain substantially unaltered. On the contrary, when PCA is applied to chromatograms, changes in peak shape represent a prominent source of variation that cannot be corrected only by warping and cannot be adequately modeled by a bilinear model. External normalization based on reference oils (10) may to some extent correct for this, but further studies are required. Furthermore, changes in peak shapes are more frequent when oil samples are analyzed over a long period of time. Hence, it is even more important to monitor the quality of the chromatographic data for the proposed method than for standard quantitative analysis. In this sense, the aligned reference samples would be the natural choice. However, this would require extensive investigations that are beyond the scope of this work.

Another important issue is the effect of weathering and biodegradation on the chemical composition. We treated short-term weathering processes implicitly by considering recalcitrant compounds such as tri- and tetracyclic steranes, but this solution is not sufficient when oil samples have been exposed to long-term weathering and biodegradation. Nevertheless, oil spill samples with varying degrees of weathering can be used to estimate the relative uncertainties instead of the references. Consequently, compounds affected by these processes will have a large uncertainty and will be deselected or downscaled.

Chemical fingerprinting of biomarkers focuses on tetracyclic steranes and terpanes (4, 8). A separate PCA applied to the former, which elute between 35 and 42 min, had a lower discriminative power than the PCA described thus far, which also comprises tricyclic steranes and several compounds yet unidentified. Hence, with proper preprocessing, the discriminative power of the method increases when more information is included, and the additional variables are unaffected by weathering and biodegradation. Likewise, we expect that the addition of compositional information from other groups of petroleum hydrocarbons would improve the results. This can be accomplished by combining chromatograms after alignment into a single data matrix.

Acknowledgments

The authors acknowledge Parmely Pritchard and Rasmus Bro for fruitful comments on the manuscript, Jørgen Avniskjold for technical assistance, and Gilda Kischinovsky for proofreading. This work was financed partly by the Department of Environmental Chemistry and Microbiology, National Environmental Research Institute, and partly by the Natural Science Research Council, Denmark.

Literature Cited

- (1) Barakat, A. O.; Mostafa, A. R.; Qian, Y. R.; Kennicutt, M. C. Application of petroleum hydrocarbon chemical fingerprinting in oil spill investigations – Gulf of Suez, Egypt. *Spill Sci. Technol. Bull.* **2002**, *7*, 229–239.
- (2) Peters, K. E.; Moldowan, J. M. *The Biomarker Guide: Interpreting Molecular Fossils in Petroleum and Ancient Sediments*; Prentice Hall: Englewood Cliffs, New Jersey, 1993.
- (3) Wang, Z. D.; Fingas, M.; Sergy, G. Study of 22-year-oil Arrow oil samples using biomarker compounds by GC/MS. *Environ. Sci. Technol.* **1994**, *28*, 1733–1746.
- (4) Barakat, A. O.; Mostafa, A. R.; Rullkotter, J.; Hegazi, A. R. Application of a multimolecular marker approach to fingerprint petroleum pollution in the marine environment. *Mar. Pollut. Bull.* **1999**, *38*, 535–544.
- (5) Stout, S. A.; Uhler, A. D.; McCarthy, K. J. A strategy and methodology for defensibly correlating spilled oil to source candidates. *Environ. Forensics* **2001**, *2*, 87–98.
- (6) Wang, Z. D.; Fingas, M.; Sergy, G. Chemical characterization of crude-oil residues from an Arctic beach by GC/MS and GC/FID. *Environ. Sci. Technol.* **1995**, *29*, 2622–2631.
- (7) Munoz, D.; Doumenq, P.; Guiliano, M.; Jacquot, F.; Scherrer, P.; Mille, G. New approach to study of spilled crude oils using high-resolution GC-MS (SIM) and metastable reaction monitoring GC-MS-MS. *Talanta* **1997**, *45*, 1–12.
- (8) Daling, P. S.; Faksness, L. G.; Hansen, A. B.; Stout, S. A. Improved and standardized methodology for oil spill fingerprinting. *Environ. Forensics* **2002**, *3*, 263–278.
- (9) Mudge, S. M. Reassessment of the hydrocarbons in Prince William Sound and the Gulf of Alaska: Identifying the source using partial least-squares. *Environ. Sci. Technol.* **2002**, *36*, 2354–2360.
- (10) Christensen, J. H.; Hansen, A. B.; Mortensen, J.; Tomasi, G.; Andersen, O. A new integrated methodology for forensic oil spill identification. *Environ. Sci. Technol.* **2004**, *38*, 2912–2918.
- (11) Fraga, C. G.; Prazen, B. J.; Synovec, R. E. Comprehensive two-dimensional gas chromatography and chemometrics for the high-speed quantitative analysis of aromatic isomers in a jet fuel using the standard addition method and an objective retention time alignment algorithm. *Anal. Chem.* **2000**, *72*, 4154–4162.
- (12) Nielsen, N. P. V.; Carstensen, J. M.; Smedsgaard, J. Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping. *J. Chromatogr. A* **1998**, *805*, 17–35.
- (13) Tomasi, G.; van den Berg, F.; Andersson, C. Correlation Optimized Warping and Dynamic Time Warping as preprocessing methods for chromatographic data. *J. Chemometr.* **2004**, *18*, 231–241.
- (14) Bylund, D.; Danielsson, R.; Malmquist, G.; Markides, K. E. Chromatographic alignment by warping and dynamic programming as a preprocessing tool for PARAFAC modelling of liquid chromatography–mass spectrometry data. *J. Chromatogr. A* **5–7–2002**, *961*, 237–244.
- (15) Moldowan, J. M.; Albrecht, P.; Philp, R. P. *Biological markers in sediments and petroleum*; Prentice Hall: Englewood Cliffs, New Jersey, 1992.
- (16) Christensen, J. H. Application of multivariate data analysis for assessing the early fate of petrogenic compounds in the marine environment following the Baltic Carrier oil spill. *Polycyclic Aromat. Compd.* **2002**, *22*, 703–714.
- (17) Faksness, L. G.; Daling, P. S.; Hansen, A. B. Round Robin study – Oil spill identification. *Environ. Forensics* **2002**, *3*, 279–291.
- (18) Candolfi, A.; De Maesschalck, R.; Jouan-Rimbaud, D.; Haily, P. A.; Massart, D. L. The influence of data-preprocessing in the pattern recognition of excipients near-infrared spectra. *J. Pharm. Biomed. Anal.* **1999**, *21*, 115–132.
- (19) Malmquist, G.; Danielsson, R. Alignment of chromatographic profiles for principal component analysis – A prerequisite for fingerprinting methods. *J. Chromatogr. A* **1994**, *687*, 71–88.
- (20) Jolliffe, I. T. *Principal component analysis*; Springer-Verlag: 1986.
- (21) Martens, H.; Næs, T. *Multivariate calibration*; John Wiley & Sons: Chichester, U.K., 1996.
- (22) Bro, R.; Sidiropoulos, N. D.; Smilde, A. K. Maximum likelihood fitting using ordinary least squares algorithms. *J. Chemometr.* **2002**, *16*, 387–400.
- (23) Bro, R.; Smilde, A. K. Centering and scaling in component analysis. *J. Chemometr.* **2003**, *17*, 16–33.
- (24) Farrimond, P.; Taylor, A.; Telnaes, N. Biomarker maturity parameters: the role of generation and thermal degradation. *Org. Geochem.* **1998**, *29*, 1181–1197.

Received for review February 2, 2004. Revised manuscript received September 28, 2004. Accepted September 29, 2004.

ES049832D

Paper VI

Signal preprocessing and modeling in
chromatography

I. García, R. Bro, G. Tomasi, L. Sarabia, M.C. Ortiz

In preparation

Signal preprocessing and modelling in chromatography.

1. Introduction

This paper is concerned with data preprocessing in chromatography that can help reduce the influence of baseline variations and retention time shifts when modelling chromatographic data with the so-called multi-way methods. Some multi-way methods are capable of separating, *mathematically*, any overlapping peaks, but require that the influence of the aforementioned factors is minimized as much as possible.

High performance liquid chromatography with diode array detection (HPLC-DAD) is frequently used for the quantitative analysis and the identification of organic residues or contaminants in live animals and animal products destined for human consumption. This technique has been considered to be suitable [1] for detecting contaminants classified in group B in Annex I of Directive 96/23/EC.

HPLC-DAD is a hyphenated instrument, i.e. two first-order instruments, a chromatograph and a spectrophotometer, working in tandem. This means that a full ultraviolet spectrum is recorded for each (discrete) elution time of the chromatographic profile and a data matrix is obtained for each sample; that is, the absorbance is recorded as a function of the wavelength and the elution time. Other detectors allowing three-dimensional data with liquid chromatography are fast-scanning fluorescence (FSFS) [2] and mass spectrometry (MS) [3,4]. Data matrices of different samples can be arranged in a tensor, also called multi-way array, and analysed by three-way models such as parallel factor analysis (PARAFAC) [5,6].

PARAFAC ideally decomposes multi-way data [10] in such a way that, it uniquely resolves the true underlying profiles, i.e. instrumental and concentration loadings. Because it retains this property for estimating profiles of pure analytes even in presence of uncalibrated interferences, PARAFAC is a useful tool to comply with the requirements established in the European Decision 2002/657/EC [1]. PARAFAC allows one not only the quantification but also the identification and confirmation of analytes in biological samples when several matrix interferences coelute with the analyte of interest.

One of the basic assumptions of PARAFAC is that data must be low-rank trilinear [10]. As it is shown in the present paper, biased results might be obtained if the model is applied to a data set which deviates from this trilinearity.

Some phenomena, such as baseline variations, different baselines between samples, constant offsets and variation in the retention time from run to run, are common

artefacts in chromatography and might alter [11] the trilinear structure of the data. Depending on the purpose of the analysis, namely qualitative or quantitative study, the effects of the lack of trilinearity on the results are different and several kinds of corrections have been presented in ref. [11]. Most of them are based on the selection of the model to fit the data or on the experimental design to get the three-way data. However, once data have been recorded, some of the shortcomings might be corrected by some kind of data preprocessing or by selecting an appropriate three-way model.

The aim of this paper is to show the effectiveness of different tools to minimize the effect of the baseline (section 1.1) and of the variations in the retention time (section 1.2) on the estimations of the three-way models when sulfonamides are extracted from kidney samples. We will focus on the quantitative results and on the estimations of the chromatographic and spectral profiles of the analyte of interest.

1.1. Baseline variations.

The effect of the variations in the baseline can be minimized by data pretreatment which leads to simpler models with fewer components. In this paper, we will focus on simple approaches to centre signals, first and second-order derivatives of the chromatograms and baseline subtraction. These pretreatment techniques have been mostly applied in near infrared (NIR) spectroscopy to reduce spectral variations but few references have been found for the preprocessing of second-order data from HPLC-DAD modelled by three-way models. For example, ref. [12] deals with the effect of the preprocessing method (multiplicative signal correction, first and second-order derivatives) on a multivariate figure of merit, the sensitivity.

In a three-way tensor [13], centering across one mode involves subtracting the column mean from each element. This procedure does not change distances between points but the origin of the space. Hence, centering is mainly used to remove constant terms in one of the modes of the data so that there will be no need to include extra components in the model to explain the offset. This technique is frequently applied in batch processes [14] to remove the major but common, and hence uninteresting, time trajectory of the process variables. Thus, subsequent models only need to focus on the minor but relevant deviations from the average trajectory. In this paper, centering across the sample mode has been performed to remove common effects between samples.

Differentiation has been widely applied in chromatography for finding the retention time of the peak (maximum) as well as to determine the initial and final points for signal integration. From the identification point of view, signal differentiation may be appealing to enhance the resolution between two or more analytes which elute at close retention times. In ref. [15], the first-derivative chromatograms have been modelled by bilinear partial least squares (PLS) to determine pesticides in groundwater and soil samples. Complex bands have been partially resolved so that the algorithm can easily converge and the results are more reproducible. The differentiation of chromatograms removes backgrounds and differences in the baseline between samples that may interfere with the quantitative information present in the chromatogram. From the quantitative point of view, the results reported in ref. [16] with respect to differentiation are more precise than those with raw signals because the background of the signals of those samples with different nature to the training set is removed. Regarding multi-way analysis, the first derivative with the Savitzky-Golay transformation has been used for smoothing signals and solving the baseline effects in NIR spectroscopy prior to fitting a PARAFAC model [17]. However, no reference has been found for the analysis of derivative chromatograms with three-way models.

Another option for correcting changes in the baseline is to fit a model for the baseline and then to subtract it from the data matrix. This technique has been applied to correct signals from liquid chromatography [18], gas chromatography [19] and capillary electrophoresis [20].

1.2. Retention time shifts.

Other kinds of artefacts disturbing the trilinearity of the signals are the instrumental shifts in the elution time from sample to sample due to variations in the column during use, fluctuations in the chromatographic conditions (such as temperature, flow rate, injection...) or drift in the chromatograph which makes the analytes elute later from sample to sample. Chromatographic shifts affect the estimations yielded with PARAFAC. They make the model more difficult to interpret or reduce its robustness.

The problem of non-chemical shifts can be solved either by aligning signals or by using other models not affected by the chromatographic shifts as PARAFAC, such as PARAFAC2 [22] and multivariate curve resolution-alternating least squares (MCR-ALS) [22,23]. Several algorithms have been advocated for signal alignment like the dynamic time warping (DTW) [4,24], the correlation optimised warping (COW) [24,25] and the peak matching algorithm [26]. Some authors have also applied the curve resolution with iterative target transformation factor analysis (ITTFA) [27].

Some of these problems are also found to a great extent in batch processes. Like in chromatography, signals for monitoring batch processes change from one batch to another, do not follow exactly the same time trajectory, and in many cases, batches do not have the same dimension. This problem has been handled by using PARAFAC2 [28] or other approaches such as the DTW algorithm [29].

This paper details the application of different data pretreatment to deal with the baseline variations (section 4.1, centering, first and second-order derivative, baseline estimation/subtraction) and the chromatographic shifts (section 4.2, signal alignment by DTW and PARAFAC2).

2. Experimental

2.1. Chemicals and solutions.

Formic acid, phosphoric acid (85%), dichloromethane and petroleum benzin were purchased from Sigma-Aldrich (Madrid, Spain). Methanol was obtained from Merck (Darmstadt, Germany). Sulfadiazine and sulfamethoxypyridazine were acquired from Fluka (Madrid, Spain).

Deionised water was obtained by the Milli-Q Gradient A10 water purification system of Millipore (Bedford, MA, USA).

1 g l⁻¹ standard solutions of sulfadiazine and sulfamethoxypyridazine were individually prepared in methanol. A diluted standard solution (20 mg l⁻¹) containing the two analytes was daily arranged in methanol by dilution of the standard solution. Ten pure standards at concentrations ranged between 1.33 and 10.33 mg l⁻¹ were prepared from the 20 mg l⁻¹ diluted solution.

The solution of formic acid (0.1%, v/v) was prepared by diluting formic acid in deionised water, filtering through 0.45 µm filters and degassing in an ultrasonic bath.

2.2. Pre-treatment and clean-up procedure.

10 g of homogenized kidney samples were enriched with the sulfonamides at concentrations between 40 and 310 µg kg⁻¹ in the samples (or between 1.33 and 10.33 mg l⁻¹ of each sulfonamide in the vial). 25 ml of dichloromethane, 2 ml of deionized water and 35 µl of phosphoric acid were added. Samples were stirred in an ultrasonic bath for 10 minutes and centrifuged at 3,000 rpm for 10 minutes in a High Speed Refrigerated Centrifuge 4239R from ALC (Milan, Italy). The lower organic phase was filtered and preserved. The extraction procedure was repeated with 25 ml of dichloromethane. Extracts from the two extractions were mixed and 25 ml of petroleum benzin were added.

The Sep-Pak® Vac RC silica cartridges from Waters (Bedford, MA, USA) were used without being

previously activated. The extracts were passed across them at a pressure of 5 mm Hg (around 15 minutes) and then the cartridges were air-dried for 5 minutes. Sulfonamides were eluted with 8 ml of methanol, evaporated to dryness under a stream of nitrogen and dissolved with 300 µl of formic acid (0.1%)/methanol (70/30, v/v).

2.3. Instrumental analysis and three-way data.

20 µl were injected into a high performance liquid chromatograph from Waters (Bedford, MA, USA) equipped with a Waters 510 Pump, a Waters 717 Injector and a Waters 996 UV-vis absorbance detector. Full ultraviolet spectra were monitored between 230 and 440 nm. Each standard was injected twice in order to have instrumental replicates and evaluate the stability of the models. The separation was performed under gradient conditions on a XTerra® MSC₁₈ column (4.6×100 mm) from Waters with 5 µm of particle size. The composition of the mobile phase, formic acid (0.1%, v/v)/methanol was programmed as follows: the initial composition of the mobile phase was 5% of methanol which was raised until 40% in 6 minutes, then increased up to 50% in 4 minutes, raised to 60% in 2 minutes, subsequently decreased until 40% in 2 minutes and returned to the initial conditions (5% of methanol) in 1 minute. These conditions were kept for 5 minutes for column equilibration.

The flow rate was fixed at 1.0 ml min⁻¹ and the run time at 20 minutes. The Millenium 32 software from Waters was used for controlling the system.

Data from HPLC-DAD (ultraviolet-visible spectra registered at different elution times) were arranged in two tensors with dimensions 38×66×66 and 36×66×61 for sulfadiazine and sulfamethoxypyridazine respectively. The first dimension refers to the sample mode (number of standards, pure standards and kidney samples), the second to the spectral mode (number of wavelengths registered) and the third to the chromatographic mode (number of elution times). The pure standard containing 6.33 mg l⁻¹ and its instrumental replicate were considered outliers in both data sets and the pure standard with 10.33 mg l⁻¹ was outlier for sulfamethoxypyridazine. The chromatograms from each analyte were independently examined, firstly sulfamethoxypyridazine (from 9.52 to 10.02 minutes) because its signals are specific and then sulfadiazine (from 6.19 to 7.60 minutes) with which several interferences coelute.

2.4. Software

All data pretreatment (centering, derivatization and baseline subtraction) were performed with the PLS Toolbox for MATLAB® 6.5. The DTW algorithm for signal alignment can be downloaded from www.models.kvl.dk. The PARAFAC and the PARAFAC2

[30] models were built with the PLS Toolbox for MATLAB.

3. Data pretreatment.

3.1. Pretreatment focused on the elimination of constant offsets and baseline drift.

3.1.1. Centering.

Centering involves the projection of the data [13,31] on the nullspace of vectors of ones in a particular mode. In this paper centering across the first mode (sample mode) has been performed by subtracting from every element the column-mean:

$$Z_{ijk} = X_{ijk} - \frac{\sum_{i=1}^I X_{ijk}}{I} \quad (1)$$

3.1.2. First and second-order derivatives.

The simplest way to determine the derivative of a signal is to subtract two consecutive data points. Unfortunately, this simple approach increases considerably the noise in the signal. This can be remedied by combining [32] differentiation with a smoothing technique such as that proposed by Savitzky and Golay [33]. Polynomial smoothing consists of selecting a window width with odd number of data points. A polynomial model is fitted to the points in the selected window and the central point of the window is replaced by the value of the first derivative of that polynomial in that point. Thereafter, the window is shifted one data point and the fitting process is repeated with all data points.

Some considerations should be taken into account before applying this smoothing technique. Firstly, large window sizes improve the signal to noise ratio of the signals but introduce some distortion; in particular, peaks become wider [32]. On the other hand, when smoothing is carried out together with derivatization, changes in the retention time with respect to raw signals occur [32]. The greater the order of the derivative, the more significant the shift. This means that additional artefacts are introduced in the data that need to be modelled. In this paper, for estimating the first and the second-order derivative, a third-order polynomial model was adjusted to the chromatograms fixing the window width at 11 time-points.

3.1.3. Baseline correction with linear regression.

The procedure [18] consists of selecting several baseline regions, preferably both after and before component elution. Then, the baseline is fitted to a linear

model whose coefficients, \mathbf{b} , can be estimated by least squares:

$$\mathbf{b} = (\mathbf{D}^t \mathbf{D})^{-1} \mathbf{D}^t \mathbf{x} \quad (2)$$

where \mathbf{D} is a matrix whose first column contains ones and the second column contains the scan time in baseline regions. \mathbf{x} is the vector of absorbances over the specified baseline regions, and \mathbf{b} is a vector of two coefficients, the intercept (offset) and the slope (drift) for each wavelength.

Once the coefficients, b_j , have been calculated, the baseline is estimated for all retention times that is, baseline regions and chromatographic peaks, and subtracted from the original data set. The procedure is repeated for each wavelength, j , because drift and offset depend on the wavelength. Data obtained in this way will be baseline-corrected only if the baseline is a linear function of the scan time.

3.2. Pre-treatment focus on removing chromatographic shifts.

3.2.1. Dynamic time warping (DTW).

The DTW algorithm corrects shifts in the time mode by aligning each chromatogram with a reference chromatogram. Therefore, two signals intervene in each step of the warping procedure: the sample, denoted here as s , with length M , and the reference, r , with length N . The notation and terminology of this paper is the same as that detailed in reference [24] which uses m and n to specify

the index in the sample and the reference respectively.

The alignment of both the reference and the sample signals by means of DTW is based on dynamic programming and can be considered to be an optimization procedure where a linking path between the reference and the sample is built so that their cumulative distance is minimized.

$$\arg \min_F D(F) = \frac{\sum_{k=1}^K d_{rs}[m(k), n(k)] w(k)}{\sum_{k=1}^K w(k)} \quad (3)$$

where $d_{rs}[m(k), n(k)]$ is a dissimilarity measure, the squared euclidean distance between $r[n(k)]$ and $s[m(k)]$. $m(k)$ and $n(k)$ are the indexes for the sample and the reference at the k th point on the warped time axis. K is the length of the warping path and $w(k)$ are suitable weights [24]. But for the weights, the minimization procedure would be biased towards shorter paths, which entails the summation of fewer terms. The optimal warping path is consequently independent of its length, K . The warping path is defined as follows:

$$F = \{[m(k), n(k)] \mid k = 1, \dots, K\}$$

The warping path, F , is a sequence of elementary transitions. A transition is defined as a single advancement in the common time axis and a sequence of elementary transitions constitutes a rule which describes a possible warping path at a local level.

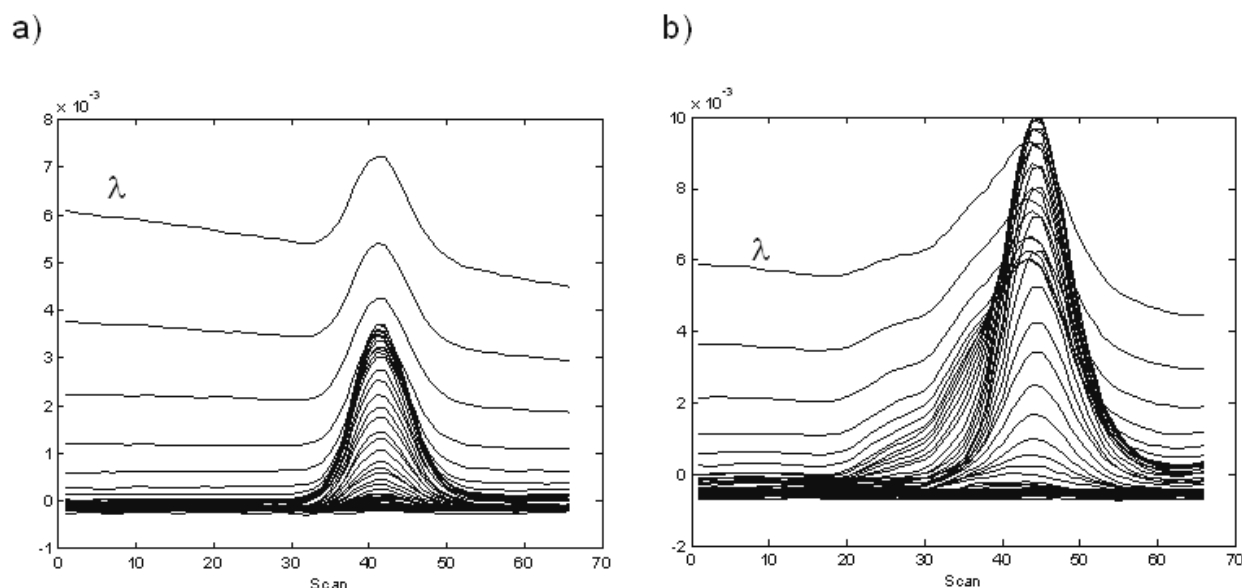


Figure 1. Chromatograms registered at different wavelengths between 242 and 395 nm (a) of a pure standard containing 1.33 mg l⁻¹ of sulfadiazine and (b) of a kidney sample enriched at 40 µg kg⁻¹ (1.33 mg l⁻¹ in vial). Scan 1 and scan 66 are 6.19 and 7.60 minutes respectively.

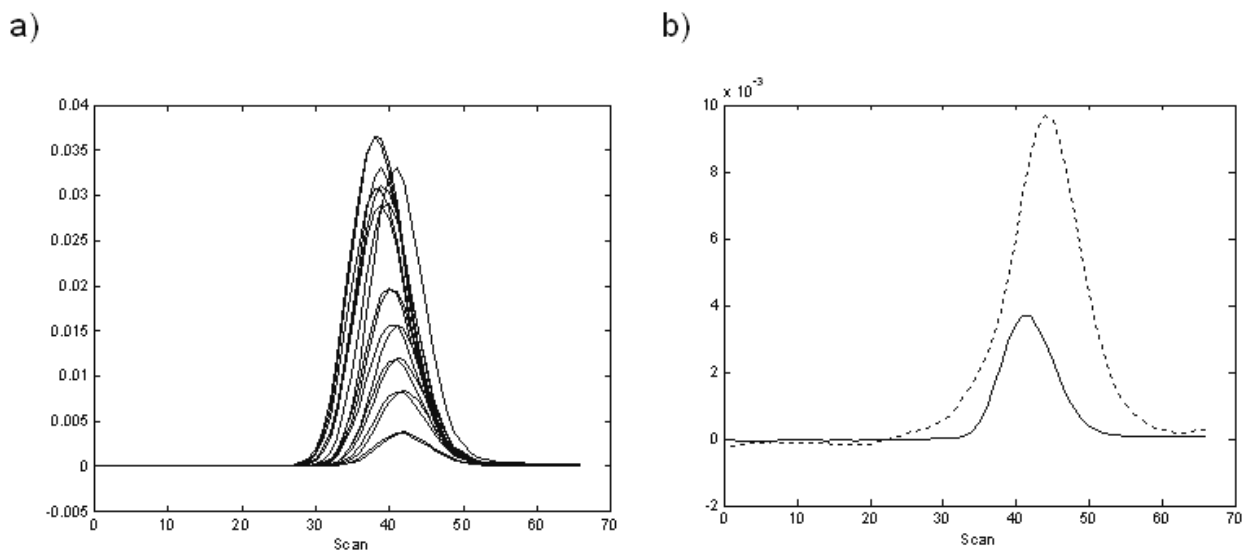


Figure 2. Chromatograms registered at 268 nm (a) of pure standards containing between 1.33 and 10.33 mg l⁻¹ of sulfadiazine (b) of a pure standard (solid line) and a kidney sample (dotted line) both containing 1.33 mg l⁻¹ of sulfadiazine in vial. Scan 1 and scan 66 are 6.19 and 7.60 minutes respectively.

However for the DTW algorithm to be effective not just any sequence of elementary transitions is allowed and some restrictions have to be imposed. The set of the allowed transitions to build the warping path is called look-up table and represented by $T^{(x,y)}$. x is the largest block distance covered by any of the rules in the table and y the maximum number of horizontal/vertical consecutive transitions allowed.

Once the optimal warping path has been found, a synchronization step is needed so that all the aligned samples (with size K) will have the same length as before warping (size M) [24].

4. Discussion and results.

Figure 1a shows the chromatogram of a pure standard containing 1.33 mg l⁻¹ of sulfadiazine for different wavelengths recorded between 242 and 395 nm. Chromatograms at low wavelengths, where the absorption of the mobile phase becomes more important, display not only baseline drift but also background. The lower the wavelength, the more important the drift and the offset. Besides, some wavelengths exhibit negative values of absorbance. On the other hand, it can be observed in Figure 2a that variations in the retention time from sample to sample become more important when gradient flow is applied.

A two-component PARAFAC model is built with the raw signals of pure standards of sulfadiazine. The chromatographic and spectral profiles estimated are shown in Figure 3a and Figure 3b respectively. The analysis of the chromatographic profiles concludes that one of the components, in solid line, is associated with the analyte

and the other one, in dotted line, jointly explains two effects, the baseline drift and the chromatographic shifts during the elution of the analyte. The last component does not contain chemical information related to the analyte and renders the models more complex.

To avoid these kinds of artefacts, different preprocessing techniques have been firstly applied to the signals of sulfadiazine in pure standards. An ideal preprocessing technique of a single component sample should modify the data so that the chromatogram will consist of a peak of rank one.

Figure 4 displays the signal of the pure standard, with 1.33 mg l⁻¹ of sulfadiazine in vial, corrected by all four preprocessing techniques, that is, centering across the sample mode (Figure 4a), first-order derivative (Figure 4b), second-order derivative (Figure 4c) and baseline correction (Figure 4d). Because of the dependence of the drift on the wavelength, the chromatogram of a standard is displayed at all variables (between 242 and 395 nm).

Then, the signals of kidney samples (Figure 1b) will be included in the data set to study the effect of the different nature of the samples with respect to pure standards. These signals will be more complex because of the matrix effects and the elution of interferences. As observed in Figure 1b, unknown compounds coelute with sulfadiazine and the baseline is not so well reproduced as in the case of pure standards. The background is evident but the baseline drift is not so clear. Negative values of absorbance are also observed. The matrix effects can be observed in Figure 2b which displays the chromatogram of a pure standard (solid line) and a kidney sample (dotted line) both at 1.33 mg l⁻¹ of sulfadiazine in vial. Despite the fact that the concentration in both standards is the same,

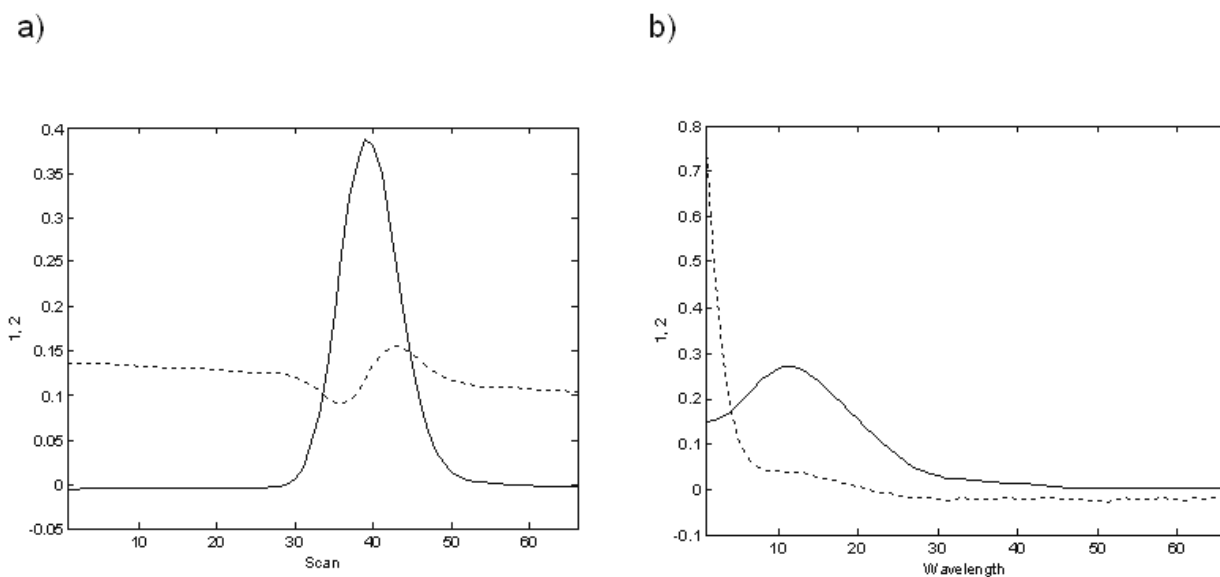


Figure 3. (a) Chromatographic and (b) spectral profiles estimated by a two-component PARAFAC model built with raw data of sulfadiazine. In solid lines the factor associated with sulfadiazine.

the chromatogram of the kidney sample is more intense (around three times) and wider as a consequence of the matrix effect. The retention time of kidney samples also changes and, at that wavelength, is longer with respect to pure standards. But now this change has two contributions, the chromatographic shifts (as for pure standards) and the elution of the interferences.

4.1. Techniques for baseline removal.

4.1.1. Centering.

It can be deduced from Figure 4a that centering across the sample mode removes both the background and the baseline drift at all wavelengths for pure standards. The shape of the chromatograms does not change but the baseline has been removed. Samples with low concentration display negative chromatograms (absorbance values below the mean) whereas standards with high concentration have positive chromatograms (absorbance values above the mean). Samples with intermediate concentration show absorbance values close to zero. In this paper, only one of many samples is shown in Figure 4a (1.33 mg l^{-1}) which is why data are not centered on zero. The baseline has been removed by centering which means that is constant over the sample mode and it does not depend either on the concentration of the analyte or on the order in which the standards were measured. It proves that the instrument, both the chromatograph and the detector, do not have a drift over time.

Regarding the spectral mode, the absorption of the mobile phase, mainly at low wavelengths, is eliminated. Therefore, it can be concluded that the baseline drift was caused by the absorption of the mobile phase whose

composition changes during the registration of the chromatograms.

As centering depends on the samples included in the data set, a new set is built with only the signals of kidney extracts. These signals (Figure 1b) contain the analyte and other matrix compounds. The result of centering across the sample mode is depicted in Figure 5. The offset and the baseline have been cancelled. Besides, the effect of the interference coeluting close to the analyte has been minimized such that the resolution between the chromatogram of the analyte and of the interference has increased.

Finally, the effect of the different sample nature in the data set is examined. To do so, a new data set is built by joining signals of pure standards and kidney samples in a tensor such that the dimension in the sample mode will be 38. Signals centered are shown in Figure 6a. Both the background and the baseline drift are corrected for all the samples which proves that they do not depend on the different sample nature and they are the same for pure standards and kidney extracts. In the case that there was extra variability in the baseline of the samples, centering across the chromatographic mode (double centering) might solve the difficulty. However one should take into account that centering across two modes should be performed sequentially [13,14]. Otherwise additional systematic variation could be included in the signals and consequently the complexity of the models would increase. The additional peak in Figure 6a with respect to Figure 4a might be due to the variation of the retention time of kidney samples with respect to pure standards. As the retention time is longer for kidney samples a peak with positive absorbances appears around scan number 50.

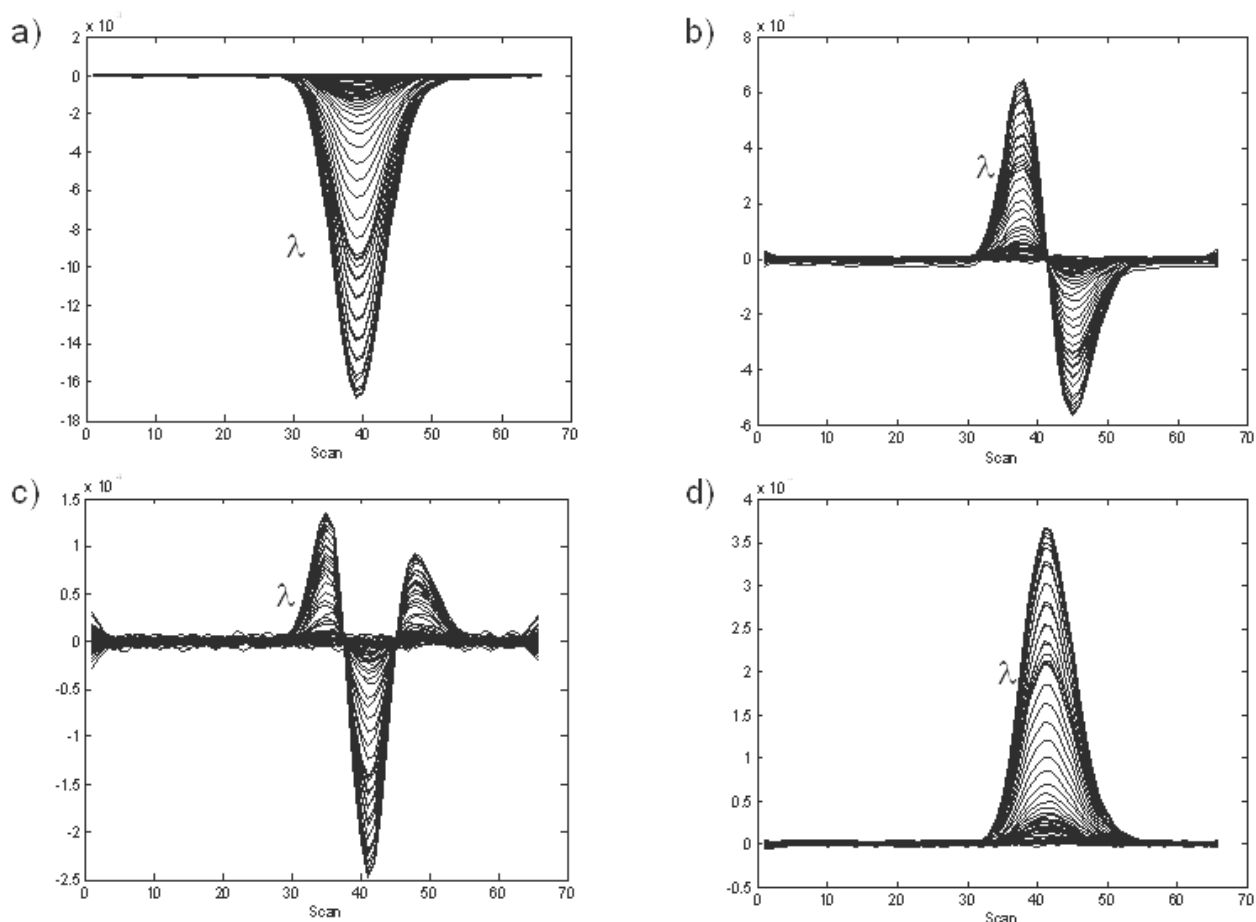


Figure 4. Preprocessed chromatograms at different wavelengths of a pure standard containing 1.33 mg l^{-1} of sulfadiazine. (a) Centering the set of pure standards across the sample mode, (b) first-order derivative, (c) second-order derivative and (d) baseline subtraction.

4.1.2. First and second-order derivatives.

The first-order derivative removes a constant background of varying magnitude but not a baseline drift (slope) which is transformed into a constant offset. The slope of the drift is removed by the second-order derivative. The first and second-order derivatives of a pure standard are shown in Figure 4b and Figure 4c and those of a kidney sample in Figure 6b and Figure 6c respectively. In both cases, the effect of the baseline is minimized and what is more, the resolution between the analyte and the interference is enhanced. However, it can be observed that despite smoothing, the baseline is noisier above all for the second-order derivative. The greater the order of the derivative the worse the signal-to-noise ratio and the greater the shift of the maximum peak [32] with respect to the raw data (compare the retention times in Figure 4a, Figure 4b and Figure 4c).

The advantage of derivatization with respect to centering is that the effectiveness of derivatization to remove the offset is independent of the rest of the samples in the data set. The signal centered across the sample mode

is obtained by subtraction of the column mean of the samples from every element. Therefore, the signal of all the samples in the set contributes to a certain extent to the centered signal of one sample. The effect can be observed comparing the centered signals of a kidney sample when only kidney samples are included in the set (Figure 5) and when the set is constituted by pure standards and kidney samples (Figure 6a). Differences between both signals are clear.

4.1.3. Baseline correction.

The results of the fourth correction technique based on estimating the baseline by a linear regression (equation 2) and subtracting it from the raw signal are depicted in Figure 4d. The background and the drift have been eliminated. However, the correction of the signals of kidney samples (Figure 6d) is not completely effective. The baseline drift and the offset have been eliminated but negative absorbances are found at elution times close to the peak of the analyte (before and after). The disadvantage of applying this correction technique is that the resolution between the peaks should be high enough to

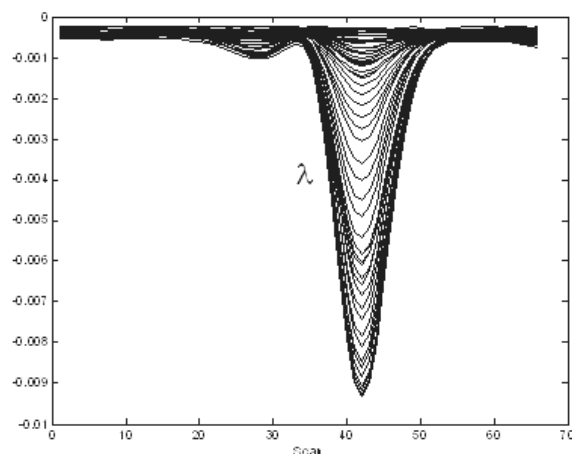


Figure 5. Centered chromatograms at different wavelengths of a kidney sample enriched with 1.33 mg l^{-1} . Centering was performed across the sample mode and only taking into account signals of kidney samples.

have an adequate amount of points before and after the elution to describe the baseline. As happened in this paper, it might be difficult to estimate the baseline of complex chromatograms because of the interferences. Besides, an additional metaparameter defining the baseline windows is needed.

4.2. Techniques dealing with chromatographic shifts.

As previously stated, one of the basic assumptions of PARAFAC is that the data must be trilinear. Because of the chromatographic changes from sample to sample both in retention time and shape of the signals (see Figure 2a for raw data and Figure 7a for first-order derivative signals), chromatography often does not comply with this condition. In this example the maximum difference in the retention time of sulfadiazine standards is about 4 seconds. Sometimes the variations are not so important and the PARAFAC model under some restrictions [6] can handle them. In other cases, as the presented in this paper, the estimations derived from the model are biased (Figure 3).

Two alternatives are proposed in this paper to cope with chromatographic shifts. The first one consists of aligning the signals (section 4.2.1) by the dynamic time warping (DTW) algorithm. The second one is the performance of other three-way models not affected by

changes in one of the modes as PARAFAC. This is the case of PARAFAC2 (section 4.2.2) which does not assume invariant profiles.

4.2.1. Dynamic time warping (DTW).

It has been proved [24] that the DTW algorithm is an effective technique for aligning chromatograms. However, some aspects must be taken into account before applying DTW. Changes in the baseline of both sample and reference chromatograms can have a negative impact on the DTW ability for correcting chromatographic shifts. Therefore, one of the techniques detailed in section 4.1 should be applied for removing the baseline from the signals. In this section, the first-order derivative was applied to simultaneously remove drift and baseline differences between the samples and the reference.

The effectiveness of the DTW algorithm was tested on three different data sets:

- i) Data set containing 18 pure standards of sulfadiazine (section 4.2.1.1) so that the signals have neither interferences nor matrix effects. Different look-up tables are applied for warping signals.
- ii) Data set containing 18 pure standards and 18 kidney samples enriched with sulfamethoxypyridazine (section 4.2.1.2) so that the effect of aligning signals on a rank-one PARAFAC model can be examined.
- iii) Data set containing 18 pure standards plus 20 kidney samples enriched with sulfadiazine (section 4.2.1.3) to analyse the effect of aligning signals on a rank-two PARAFAC model.

4.2.1.1. Warping signals of pure standards.

The signals to be aligned are shown in Figure 7a. The sample containing 5.33 mg l^{-1} of sulfadiazine was chosen as the reference because it has intermediate retention time with respect to the other samples.

The effectiveness [24] of the DTW algorithm to align the signals depends both on the number of time-points (segment length) covered by rules in the look-up table and on the maximum number of horizontal/vertical transitions allowed by the look-up table. Depending on the values of both parameters two extreme cases can be found.

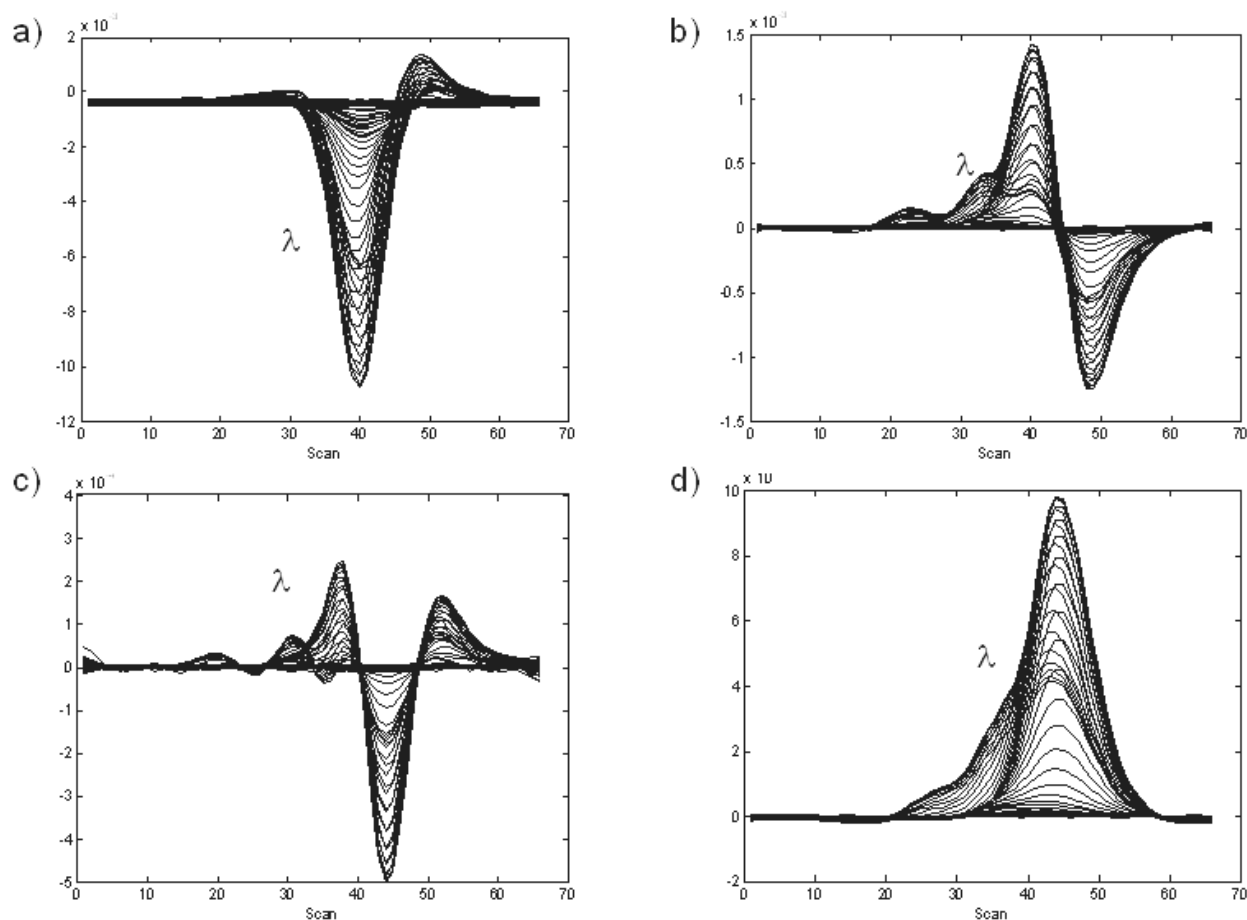


Figure 6. Preprocessed chromatograms at different wavelengths of a kidney sample spiked with $40 \mu\text{g kg}^{-1}$ of sulfadiazine (1.33 mg l^{-1} in vial). (a) Centering the set of pure standards and kidney samples across the sample mode, (b) first-order derivative, (c) second-order derivative and (d) baseline subtraction.

Unconstrained or poorly constrained DTW is too flexible because it allows for many consecutive horizontal/vertical transitions of short segments. Under these conditions, small features included in the chromatogram such as noise could be warped and what is more, the chromatographic profile might be deformed. This artefact has been depicted in Figure 7b, which represents the signal of a standard before and after warping. The segment length was fixed at 2, i.e. there will be 33 segments, and the maximum number of consecutive vertical/horizontal transitions at 1, look-up table $T^{(3,1)}$. The shape of the warped peaks (dotted line) is visibly distorted with respect to the original chromatogram (solid line).

Stricter restrictions entail fewer consecutive horizontal/vertical transitions of longer segments and prevent peak distortion. However, too constrained DTW may cause signal adjustment to fail due to the fixed end point restriction [34]: as the start and the end points are fixed, the extremes of the chromatogram might be more difficult to match. Figure 7c displays the chromatographic profile of 18 samples after DTW, $T^{(65,1)}$, was applied.

Signals are not aligned due to the excessive constraints imposed to the DTW algorithm.

Some authors have proposed [9,24] use the width of the narrowest chromatographic peak to be aligned as segment length. The band width in this example is around 30 time-points. As no alignment was achieved with this constraint two alternatives proposed in ref. [24] were followed. The first one consists of extending the baseline by including, at both extremes of the chromatogram, a segment containing only white noise. The second option for avoiding the effects of DTW with rigid slope constraints is to duplicate of the number of points of the chromatogram by linear interpolation. The extra points introduced by both these methods are eliminated after the warping procedure is completed. 400 baseline points of white noise were added at both extremes of the chromatogram and the number of data points in the actual chromatogram was duplicated. As a consequence of the latter procedure, the peak width is also doubled, thus a $T^{(129,1)}$ table, which uses 60-point segments was used to prevent peak distortion. The results are shown in Figure 7d. Chromatograms have been aligned and what it is more

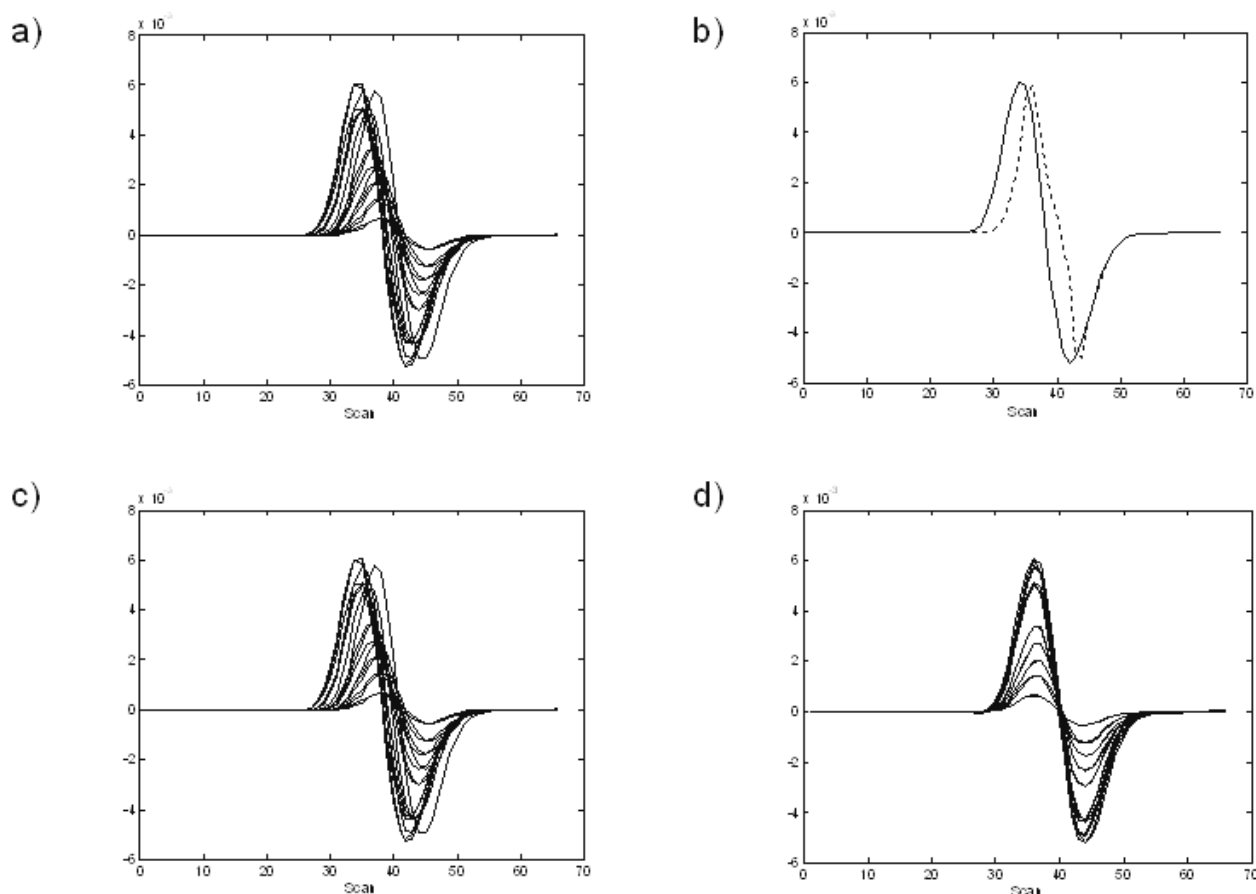


Figure 7. First-order derivative chromatograms of pure standards of sulfadiazine at 268 nm. (a) non-aligned signals, (b) original first-order derivative chromatogram of a pure standard containing 10.33 mg l^{-1} of sulfadiazine (solid line) and aligned signal (dotted) with the look-up table, $T^{(3,1)}$ (c) aligned signals with the look-up table $T^{(65,1)}$ and (d) aligned chromatograms with the look-up table $T^{(129,1)}$, duplicating the number of points in the chromatograms and adding a baseline with 400 points of white noise at the extremes of the chromatogram.

important its shape does not change with respect to the original one.

4.2.1.2. Warping signals of pure standards and kidney samples without interferences.

In this section, the data set of sulfamethoxypyridazine will be used. There are no interferences (see raw and first-order derivative signals in Figure 8) so a one-component PARAFAC model was fitted with the non-aligned first-order derivative signals. The percentage of variance explained by the model is 75.70%.

The effect of the chromatographic shifts on the loadings estimated by PARAFAC in the different modes will be analysed because those profiles can be used to identify (spectral loadings) and quantify (scores) the analytes. The correlation coefficient between the spectrum estimated by PARAFAC and the spectrum of the analyte in pure standards is 0.9999 which implies that the PARAFAC model satisfactorily determines the true underlying profile

in the second mode even when signals are affected by chromatographic shifts.

As PARAFAC is a decomposition model, the scores in the sample mode can be used to quantify by building a calibration model between the scores of pure standards (the first instrumental replicate) and the concentration. To evaluate the capability of prediction of the model in pure standards, the second instrumental replicate of pure standards is quantified using the scores and the regression model built. The mean from the relative errors in absolute value is 4.88% which shows that, despite the chromatographic shifts, the ability of the PARAFAC model to predict the concentration of standards whose signals are similar to those of the training set is good.

However, greater differences were found when the instrumental replicates of kidney samples were quantified. The same extract was injected twice. For example, the predicted concentration in the sample enriched with 7.33 mg l^{-1} is 2.60 and 3.62 mg l^{-1} for the first and the second instrumental replicate respectively, and in terms of recovery, $(\text{Conc}_{\text{pred}}/\text{Conc}_{\text{true}} \times 100)$, 35.47 and 49.33%

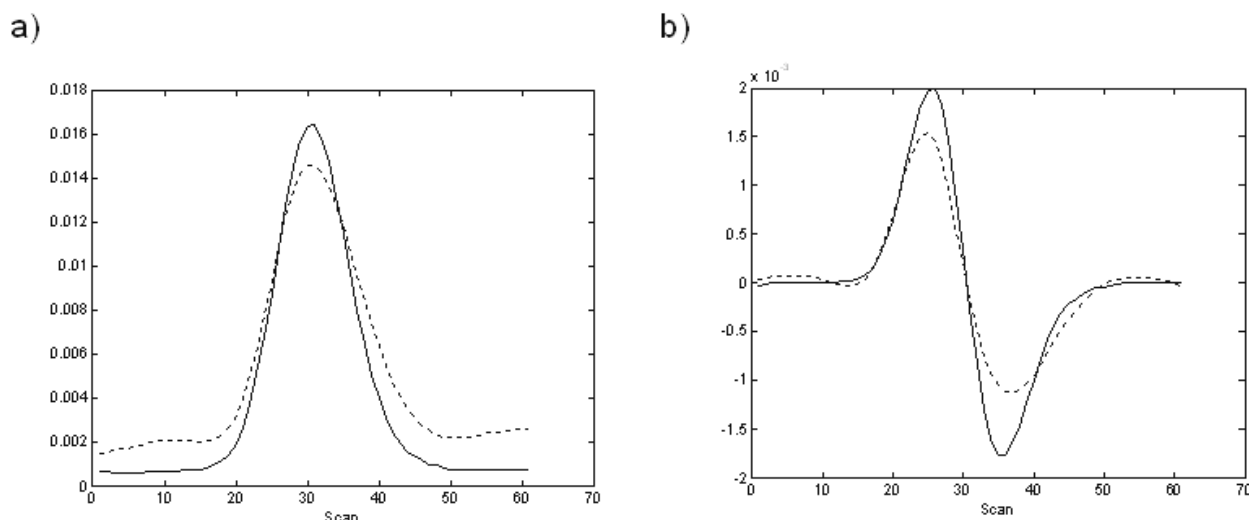


Figure 8. (a) Chromatogram and (b) its first-order derivative of sulfamethoxypyridazine (2.33 mg l^{-1} in vial) in pure standards (solid line) and kidney samples (dotted line). Signals are displayed at 268 nm. Scan 1 refers to 9.52 minutes and scan 61 to 10.02 minutes.

respectively. The difference between the recoveries of both instrumental replicates is around 14%, which is too high. This proves that the model lacks reproducibility and is unstable. Similar results were obtained for the rest of samples spiked at different levels, which rejects the idea of being outliers. The analysis of other sulfonamides (results not shown) leads to the same conclusions.

To check whether the lack of reproducibility of the PARAFAC model is related to the chromatographic shifts or not, the first-order derivative signals were aligned by DTW. The sample with concentration $280 \mu\text{g kg}^{-1}$ of sulfamethoxypyridazine in kidney was chosen as reference. A baseline with 800 time-points was added to both extremes of the chromatogram. The number of points of the chromatogram (66) were duplicated by linear interpolation and the look-up table $T^{(159,1)}$ was applied.

The one-component PARAFAC model built with the aligned signals explains 99.08% of the variability, which is greater than the variance explained by the model built with the non-aligned signals, 75.70%. This means that approximately 23% of the variability found in the non-warped signals is due to the chromatographic shifts that have successfully been removed by DTW.

Regarding the spectral profile, the correlation coefficient between the spectra estimated by PARAFAC and the spectrum of a pure standard is greater than 0.999 which allows the analyst to identify the analyte. The advantage of aligning signals is found in the sample mode (quantification). The predicted concentration of the kidney sample spiked with 7.33 mg l^{-1} is 4.54 and 4.59 mg l^{-1} for the first and the second instrumental replicate respectively, which means 61.88 and 62.65% of recovery. It can be observed that the difference between both replicates is less than 1%. This result is more precise than that obtained with the non-aligned signals (difference around 14%) and

demonstrates that the estimations in the sample mode are more reliable after the chromatograms have been warped. For the rest of the samples enriched at different levels of sulfamethoxypyridazine, differences between instrumental replicates are smaller than 1%, which shows the need to align the chromatograms before applying PARAFAC to avoid the loss of data trilinearity.

4.2.1.3. Warping signals of pure standards and kidney samples with interferences.

Figure 1b shows the chromatograms of sulfadiazine in an extract of kidney. Some unknown interferences elute close to the analyte which modifies not only the retention time but also the shape of the peaks with respect to the chromatogram of the pure standard (Figure 1a).

With the aim of reducing the rank of the model, the first fifteen wavelengths were removed so that a two-component PARAFAC model can be fitted with the non-aligned first-order derivative signals. The percentage of explained variance is 97.75%. The estimation of the spectral profiles is not affected by the chromatographic shifts and the correlation coefficient between the spectrum estimated by PARAFAC and that of a pure standard is higher than 0.9999. Figure 9 displays the estimations of PARAFAC in the chromatographic mode. The solid line represents the profile associated with sulfadiazine whereas the dotted line should correspond to that of the interference. It can be observed that the loadings in the chromatographic mode of the factor associated with the interferences are not the first-order derivative of a chromatogram because the chromatographic shifts are also included in this component as they were in Figure 3a.

The scores in the concentration mode of the component associated with the analyte were used to

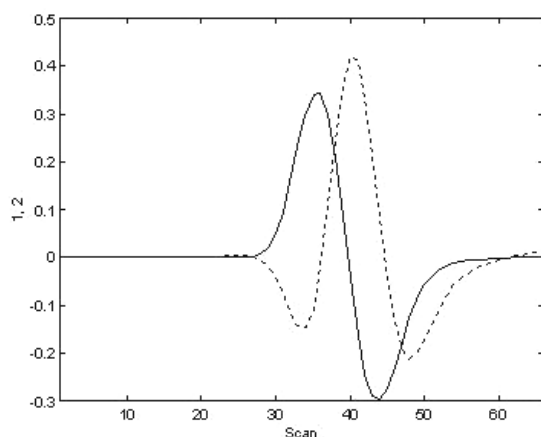


Figure 9. Chromatographic profiles estimated by the two-component PARAFAC model built with the first-order derivative signals of sulfadiazine. The factor associated with the analyte is in solid line and the factor associated with the interference and the chromatographic shifts in dotted line.

quantify as it was described in the previous section. The mean of the relative errors in absolute value when the concentration of pure standards is predicted is 1.95%. The predicted concentration of a kidney sample enriched with 7.33 mg l^{-1} is 4.55 mg l^{-1} for both instrumental replicates that is, 62% of recovery. This means that the two-component PARAFAC model built with the non-aligned signals of sulfadiazine is stable and robust, unlike the one-component model built for sulfamethoxypyridazine. The reason might be that the variability due to the chromatographic shifts has been included in the model through the component related to the interference (dotted line in Figure 9).

Signals were aligned choosing the sample containing $280 \text{ } \mu\text{g kg}^{-1}$ of sulfadiazine in kidney as reference. The number of points in the chromatogram was duplicated by linear interpolation and the baseline was extended with 600 points of white noise at both extremes of the chromatogram. The look-up table $T^{(119,1)}$ was applied and a two-component PARAFAC model was fitted. The percentage of explained variance is 99.42%. The profiles estimated in all three modes are displayed in Figure 10. Because of the uniqueness property of PARAFAC there should be one factor associated with the analyte and another one with the interference. However it can be observed in Figure 10a that there is one factor that might be associated with the analyte, plus sign, increasing concentration of the analyte in the calibration curve, but

the other one (circles) is not so clearly related to the interference. Most of the scores of pure standards, samples 1-18, are around zero, which is logical because pure standard do not contain interferences whereas those of kidney samples, samples 19-36, are different from zero. Nevertheless, it can be observed that the scores of pure standards 11, 12 and 13 are of the same order as kidney samples. This means that this factor is not completely associated with the interference and it explains another effect. From the quantitative point of view, the concentration predicted in a kidney sample spiked with 7.33 mg l^{-1} is 5.14 and 5.00 mg l^{-1} for the first and the second instrumental replicate respectively, which corresponds to 70.12% and 68.21% of recovery. The difference between both replicates is not excessive (around 2% in recovery), but the estimated concentration is higher than that estimated with non-aligned signals (4.55 mg l^{-1} , 62% of recovery). The reason is that the shape of the chromatograms of kidney extracts and of pure standards are different (see Figure 1). As warping minimizes the distance between features in the reference and in the sample, when two peaks are highly overlapping in either chromatogram, and the other chromatogram only contains one peak, the alignment fails. This result together with the fact that the scores of the factor associated with the interference in pure standards 11, 12 and 13 are similar to those in kidney samples allow one to conclude that DTW is not effective to align the signals which contain interferences affecting the shape of the chromatogram of the analyte. Thus, analyte, interference and shift are still mixed after the PARAFAC decomposition. These conclusions will be supported with the results obtained with the PARAFAC2 model which is theoretically unaffected by the shifts.

4.2.2. The PARAFAC2 model.

The comparative results between PARAFAC, DTW+PARAFAC and PARAFAC2 are shown in Table 1 for sulfamethoxypyridazine. A one-component PARAFAC2 model was carried out with the first-order derivative of the chromatograms of sulfamethoxypyridazine whose chromatograms are only affected by shifts but not by interferences. The percentage of explained variance is 99.83% which means that most of the variability has been explained. The percentage is close but higher than that obtained by PARAFAC built with warped signals, 99.08%.

The estimation of the spectral profile is good and the correlation coefficient between the spectra estimated and that of a pure standard is close to 1.

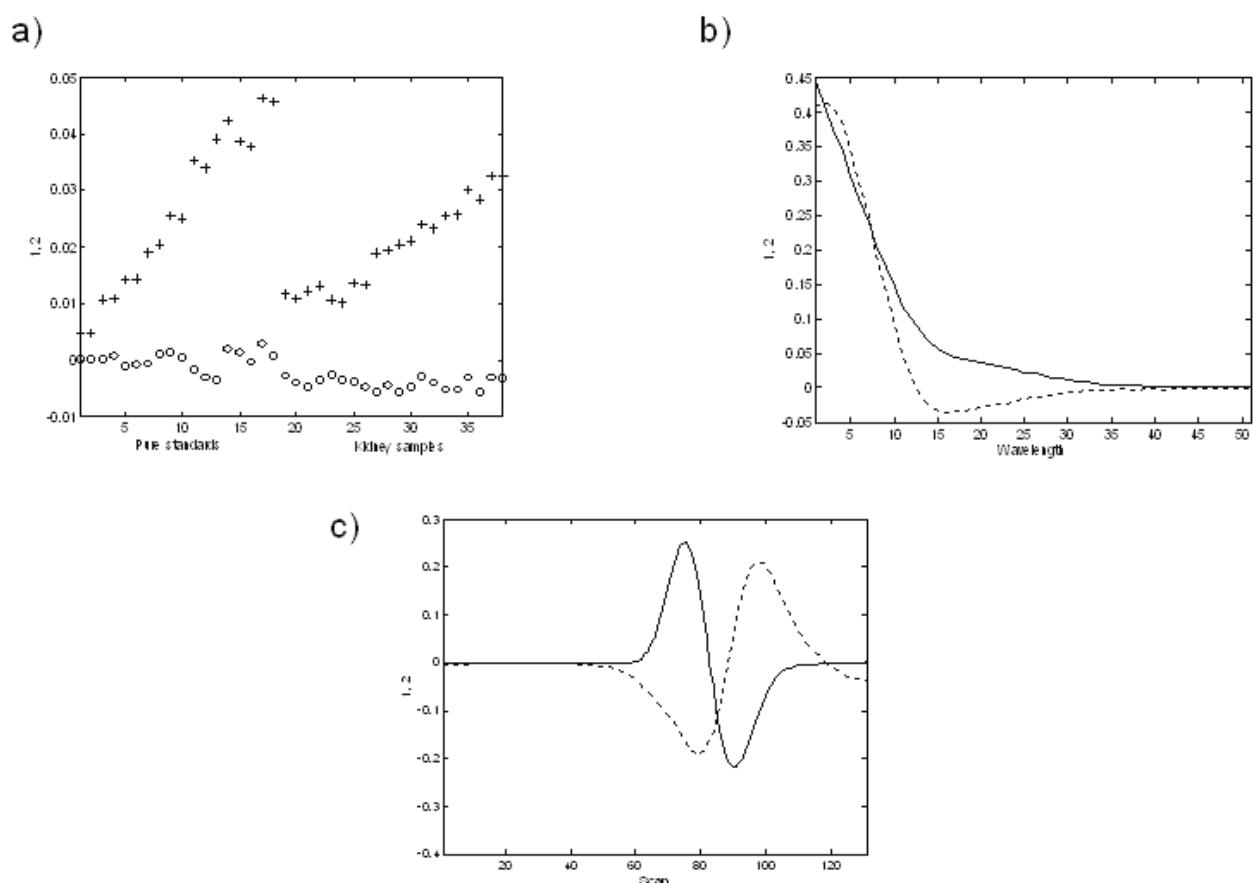


Figure 10. Loadings estimated by a two-component PARAFAC model after signals of pure and kidney standards of sulfadiazine were aligned. (a) Sample mode (b) spectral mode and (c) chromatographic mode.

From the quantitative point of view, the mean of the relative errors in absolute value when pure standards are quantified is 2.86% which is smaller to that of PARAFAC (4.88%) and close to that of DTW+PARAFAC (2.92%). Regarding kidney samples, the concentration of two instrumental replicates of a sample enriched with 7.33 mg l⁻¹ of sulfamethoxypyridazine is 4.54 and 4.57 mg l⁻¹ respectively (61.95 and 62.29% of recovery). The difference between both replicates is small (less than 1%) which proves that PARAFAC2 is more stable and is not so

affected by the chromatographic shifts as PARAFAC whose difference between both instrumental replicates is 14%. It can be observed in Table 1 that, in the case of specific signals, the results obtained with PARAFAC2 are always better than those of PARAFAC and similar to those of PARAFAC with warped signals. This means that the DTW algorithm successfully corrects the chromatographic shifts which cause the lack of trilinearity in the data.

To find out to what extent PARAFAC2 is

Table 1. Comparative study of the PARAFAC and the PARAFAC2 models built with non-aligned and aligned signals of sulfamethoxypyridazine. Percentage of explained variance, mean of the relative errors in absolute value (%) when pure standards are quantified and recovery of two instrumental replicates of a sample containing 7.33 mg l⁻¹ of sulfamethoxypyridazine.

	PARAFAC		PARAFAC2	
	Non-aligned	Aligned (DTW)	Non-aligned	Aligned (DTW)
Explained variance (%)	75.70	99.08	99.83	99.83
Error (%)	4.88	2.92	2.86	2.85
R ₁ ^a , R ₂ ^b	35.45, 49.33	61.88, 62.65	61.95, 62.29	62.32, 63.04
(R ₁ ^a -R ₂ ^b)	(13.86)	(0.76)	(0.34)	(0.72)

^a: recovery (%) from the first instrumental replicate.

^b: recovery (%) from the second instrumental replicate.

influenced by the shifts, a one-component PARAFAC2 model was built with the first-order derivative chromatograms after warping. The percentage of explained variance is 99.83% which is the same as that obtained with the PARAFAC2 model with non-aligned signals. The qualitative and quantitative results from PARAFAC2 and from DTW+PARAFAC2 are almost identical (the last two columns in Table 1) which allows one to conclude that PARAFAC2 is not affected by the shifts.

The two-component PARAFAC2 model built with the signals of sulfadiazine, which contains interferences, explains 99.96% of the variability found in the data set. The concentration predicted in two instrumental replicates of a kidney sample spiked with 7.33 mg l⁻¹ is 4.55 mg l⁻¹ (62% in terms of recovery). This result is similar to that obtained with PARAFAC without signal alignment and proves that the concentration predicted by DTW+PARAFAC, 70.12% and 68.21%, is biased because the model does not distinguish between the analyte and the interference.

Conclusions

Through this paper some common artefacts in chromatography have been examined. It has been studied how they disturb the trilinearity of the data and/or increase the complexity of the models as well as the strategies to minimize their effect on the estimations of the PARAFAC model. Some of the strategies like centering across the sample mode, first and second-order derivatives and baseline correction are aimed at minimizing the effect of the baseline drift and of constant offsets.

It has been shown that the chromatographic shifts from run to run disarrange data trilinearity which causes lack of accuracy in the predictions of the PARAFAC model. The spectral profiles are not highly affected by the chromatographic shifts which might be a practical problem because a good identification through the spectral profiles does not necessarily provide true estimations of the analyte concentration.

For specific signals, chromatographic shifts have been corrected with the DTW algorithm. Once the time-shifts have been eliminated, the loadings in the sample mode become steadier and the estimations of PARAFAC and PARAFAC2 become similar. On the other hand, no difference has been found between the PARAFAC2 model and DTW+PARAFAC2 which demonstrates that the chromatographic shifts have an effect on PARAFAC but not on PARAFAC2 and that removing them the consistency of the PARAFAC model improves.

With non-specific signals, PARAFAC2 is more effective than DTW+PARAFAC because the procedure for warping overlapping signals with specific signals and vice versa is not effective.

References

- [1] Commission Decision of 12 August of 2002 implementing Council Directive 96/23/EC concerning the performance of analytical methods and the interpretation of the results (2002/657/EC).
- [2] J.L. Beltrán, J. Guiteras, R. Ferrer, *J. Chromatogr. A*, 802 (1998) 263.
- [3] H. Idborg, P.O. Edlund, S.P. Jacobsson, *Rapid Commun. Mass Spectrom.*, 18 (2004) 944.
- [4] D. Bylund, R. Danielsson, G. Malmquist, K.E. Markides, *J. Chromatogr. A*, 961 (2002) 237.
- [5] E. Comas, A. Gimeno, J. Ferré, R.M. Marcé, F. Borrull, F.X. Rius, *J. Chromatogr. A*, 1035 (2004) 195.
- [6] I. Garcia, L. Sarabia, M.C. Ortiz, J.M. Aldama, *Analyst*, 129 (2004) 766.
- [10] R. Bro, *Chemom. Intell. Lab. Syst.* 38 (1997) 149.
- [11] K.D. Booksh, B.R. Kowalski, *Anal. Chim. Acta*, 348 (1997) 1.
- [12] N.K.M. Faber, *Anal. Chem.*, 71 (1999) 557.
- [13] R. Bro, A.K. Smilde, *J. Chemom.*, 17 (2003) 16.
- [14] A.K. Smilde, *J. Chemom.*, 15 (2001) 19.
- [15] M. Martínez-Galera, J.L. Martínez-Vidal, A. Garrido-Frenich, M.D. Gil-García, *J. Chromatogr. A*, 778 (1997) 139.
- [16] J.H. Christensen, G. Tomasi, A.B. Hansen, *Environ. Sci. Technol.*, 39 (2005) 255.
- [17] P. Geladi, J. Forsström, *J. Chemom.* 16 (2002) 329.
- [18] C. Bessant, R.G. Brereton, S. Dunkerley, *Analyst*, 124 (1999) 1733.
- [19] C. Demir, R.G. Brereton, *Analyst*, 122 (1997) 631.
- [20] S. Zomer, C. Guillo, R.G. Brereton, M.H. Brown, *Anal. Bioanal. Chem.*, 378 (2003) 2008.
- [22] A. de Juan, R. Tauler, *J. Chemom.*, 15 (2001) 749.
- [23] K. Johnson, A. de Juan, S.C. Rutan, *J. Chemom.*, 13 (1999) 331.
- [24] G. Tomasi, F. van den Berg, C. Andersson, *J. Chemom.* 18 (2004) 1.
- [25] N.P.V. Nielsen, J.M. Carstensen, J. Smedsgaard, *J. Chromatogr. A*, 805 (1998) 17.
- [26] K.J. Johnson, B.W. Wright, K.H. Jarman, R.E. Synovec, *J. Chromatogr. A*, 996 (2003) 141.
- [27] E. Comas, R.A. Gimeno, J. Ferré, R.M. Marcé, F. Borrull, F.X. Rius, *Anal. Chim. Acta*, 470 (2002) 163.
- [28] B.M. Wise, N.B. Gallagher, E.B. Martin, *J. Chemom.*, 15 (2001) 285.
- [29] H.J. Ramaker, E.N.M. van Sprang, J.A. Westerhuis, A.K. Smilde, *Anal. Chim. Acta*, 498 (2003) 133.
- [30] R. Bro, C.A. Andersson, H.A.L. Kiers, *J. Chemom.*, 13 (1999) 295.
- [31] A. Smilde, R. Bro, P. Geladi, in *Multiway Analysis. Applications in the chemical sciences*. Wiley, 2004, Chichester, England.

- [32] B.G.M. Vandeginste, D.L. Massart, L.M.C. Buydens, S. de Jong, P.J. Lewi, J. Smeyers-Verbeke, Handbook of chemometrics and qualimetrics: Part B, Elsevier, 1998, Amsterdam.
- [33] A. Savitzky, M.J.E. Golay, Anal. Chem., 36 (1964) 1627.
- [34] V. Pravdova, B. Walczak, D.L. Massart, Anal. Chim. Acta, 456 (2002) 77.