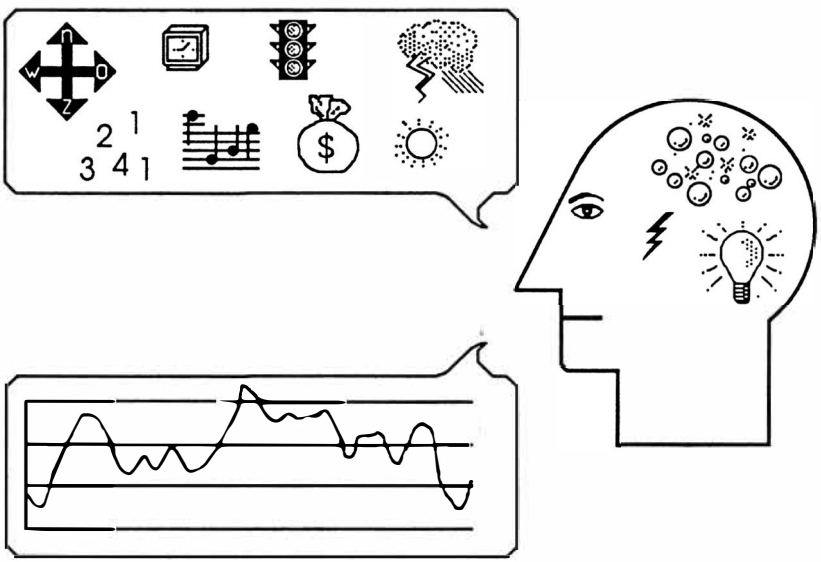
OPTIMAL SCALING OF TIME SERIES

Stef van Buuren



DSWO PRESS

OPTIMAL SCALING OF TIME SERIES

M&T SERIES 16

Available from DSWO Press
Wassenaarseweg 52
2333 AK Leiden
The Netherlands
Tel. (071) 273795

Earlier publications in this series:

Jacqueline Meulman, Homogeneity analysis of incomplete data.
M&T series 1, 1982

Pieter M. Kroonenberg, Tree-mode principle component analysis.
M&T series 2, 1983, reprint 1989

Jan de Leeuw, Canonical analysis of categorical data.
M&T series 3, 1984

Ronald A. Visser, Analysis of longitudinal data in behavioural and social research
M&T series 4, 1985

John P. van de Geer, Introduction to linear multivariate data analysis.
M&T series 5, volume 1 & 2, 1986

Jacqueline Meulman, A distance approach to nonlinear multivariate analysis.
M&T series 6, 1986

Jan de Leeuw, Willem Heiser, Jacqueline Meulman, Frank Critchley (editors), Multidimensional data analysis.
M&T series 7, 1987

Peter G.M. van der Heijden, Correspondence analysis of longitudinal categorical data.
M&T series 8, 1987

Jan van Rijckevorsel, The application of fuzzy coding and horseshoes in multiple correspondence analysis.
M&T series 9, 1987

Abby Israëls, Eigenvalue techniques for qualitative data.
M&T series 10, 1987

Eeke van der Burg, Nonlinear canonical correlation and some related techniques.
M&T series 11, 1988

Kees van Montfort, Estimating in structural models with non-normal distributed variables: some alternative approaches.
M&T series 12, 1989

Jan T. A. Koster, Mathematical aspects of multiple correspondence analysis for ordinal variables.
M&T series 13, 1989

Catrien C. J. H. Bijleveld, Exploratory linear dynamic systems analysis.
M&T series 14, 1989

Henk A. L. Kiers, Three-way methods for the analysis of qualitative and quantitative two-way data.
M&T series 15, 1989

Stef van Buuren, Optimal Scaling of Time Series.
M&T Series 16, 1990

OPTIMAL SCALING OF TIME SERIES

Stef van Buuren

Department of Psychonomy, Methodology Unit
University of Utrecht

1990 DSWO Press, University of Leiden

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Van Buuren, Stef

Optimal scaling of time series / Stef van Buuren. - Leiden: DSWO Press. - Ill. - (M & T series; 16)
Also publ. as Thesis Utrecht, 1990. With index, ref. – With summary in Dutch.
ISBN 90-6695-040-4
SISO 300.6 UDC 303.733
Subject headings: multivariate analysis / time series analysis.

This book was written on a Mac II using the Nisus wordprocessor. Proofs and final prints were produced on a LaserWriter Plus at 'O.P.&P. Tekstwerk B.V.', Utrecht. The body text is set in New Century Schoolbook and the figure captions and headers are set in Avant Garde. Special symbols are taken from the Symbol, Zapf Chancery and Zapf Dingbats fonts. Computations were performed in APL.68000 Graphics were created with Excel, Cricket Graph and SuperPaint.

Cover drawing: Stef van Buuren and Eveline Kroezen
Printed by Sinteur b.v., Leiden
and by Printing office 'Karstens Drukkers bv, Leiden'

ISBN 90-6695-040-4

Aan mijn ouders

CHAPTER 1

Introduction

"Nothing disturbs me more than time and space; and yet nothing disturbs me less, since I never think about it" Charles Lamb wrote. Sure enough, the analysis of data that are ordered in time and space is not easy: it can even be disturbing now and then. This book is written for those who are interested in analyzing categorical time series. The emphasis will be on multivariate time series data and on applications in the behavioral sciences. The introductory chapter phrases the central problem and guides the reader through the remainder of the book.

1.1 Time series

The world is full of time series. Perhaps the best known time series is the Dow Jones index. We have all seen those typical line plots in the financial section of the principal newspapers that graph the development of the major stock prices. But there exist many other series.

A time series consists of a sequence of values. The typical property of a time series is that its values are ordered in some sense. The values can be ordered in time, but they can also be ordered in space, time and space or along other criteria. For example, a meteorologist may be interested in the distribution of the daily amount of rainfall in different locations over an entire continent at one point of time.

The values of the series need not be numerical. Instead a series can be an ordered sequence of a finite number of discrete states. We call this kind of series categorical. Categorical time series are very common. Elementary piano music consists of simple notes played one after another. Each note falls into a category. Stretching a DNA spiral renders a long sequence of peptides. There are four different peptides, so we have a series with four categories. At its most basic level, a computer program is a binary time series.

Here we study methods to analyze categorical time series. We focus on time series that result from research in the behavioral sciences. This implies that the value of our methods for composing and analyzing music, cracking genetic codes and computer programming is very limited.

1.2 Optimal scaling

We distinguish between two major approaches to the analysis of categorical data in the social sciences: loglinear analysis and optimal scaling. Loglinear analysis aims at modelling a contingency table by decom-

posing the logarithm of the observed frequencies by a linear model. Standard references are Bishop, Fienberg and Holland (1975), Goodman (1978) and Fienberg (1980). There exists a well established tradition in analyzing time dependent data using loglinear analysis (see Bishop et al., 1975; Landis & Koch, 1979; Mason & Fienberg, 1985; Van der Heijden, 1987). According to Van der Heijden (1987, 11) the success of the loglinear approach in the analysis of time dependent data is explained by the fact that it is relatively easy to translate the Markov model into a loglinear model. Drawbacks of the loglinear approach are the empty cell problem and difficulties regarding the number and interpretation of the model parameters. In general, if the number of variables or categories grows, the practical usefulness of the loglinear model tapers off.

Broadly speaking, optimal scaling is a method to analyze categorical data as numerical data. It works by assigning numbers to categories. These numbers have the property that they are optimal with respect to some well-defined criterion. For example, in regression analysis such a criterion would be to maximize the multiple correlation. The transformation can be restricted to preserve the measurement level of the data. Optimal scaling accommodates for mixtures of nominal, ordinal and interval measurement levels. Optimal scaling systems have been developed by Gifi (1981), Young (1981) and others. See Chapter 3 for more references. So far, the analysis of categorical time series has received little attention within the optimal scaling tradition. There exist some isolated applications to time series (e.g. Visser, 1982, 146; Van der Burg, 1984) and event history data (Deville & Saporta, 1980, 1983; De Leeuw et al., 1985; Van der Heijden 1987) but there is still room for growth. Bijleveld (1989) explores optimal scaling of time series from a state space perspective. We concentrate on the possibility for using the optimal scaling technique in a number of other time series models.

1.3 Formulation of the problem

The initial goal of this dissertation was to synthesize dynamic factor analysis and optimal scaling techniques. Dynamic factor analysis is a method to determine the most typical latent time series from a multivariate time series. Regarding the work of Stobberingh (1972), Molenaar (1981, 1985) and Immink (1986) the University of Utrecht has established a tradition on dynamic factor analysis. The Gifi system matured at the University of Leiden. The resulting dynamic factor analysis for categorical data would be able to extract latent time series under optimal scaling of variables.

Since the dissertation by Bijleveld (1989)[1] on state space models also covers dynamic factor analysis of categorical series, it was decided to widen the scope of this dissertation a little. The main question studied in this dissertation is:

In what way can we integrate time series analysis and optimal scaling ?

This is a simple question. As with most simple questions, many different answers can be given. First, we need to make clear which types of time series analysis we want to study. Next, we have to decide on the precise form of the optimal scaling technique. Finally, we must devise a plan to amalgamate the two.

1.4 A closer look at the problem

To start with the first issue: we limit ourselves to time domain analysis as opposed to frequency domain analysis. This excludes for example Fourier analysis from our discussion. Also, the emphasis will be on

1) Bijleveld (1989, University of Leiden) also treats the analysis of categorical time series, but it differs from the present text in its orientation towards state space modeling. The dissertations are independent of each other to a large extent, but some common elements are the emphasis on optimal scaling, the use of the backshift matrix and the type of algorithm.

multivariate rather than univariate models. An exception is Chapter 4 which deals with univariate methods. Moreover the autoregressive model plays a much larger role than the moving averages model. More specifically, we will study ARMA models, state space models, canonical correlation analysis, dynamic factor analysis and intervention analysis. These models appear to be of primary importance in the social sciences.

The optimal scaling techniques we use are derived from the Gifi system of nonlinear multivariate analysis (Gifi, 1981). This comprehensive system is based on least squares data fitting for a large class of linear models. Each category of a variable defines a parameter of the model to be fitted. Minimizing an appropriate loss function over these category parameters yields numerical estimates that serve to assign a score to each category. Each category is thus scaled optimally with respect to the least squares solution.

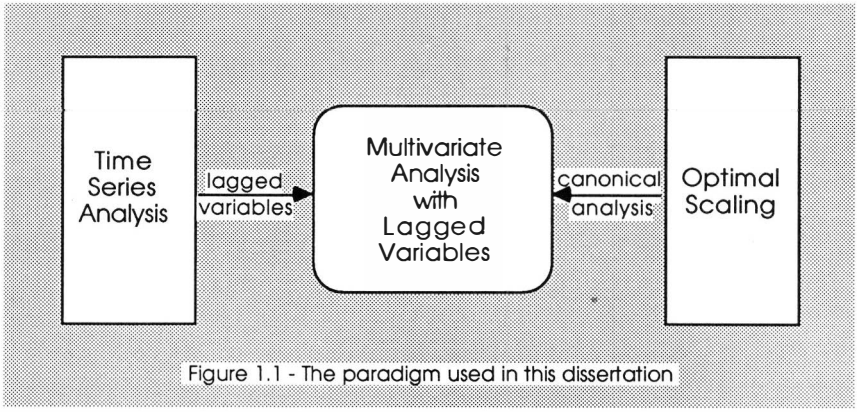


Figure 1.1 - The paradigm used in this dissertation

The blending of time series analysis and optimal scaling as presented here works by translating both into a common framework. This framework consists of least squares approximation, canonical correlation analysis and lagged variables. Canonical correlation analysis is a useful multivariate technique when combined with optimal scaling since many scaling methods can be formulated in terms of it. Lagged variables

provide a way to formulate time dependent models as multivariate problems. The central idea of this dissertation is to express a time series model as a multivariate optimal scaling problem with lagged variables. Figure 1.1 illustrates the idea. We view time series analysis as a branch of multivariate analysis in which lagged variables appear in the model.

1.5 Fields of application in psychology and sociology

Although the mass of psychological and sociological research is cross-sectional, time series are to be found at various places. Below we portray some of these fields.

Single subject research is popular in clinical psychology for evaluating the effect of psychotherapy. References are Glass, Willson and Gottman (1975), Hersen and Barlow (1976), Jones et al. (1977), Kratochwill (1978) and Kazdin (1980, 1982). The importance of this idiographic approach to psychology has been repeatedly emphasized by Allport (1961). In the N=1 type of study the investigator is usually interested in the effect of an intervention on the behavior of the subject. The observed series are usually very short. In about 80% of the studies published in the Journal of Applied Behavior Analysis the number of baseline (pre-intervention) observations is lower than 10 (cf. Huitema, 1986, 205). This is on the lower side for a reliable application of t-tests and other inferential procedures. A famous example of a multiple time series are the data gathered by Mefferd, Moran and Kimble (1958) which are discussed by Anderson (1963) and others. This series contains cognitive and physiological measures of a schizophrenic for a period over 200 days. During this period the investigators administered four different treatments. Chapter 5 contains an example of intervention analysis in a crime prevention program.

Another type of time series data is the event history. An event history specifies not only the observed sequence of successive states, but also how long a subject remains in a each state. For example, we can record the

dates at which important individual events take place (e.g. change of health, becoming a Hells Angel, getting a Ph.D.) as well as those applying to social collectivities (e.g. changes in political regimes, outbreaks of strikes, riots and wars). At every instance of time we know the state of the object under study so it is possible to recode the event history into a categorical time series. Event histories can also result from time-allocation studies. The goal is there to determine how people allot their time among various daily activities. See for example Winship (1978) and Harvey et al. (1984). Dynamic models for event history data have been proposed by Singer and Spielerman (1976), Tuma et al. (1979), Flinn and Heckman (1982) and others. Social event history data have also been analyzed with Box-Jenkins methods by Hibbs (1977), Vidgerhous (1978) and Land (1980).

Dyadic interaction data are also time series. This kind of data registers essential aspects of the communication between two individuals. An example comes from observational research in developmental psychology. Every few seconds the behavior states of both the mother and her baby are scored into one of seven categories (cf. Van den Boom, 1988). The interesting thing is to determine in what way the mother's conduct influences the behavior of the baby and vice versa. See for example Van der Heijden (1987) and Iacobucci and Wasserman (1988) for methods to analyze this kind of data.

Another interesting area is psychophysiological research. Electroencephalogram (EEG) and Galvanic Skin Response (GSR) recordings are often collected to investigate biological determinants of psychological processes. Some well known examples include the study of sleeping patterns, visual information processing and, more controversially, criminal behavior. Gregson (1983) describes a number of other examples. Typically this kind of data is examined by means of Fourier analysis in order to bring out the major periodic constituents. Section 5.3 illustrates the use of dynamic factor analysis applied to psychophysiological series.

In applied behavioral therapy, the therapist often instructs the client to record the a number of factors related to the complaints. For a client with hyperventilation such data may concern the time the symptom occurs, its duration, its seriousness and its frequency. The client can be asked to keep a record of his or her daily activities. There are various ways to maintain the record, e.g. using a diary, an event history list or a mechanical clock counter, but in all these cases the data make up multiple categorical time series. Examples can be found in Hoogduin (1989) and De Haan et al. (1989).

1.6 Overview

The book is organized such that the synthesis between optimal scaling and time series analysis becomes more complete each chapter. The first two chapters deal with time series analysis and optimal scaling as isolated fields. The highest degree of integration is achieved in Chapter 6. We have summarized the contents in Figure 1.2.

In Chapter 2 we start with some key concepts in time series analysis like autocorrelation, lagged variables and the autocorrelation box. This chapter also provides finite-sample matrix formulations of ARMA models, state space models, dynamic factor analysis and canonical correlation analysis. Chapter 3 is devoted to optimal scaling. First we focus on the idea of optimal scaling. Then starting from homogeneity analysis, a particular form of optimal scaling, we describe how this technique can be molded into various types of nonlinear multivariate analysis by using restrictions. The equality restriction will be treated in detail. The equality restriction is an important concept in our optimal scaling method of time series analysis.

Chapters 4, 5 and 6 deal with progressively more complicated techniques. Chapter 4 begins with a discussion of three elementary univariate filters that show how the autocorrelation of a series changes using optimal scaling. These filters are extended to simple, seasonal and multiple

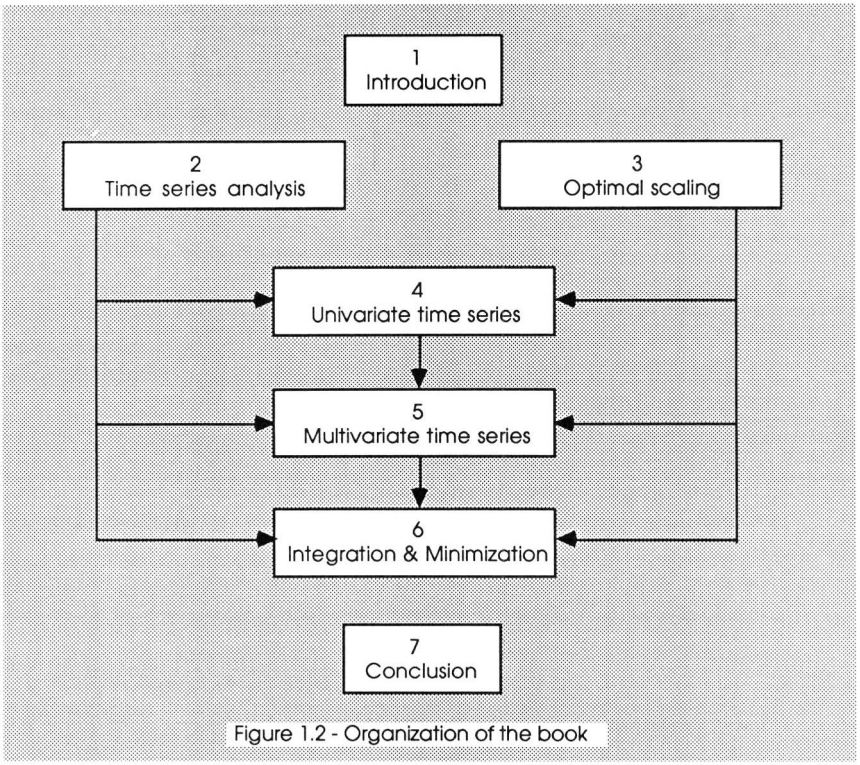


Figure 1.2 - Organization of the book

autoregressive models. Chapter 5 continues with four multivariate time series techniques: intervention analysis, the Box-Tiao transformation, dynamic components analysis and multiset dynamic components analysis. Chapter 6 unites the techniques into a single loss function and provides the solution of the minimization problem. We conclude by summarizing the main results. The APL computer program listed in the Appendix was used for all analyses.

CHAPTER 2

Multivariate time series analysis

Time series analysis is a huge and rapidly expanding field of research. Newbold (1981) calls it "one of the great academic growth industries". As a result, one can easily get lost in the maze of ARMA models, dynamic factors, state spaces and canonical reductions. In order to familiarize ourselves with the tools and goals of time series analysis this chapter outlines a number of basic concepts and major time series approaches. We discuss the backshift matrix, the concepts of autocorrelation and lagged variables and we introduce the correlation box and derive its relationship to least squares. After this, a selected number of time series models will be treated within a common framework based on finite sample matrix algebra of lagged variables. This use of matrix notation is somewhat unusual from a time series point of view since it may obscure the stochastic nature of many models, but on the other hand it greatly enhances the possibility to relate time series analysis to standard multivariate methods. The overview is not complete and is biased towards the developments in the social sciences. For a thorough survey of time series methods the reader is referred to a series of papers in the *International Statistical Review* by Makridakis (1976, 1978) and Newbold (1981, 1984, 1988). Shumway (1988) presents a nice introduction into the field.

2.1 A remark on time series analysis

One only needs to consult an arbitrary article from the *Journal of the American Statistical Society*, the *Journal of the Royal Statistical Society* or *Biometrika* to see what time series analysis is about: statistics, lots of statistics. Much of the current research effort is firmly rooted in the mathematical-statistical tradition. The average data analyst will be unable to appreciate these papers if (s)he does not have a thorough knowledge of at least the Gaussian Likelihood Function, the Akaike Information Criterium and of substantial parts of asymptotic distribution theory. Add to this the at times excessively complex notation and it is understandable that most time series research quietly proceeds its way without much impact on the social sciences.

A major problem in applying the classic approach to behavioral data is its insistence on stringent assumptions regarding the model. The basic material in the classic analysis is formed by a family of theoretical stochastic processes which specify the dependency structure among observations. From there on a model is generated and the data are fitted to see whether the model holds. In the multivariate case it is often assumed that also the relationships between different series are known in advance. While this classic approach may be fine for "hard" research like electronic engineering, it is unlikely to function properly in the theory-poor climate of social research. What we need here in the first place is a device to bring out the main time dependent features in the data. The question whether or not the data can be modelled after some mathematical representation comes in second place.

The methods discussed in this work focus on the fitting phase. Instead of starting from theory we start from the data. This explains why we exclusively use finite sample matrix algebra. Moreover, we will not assume the observations to be generated by a stochastic process. Indeed the observations need not be measured on an interval scale, but may consist of ordered or even unordered categories.

Time series analysis hardly knows a multivariate tradition which compares to the heritage found in cross-sectional analysis. For many years, techniques like principal components analysis, factor analysis, canonical correlation analysis and multidimensional scaling have been successfully applied to uncover relationships in cross-sectional data. Although most of these multivariate techniques have been formalized into a time series context, applications seem to be scarce. It is hoped that this work will contribute a little to restore the balance between theory- and data-oriented approaches.

2.2 Autocorrelation, backshift matrices and lagged variables

Autocorrelation is a crucial concept in time series analysis. The autocorrelation coefficient measures the dependency of observations over time. Let x_{t1} and x_{t2} be observations at time points t_1 and t_2 ($t_1, t_2 = 1...N$). The difference $s = t_1 - t_2$ is called the *lag* between the observations. If we define the mean of the series by

$$\underline{x} = N^{-1} \sum_{t=1}^{N} x_t \tag{2.1}$$

then the sample autocovariance at lag $s \geq 0$ of the series $x_1 ... x_N$ is

$$c_s = N^{-1} \sum_{t=1}^{N-s} (x_t - \underline{x})(x_{t+s} - \underline{x}). \tag{2.2}$$

The corresponding sample autocorrelation is given by $r_s = c_s / c_0$. Other formula for the autocovariance exist, but (2.2) has become standard. We refer to Jenkins and Watts (1968) and Anderson (1971, Ch. 6) for a discussion on the statistical properties of a number of autocovariance definitions. In the sequel, we assume that the series $x_1 ... x_N$ has zero mean.

The *backshift matrix* **B** has a cardinal function in translating the subscript notation in x_t into matrix notation. It is functionally equivalent to the backshift operator found in many time series books. The backshift

matrix is defined as the N × N matrix

$$\mathbf{B} = \begin{bmatrix} 000\ldots & 00 \\ 100 & 00 \\ 010 & \cdot \\ 001 & \cdot \\ \cdot\ \cdot & \cdot \\ \cdot & \\ \cdot & \\ 000 & 00 \\ 000\ldots & 10 \end{bmatrix} \tag{2.3}$$

Multiplying **B** by itself yields higher order backshift matrices. For example $\mathbf{B}_2 = \mathbf{BB}$ defines a second order backshift. The zero order backshift $\mathbf{B}_0$ is defined as the N × N identity matrix, and $\mathbf{B}'$ is the forward shifting matrix. If we collect the observations in the N × 1 column variable **x** then $\mathbf{B}_s\mathbf{x}$ is the s^{th} order *lagged variable* of **x**. Note that by this definition of **B** the first s observations in $\mathbf{B}_s\mathbf{x}$ become zero. Note also that since $\mathbf{B}_s'\mathbf{B}_s \neq \mathbf{I}$, i.e. shifting a series backwards s times followed by shifting it forward s times, does not produce the original series. Instead $\mathbf{B}_s'\mathbf{B}_s + \mathbf{B}_{N-s}\mathbf{B}_{N-s}' = \mathbf{I}$ makes up for the last s deleted observations. Other choices for the first row of **B** are also possible, but by defining **B** as in (2.3) we can write the sample autocovariance more concisely as

$$c_s = N^{-1}\,\mathbf{x}'\mathbf{B}_s\mathbf{x} \tag{2.4}$$

and the sample autocorrelation as

$$\begin{aligned} r_s &= c_s / c_0 \\ &= (\mathbf{x}'\mathbf{B}_s\mathbf{x}) / (\mathbf{x}'\mathbf{x}) \end{aligned} \tag{2.5}$$

The plot of r_s versus s, sometimes called a correlogram, plays an important role in time series analysis.

For a multiple series $\mathbf{x}_1 \ldots \mathbf{x}_j \ldots \mathbf{x}_M$ the autocorrelations can be computed in the same way as for the univariate case, but in addition there can be cross-correlations and cross-autocorrelations among the M series. Let

the observations be collected into an N × M matrix **X**, in which the t^{th} row contains the M measurements at time t for t = 1...N, and in which the j^{th} column corresponds to the j^{th} variable for j = 1...M. The matrix of cross-covariances among the M series can be found by

$$\mathbf{C}_0 = N^{-1}\,\mathbf{X}'\mathbf{X}, \tag{2.6}$$

and if we define $\mathbf{D} = dg\,\mathbf{X}'\mathbf{X}$ the matrix of cross-correlations is (see section 2.3 for a definition of the dg() operator)

$$\mathbf{R}_0 = \mathbf{D}^{-1/2}\mathbf{X}'\mathbf{X}\mathbf{D}^{-1/2}. \tag{2.7}$$

Similarly we find for the cross-autocovariances and cross-autocorrelations at lag s

$$\mathbf{C}_s = N^{-1}\,\mathbf{X}'\mathbf{B}_s\mathbf{X}. \tag{2.8}$$
$$\mathbf{R}_s = \mathbf{D}^{-1/2}\mathbf{X}'\mathbf{B}_s\mathbf{X}\mathbf{D}^{-1/2}. \tag{2.9}$$

The diagonal elements of $\mathbf{R}_s$ are equal to the univariate autocorrelations given by (2.5) and the off-diagonal elements are cross-autocorrelations that measure the relationship between two series at different lags. Section 2.4 describes a way to organize successive matrices $\mathbf{R}_s$ into a correlation box.

It is sometimes of interest to obtain the difference $\Delta x_t = x_t - x_{t-1}$. In matrix notation the first difference can be written as $\Delta\mathbf{X} = \mathbf{X} - \mathbf{B}\mathbf{X}$, the second difference $\Delta^2\mathbf{X} = (\mathbf{X} - \mathbf{B}\mathbf{X}) - \mathbf{B}(\mathbf{X} - \mathbf{B}\mathbf{X}) = \mathbf{X} - 2\mathbf{B}\mathbf{X} + \mathbf{B}_2\mathbf{X} = (\mathbf{I} - \mathbf{B})^2\mathbf{X}$. The d^{th} difference is equal to $\Delta^d\mathbf{X} = (\mathbf{I} - \mathbf{B})^d\mathbf{X}$.

Since autocorrelations and cross-autocorrelations are all based on the inproducts of lagged variables the time dependent information is aggregated over all N observations. A complementary way of time series analysis is not to aggregrate over time points, but over series. This approach is taken by Ramsay (1982). We will however restrict ourselves to the first approach.

2.3 Some scalar matrix operators

For any square matrix $\mathbf{A} = (a_{ij})$ of order $N \times N$ we define dg **A** or dg(**A**) as a diagonal matrix of the diagonal elements of **A**, i.e.

$$\mathrm{dg}\mathbf{A} = \begin{bmatrix} a_{11} & & & 0 \\ & a_{22} & & \\ & & a_{33} & \\ 0 & & & a_{NN} \end{bmatrix} \tag{2.10}$$

The trace of **A**, denoted by tr **A** or tr(**A**) is equal to the sum of the diagonal elements of **A**, i.e.

$$\mathrm{tr}\,\mathbf{A} = \sum_{i=1}^{N} a_{ii}. \tag{2.11}$$

The largest eigenvalue of **A** will be denoted by $\lambda_{max}(\mathbf{A})$.

For two real valued matrices **Z** and **X**, both of order $N \times R$, we define the sums of squares function ssq (**Z**, **X**) as

$$\mathrm{ssq}\,(\mathbf{Z}, \mathbf{X}) = \mathrm{tr}\,(\mathbf{Z} - \mathbf{X})'(\mathbf{Z} - \mathbf{X}). \tag{2.12}$$

The ssq () function posseses the following properties:

$$\mathrm{ssq}\,(\mathbf{Z}, \mathbf{X}) \geq 0, \text{ with equality if and only if } \mathbf{Z} = \mathbf{X}, \tag{2.13a}$$

$$\mathrm{ssq}\,(\mathbf{Z}, \mathbf{X}) = \mathrm{ssq}\,(\mathbf{X}, \mathbf{Z}) = \mathrm{ssq}\,(\mathbf{Z}', \mathbf{X}') = \mathrm{ssq}\,(\mathbf{X}', \mathbf{Z}'). \tag{2.13b}$$

In addition, for conformable matrices $\mathbf{Z}_1$, $\mathbf{Z}_2$, $\mathbf{X}_1$ and $\mathbf{X}_2$ it may be split over columns, respectively rows, or both, by

$$\mathrm{ssq}\,([\mathbf{Z}_1, \mathbf{Z}_2], [\mathbf{X}_1, \mathbf{X}_2]) = \mathrm{ssq}\,(\mathbf{Z}_1, \mathbf{X}_1) + \mathrm{ssq}\,(\mathbf{Z}_2, \mathbf{X}_2) \tag{2.13c}$$

$$\mathrm{ssq}\,([\mathbf{Z}_1', \mathbf{Z}_2']', [\mathbf{X}_1', \mathbf{X}_2']') = \mathrm{ssq}\,(\mathbf{Z}_1, \mathbf{X}_1) + \mathrm{ssq}\,(\mathbf{Z}_2, \mathbf{X}_2). \tag{2.13d}$$

2.4 The correlation box

A particularly convenient way to represent the types of correlations we have seen in section 2.2 is to arrange them into a three-dimensional *correlation box*. The correlation box can be sliced in several directions in order to obtain a number of useful submatrices and subvectors.

Let $\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_j \ldots \mathbf{x}_M]$ be an N × M matrix of observations on N consecutive time points on M variables and let $\mathbf{D} = \text{dg}\,\mathbf{X}'\mathbf{X}$. Then $\mathbf{R}_s = \mathbf{D}^{-1/2}\mathbf{X}'\mathbf{B}_s\mathbf{X}\mathbf{D}^{-1/2}$ is an M × M matrix of sample correlations at a lag s (s = 0,1...P). If we arrange the P + 1 correlation matrices $\mathbf{R}_0 \ldots \mathbf{R}_P$ after one another we obtain the correlation box **R**. The correlation box is a three-dimensional structure of order (P + 1) × M × M and is an example of a three-way three-mode data block. Although each correlation matrix $\mathbf{R}_s$ is square, it seems more appropriate to label it as two-mode rather than one-mode since both ways correspond to different entities for s ≠ 0.

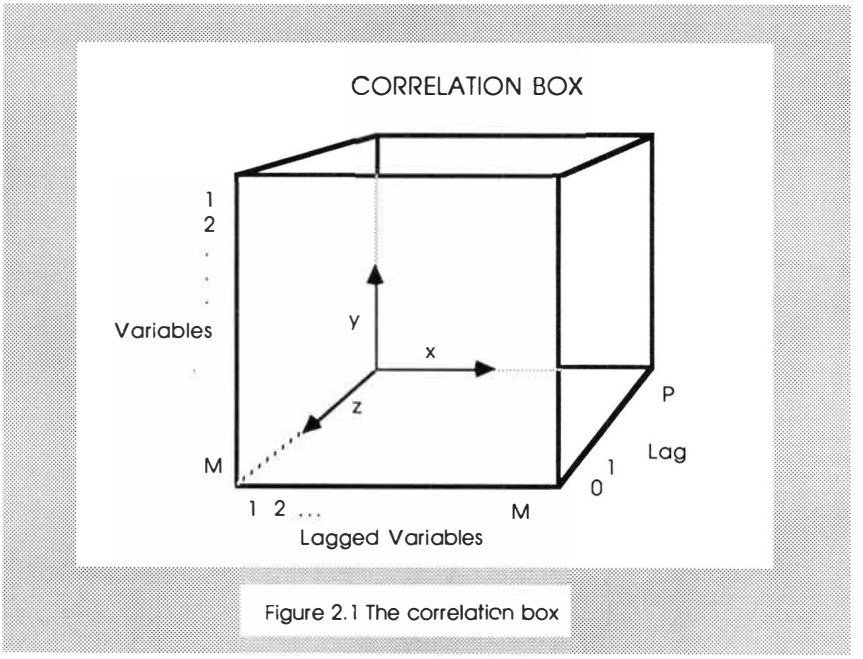


Figure 2.1 The correlation box

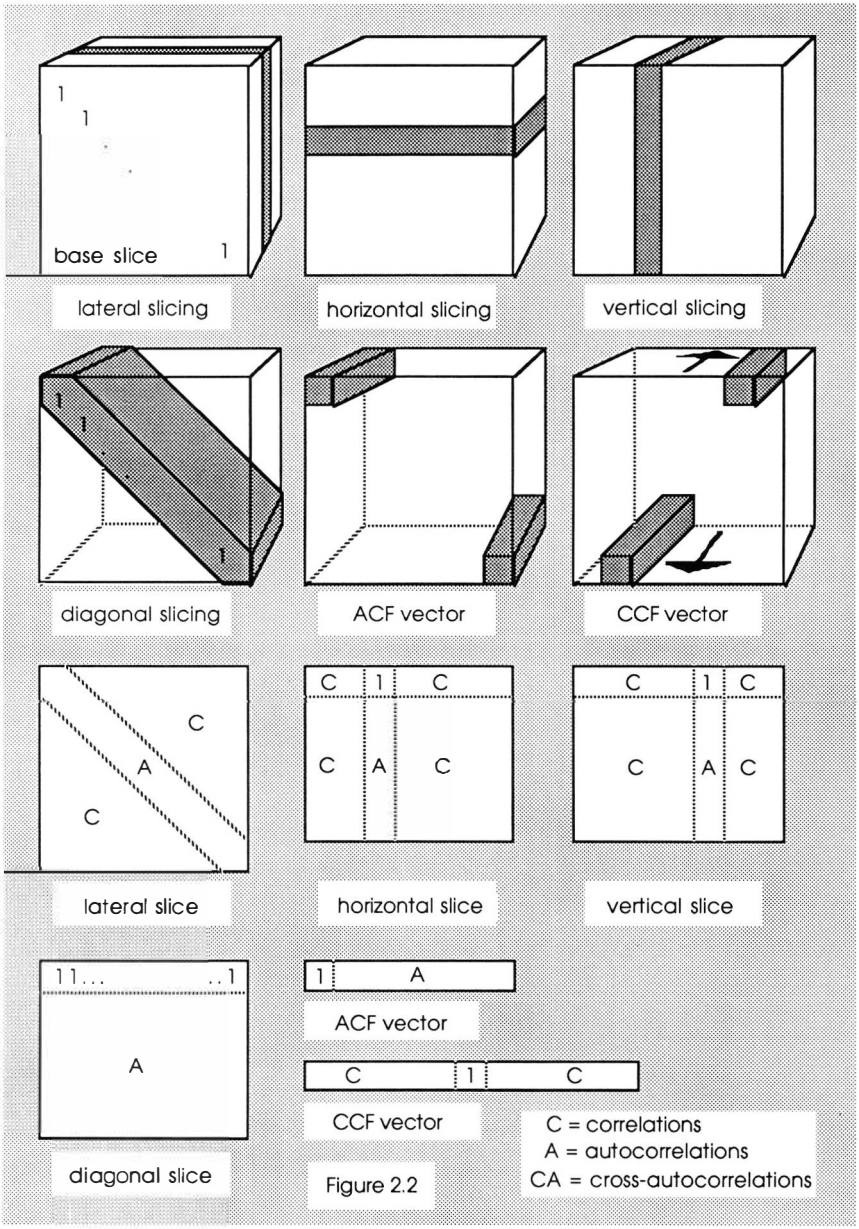

1
1
base slice
1
lateral slicing
horizontal slicing
vertical slicing
1
1
1
diagonal slicing
ACF vector
CCF vector
C
A
C
lateral slice
C 1 C
C A C
horizontal slice
C 1 C
C A C
vertical slice
11... ..1
A
diagonal slice
1 A
ACF vector
C 1 C
CCF vector
C = correlations
A = autocorrelations
CA = cross-autocorrelations

Figure 2.2

Figure 2.1 illustrates the correlation box. There are a number of ways to look at the correlation box. Submatrices and subvectors can be selected by slicing and tubing the box in appropriate directions. Below we discuss lateral, horizontal, vertical and diagonal slicing, and lateral tubing. See Figure 2.2 for a graphic representation of these operations.

Lateral slicing refers to partitioning the correlation box parallel to the x-y plane, which corresponds to breaking the box apart in the way we constructed it. Viewing the correlation box from some point in the x-y plane, we see that it is made up of P + 1 correlation slices $\mathbf{R}_s$ for s = 0...P. The slice $\mathbf{R}_s$ summarizes all sample second moment information at some time distance s. Each slice $\mathbf{R}_s$ is of order M × M and its jj'th entry $r_{sjj'}$ corresponds to the cross-autocorrelation between series j and j' at lag s, where j is the leading series (j, j' = 0...M). Since $r_{sjj'} \neq r_{sj'j}$ except for s = 0, $\mathbf{R}_s$ is in general not symmetric. A very special slice is the base slice $\mathbf{R}_0$. This is the conventional correlation matrix, the cornerstone of sixty years of multivariate analysis. Viewed in terms of the correlation box, the classic statistical "independence of observations" assumption comes down to the expectation that any slice beyond the base contains negligible entries, so that the base slice provides an adequate summary of the relationships between the variables. If in addition, the off-diagonal elements of the base slice are negligible we obtain a white noise box. In this case the underlying series is said to form a multivariate white noise series, i.e. all variables are independent within and across time.

If we shift our point of view into the x-z plane, we see that the correlation box may be sliced into M matrices $\mathbf{R}_{.j}$, each of order (P + 1) × M. (The dot index signals that we select all slices of the corresponding way. Any trailing dot indices will be left out.) This is what we call horizontal slicing since the box is sliced horizontally here. The slice $\mathbf{R}_{.j}$ contains the cross-autocorrelations of all M variables with the target variable j. We use the convention that $\mathbf{R}_{.j}$ begins to count a lag zero and that its values are organized such that row zero is equal to the jth row of $\mathbf{R}_0$. Then the null row of $\mathbf{R}_{.j}$ lists the correlations between the target and the M variables, and its jth column lists the autocorrelations of the target

variable. The remaining entries are cross-autocorrelations, with the (lagging) target. Studying the horizontal slices is important in situations where we want to determine which past and/or present variables influence the present target variable. We should then look for values of substantial magnitude. Translated into a regression context, the target plays the role of the dependent variable.

The complement of horizontal slicing is vertical slicing. This is carried out in the y-z plane. By slicing the box vertically we also obtain M matrices, each of order $(P + 1) \times M$, which we label $\mathbf{R}_{..j}$ $(j = 1...M)$. The structure of the vertical slice $\mathbf{R}_{..j}$ is identical to that of the horizontal slice $\mathbf{R}_{.j}$, but the crucial difference between them is that in the vertical slice the target variable leads the other variables, while the opposite is true for the horizontal slice. Thus, the vertical slice provides useful information if we want to determine which variables are being influenced by the target. The target then plays the role of an independent variable, or predictor.

Another useful slice is the diagonal slice $\mathbf{R}_D$ of order $(P + 1) \times M$. Its rows are equal to the diagonals of the successive lateral slices $\mathbf{R}_0...\mathbf{R}_M$. The j^{th} column of $\mathbf{R}_D$ thus contains the autocorrelations of the j^{th} variable. If the entries of the correlation box that are not located on $\mathbf{R}_D$ are all zero, the M variables are independent (within and across time) of each other. It is sometimes of interest to compress the information of some arbitrary correlation box into a new box in which all nonzero elements are located on the diagonal slice. This results in M new series that are autocorrelated but also mutually independent within and across time.

An operation related to slicing is tubing. In slicing, we select a matrix from the box. In tubing, we take out a vector. Suppose we start from some element $r_{0jj'}$ of the base slice and travel backwards along the z-axis, then we pass the sample cross-autocorrelation coefficients that relate variable j to j'. We call this lateral tubing. If $j = j'$ the vector $\mathbf{r}_{jj'}$ will hold the sample autocorrelation function (abbreviated as "sample ACF" or just "ACF"), in other cases it contains $P + 1$ sample cross-autocorrelation

functions ("sample CCF"). The CCF sample values between j and j' for negative lags can be found by travelling from element $r_{Pj'j}$ to $r_{0j'j}$, i.e. we go forwards along the z-axis. Both ACF and CCF can be plotted against their lag values. The plot of the autocorrelations is an important diagnostic tool. The CCF plays an important role in studying bivariate relationships among time series. Horizontal and vertical tubing do not appear to be useful operations since their results depend on the sequence of variables.

To illustrate some slicing operations we use a five-variate time series from Shumway (1988). The data record the mortality rate and four environmental variables (smoke, sulfur dioxide, temperature and relative humidity) in the city of London during the winter of 1958 for a period of 102 consecutive days. Figure 2.3 depicts the five series against time. The effect of a flu epidemic at the end of the mortality series was filtered out (cf. Shumway, 1988, 33) so the numbers as given can be interpreted as excess deaths over a given baseline level.

The objective of the study was to determine which, if any, of the pollution or weather factors might be most strongly associated with elevated levels of mortality. Therefore it is interesting to focus on the mortality horizontal slice of the correlation box (with P chosen as P = 24). This slice contains correlations between all (lagged) series and mortality. It is listed in Table 2.1. The first few rows, which correspond to the low lag numbers, are relatively large compared to the remainder of the slice, indicating that the most important predictors of mortality are located within a, say, 10 day lag.

Figure 2.4 shows three of the correlation series (mortality, smoke and temperature) plotted against their lags. The ACF mortality and the CCF smoke-mortality series both start with high correlations near the beginning of the plot and fluctuate more or less random after lag 10. The CCF smoke-mortality values for lags 0 to 2, respectively 0.53, 0.25 and 0.20[1], reflect a positive relationship between the present mortality rate and present and past amount of smoke. The more smoke, the higher

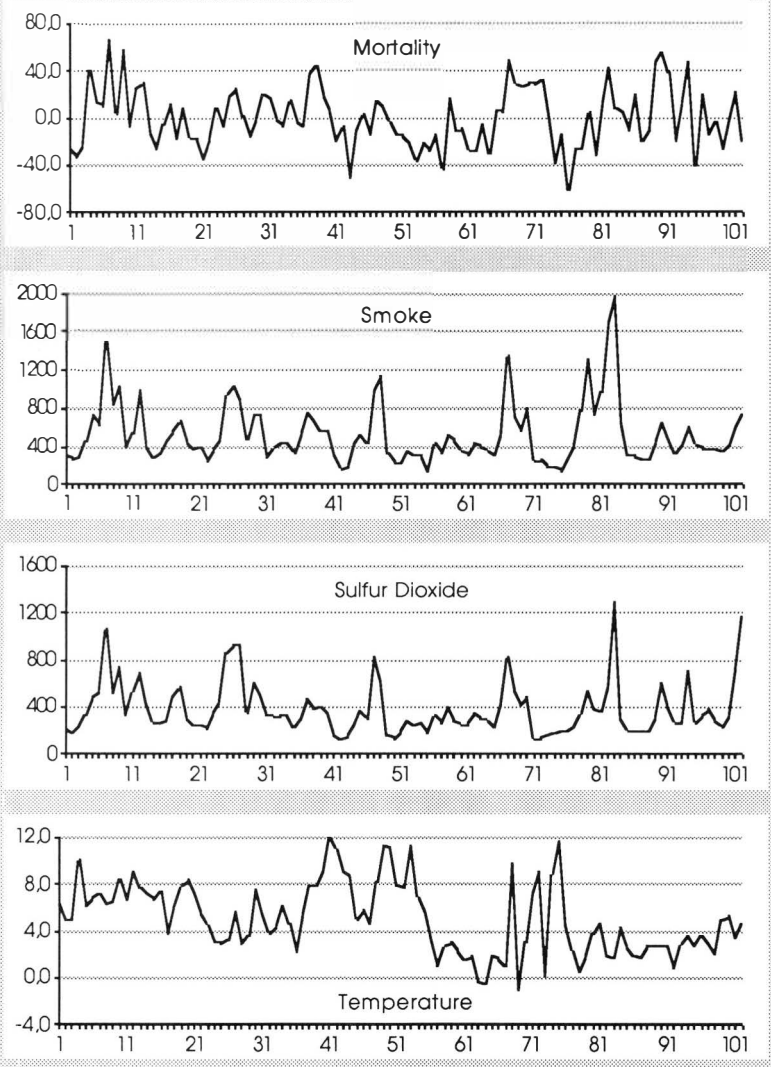


Figure 2.3 London Mortality Data (Continued on the next page)

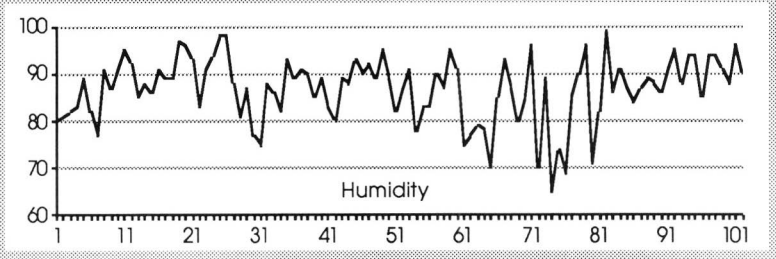


Figure 2.3 London Mortality Data (Continued)

the mortality will be now, tomorrow and the day after tomorrow. In contrast, the cross correlation between mortality and temperature is near zero, so there seems to be no relationship between temperature of the day and the number of deaths. However the lag-2 cross autocorrelation is -0.30, which reflect that a low temperature today may be related to a high mortality rate two days after now, and vica versa, so temperature seems to have a delayed effect on mortality.

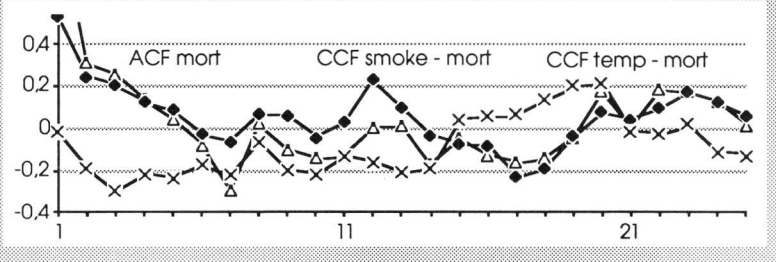


Figure 2.4 ACF and CCF Mortality Data

Note: Bartlett (1946) established that the sample ACF of a white noise process is approximately normal with mean 0 and variance 1/N. We may use this result as a rough check to infer that r_{sjj} is effectively zero if $-z\alpha/2 < r_{sjj} < z\alpha/2$ given some Type I error probability α. In the above case, for N = 102 and α = 0.05 the interval is roughly $-0.20 < r_{sjj} < 0.20$. This interval may be used also for the CCF series.

We may also be interested in studying the reverse, i.e. the effect of temperature changes on the other series. The temperature vertical slice provides useful information for this purpose. Of course we will then also find the two-day relationship between temperature and mortality, but also a negative relationship of temperature versus smoke and sulfur dioxide levels at a one-day lag. Thus, if temperature increases, the amount of smog and sulfur tends to decrease the next day. This kind of information could be valuable for those with throat or lung affections.

TABLE 2.1 Horizontal mortality slice (London data)

Lag	Mortality	Smoke	Sulfur	Temperature	Humidity
0	1.00	0.53	0.49	-0.02	0.19
1	0.31	0.25	0.18	-0.19	0.11
2	0.26	0.20	0.16	-0.30	0.08
3	0.13	0.13	0.11	-0.21	-0.13
4	0.04	0.09	0.07	-0.24	-0.04
5	-0.08	-0.03	-0.02	-0.17	-0.17
6	-0.30	-0.06	-0.03	-0.22	-0.17
7	0.02	0.07	0.05	-0.06	-0.16
8	-0.10	0.06	-0.01	-0.20	-0.15
9	-0.14	-0.04	-0.09	-0.22	-0.11
10	-0.13	0.03	-0.03	-0.13	-0.09
11	0.00	0.23	0.21	-0.16	0.14
12	0.01	0.10	0.04	-0.21	0.24
13	-0.17	-0.03	0.05	-0.19	0.11
14	-0.05	-0.08	-0.05	0.04	-0.02
15	-0.13	-0.08	-0.06	0.06	0.05
16	-0.16	-0.23	-0.18	0.07	-0.15
17	-0.14	-0.19	-0.17	0.14	-0.23
18	-0.04	-0.04	-0.04	0.20	-0.05
19	0.17	0.08	0.04	0.22	0.05
20	0.01	0.04	0.09	-0.01	-0.01
21	0.18	0.10	0.11	-0.03	0.03
22	0.17	0.17	0.13	0.02	0.12
23	0.13	0.13	0.09	-0.11	0.23
24	0.02	0.06	0.05	-0.13	-0.06

2.5 The correlation box and its relationship to least squares

This section shows how entries from the correlation box can be decomposed into independent components. The decomposition aids to clarify the properties of many least squares functions considered later.

We start with the bivariate case. Let $\mathbf{z}$ and $\mathbf{x}$ both represent $N \times 1$ vectors with zero mean and let $d_z = \mathbf{z}'\mathbf{z}$ and $d_x = \mathbf{x}'\mathbf{x}$ denote the sums of squares. The cross-autocorrelation between $\mathbf{z}$ and $\mathbf{x}$ at lag s for s = 0...N can be written as

$$r_{szx} = d_z^{-1/2}\mathbf{z}'\mathbf{B}_s\mathbf{x}\, d_x^{-1/2}, \tag{2.14}$$

where $\mathbf{B}_s$ is the s^{th} order backshift matrix. Below we show that r_{szx} can be expressed as a linear function of d_z, d_x, $(z_i - x_{i-s})^2$ and a usually small end correction term.

Theorem 2.1 (correlation partitioning):

Let $\mathbf{z}$ and $\mathbf{x}$ be any $N \times 1$ vector with zero mean, then r_{szx} can be decomposed as

$$r_{szx} = 0.5\, d_z^{-1/2} d_x^{-1/2} [d_z + d_x - (\mathbf{z} - \mathbf{B}_s\mathbf{x})'(\mathbf{z} - \mathbf{B}_s\mathbf{x}) - e_{sx}] \tag{2.15}$$

for s = 0...N and where $e_{sx} = \mathbf{x}'\mathbf{B}_{N-s}\,\mathbf{B}_{N-s}'\mathbf{x}$.

Proof:
It is not difficult to verify that for any N-vector $\mathbf{z}$ and $\mathbf{x}$ and for all s = 0...N

$$\begin{aligned} \mathbf{x}'\mathbf{x} &= \mathbf{x}'\mathbf{B}_s'\mathbf{B}_s\mathbf{x} + \mathbf{x}'\mathbf{B}_{N-s}\,\mathbf{B}_{N-s}'\mathbf{x} \\ &= \mathbf{x}'\mathbf{B}_s'\mathbf{B}_s\mathbf{x} + e_{sx} \end{aligned} \tag{2.16}$$

and

$$\mathbf{z}'\mathbf{B}_s\mathbf{x} = \mathbf{x}'\mathbf{B}_s'\mathbf{z} \tag{2.17}$$

hold. By substitution of (2.16) and (2.17) we obtain

$$\begin{aligned} r_{szx} &= d_z^{-1/2} d_x^{-1/2}\, \mathbf{z}'\mathbf{B}_s\mathbf{x} \\ &= -0.5\, d_z^{-1/2} d_x^{-1/2} [-2\mathbf{z}'\mathbf{B}_s\mathbf{x} + \mathbf{z}'\mathbf{z} + \mathbf{x}'\mathbf{x} - \mathbf{z}'\mathbf{z} - \mathbf{x}'\mathbf{x}] \\ &= -0.5\, d_z^{-1/2} d_x^{-1/2} [-d_z - d_x + \mathbf{z}'\mathbf{z} - 2\mathbf{z}'\mathbf{B}_s\mathbf{x} + \mathbf{x}'\mathbf{B}_s'\mathbf{B}_s\mathbf{x} + \mathbf{x}'\mathbf{B}_{N-s}\mathbf{B}_{N-s}'\mathbf{x}] \\ &= 0.5\, d_z^{-1/2} d_x^{-1/2} [d_z + d_x - (\mathbf{z} - \mathbf{B}_s\mathbf{x})'(\mathbf{z} - \mathbf{B}_s\mathbf{x}) - e_{sx}], \end{aligned}$$

which is equivalent to (2.15). ❑

The end correction $e_{sx} = \mathbf{x}'\mathbf{B}_{N-s}\mathbf{B}_{N-s}'\mathbf{x}$ is equal to the sum of squares of the last s elements of $\mathbf{x}$. These elements were lost at the end of the array as a result of the shifting process. The end term is usually quite small compared to the other terms, especially for low s and large N. Therefore we will often ignore it. The correction is a direct consequence from the fact that we are dealing with finite sample sizes only.

The interesting component is the middle term $(\mathbf{z} - \mathbf{B}_s\mathbf{x})'(\mathbf{z} - \mathbf{B}_s\mathbf{x})$. This term expresses the correlation as a function of squared differences between $\mathbf{z}$ and $\mathbf{B}_s\mathbf{x}$. In Chapters 4 and 5 we will frequently obtain linear and/or nonlinear transformations of the variables that minimize terms like the middle term. If we rearrange the decomposition into

$$(\mathbf{z} - \mathbf{B}_s\mathbf{x})'(\mathbf{z} - \mathbf{B}_s\mathbf{x}) = d_z + d_x - 2\, d_z^{1/2} d_x^{1/2} r_{szx} - e_{sx}, \tag{2.18}$$

we see what happens then: provided that the variances of $\mathbf{z}$ and $\mathbf{x}$ are kept constant, minimizing $(\mathbf{z} - \mathbf{B}_s\mathbf{x})'(\mathbf{z} - \mathbf{B}_s\mathbf{x})$ maximizes the correlation r_{szx}. This relationship is an important guide in studying properties of optimal nonlinear transformations.

If the variables $\mathbf{z}$ and $\mathbf{x}$ have been scaled such that $\mathbf{z}'\mathbf{z} = \mathbf{x}'\mathbf{x} = 1$, the formulas simplify considerably. The decompositions then reduce to

$$r_{szx} = 1 - 0.5\, (\mathbf{z} - \mathbf{B}_s\mathbf{x})'(\mathbf{z} - \mathbf{B}_s\mathbf{x}) - 0.5\, e_{sx}, \tag{2.19}$$

$$(\mathbf{z} - \mathbf{B}_s\mathbf{x})'(\mathbf{z} - \mathbf{B}_s\mathbf{x}) = 2\,(1 - r_{rzx}) - e_{sx}. \tag{2.20}$$

It is straightforward to generalize the decomposition to the multivariate case. Let $\mathbf{X}$ be an N × M matrix composed of M variables, each with zero mean, and let $\mathbf{D} = \text{dg } \mathbf{X}'\mathbf{X}$. We assume that the diagonal elements of $\mathbf{D}$ are all nonzero so that it has an inverse. Let $\mathbf{R}_s = \mathbf{D}^{-1}\mathbf{X}'\mathbf{B}_s\mathbf{X}$ denote the lateral slice at lag s. The diagonal of $\mathbf{R}_s$ can be decomposed as

$$\text{dg } \mathbf{R}_s = \text{dg } 0.5\ \mathbf{D}^{-1}\ [2\mathbf{D} - (\mathbf{X} - \mathbf{B}_s\mathbf{X})'(\mathbf{X} - \mathbf{B}_s\mathbf{X}) - \mathbf{E}] \tag{2.21}$$

with $\mathbf{E} = \mathbf{X}'\mathbf{B}_{N-s}\mathbf{B}'_{N-s}\mathbf{X}$. The proof runs identical to the bivariate case. The matrix of sums of squared differences can be partitioned as

$$\text{dg } (\mathbf{X} - \mathbf{B}_s\mathbf{X})'(\mathbf{X} - \mathbf{B}_s\mathbf{X}) = \text{dg } [2\mathbf{D}\ (\mathbf{I} - \mathbf{R}_s) - \mathbf{E}]. \tag{2.22}$$

Decomposition (2.22) provides a direct relationship between the diagonal elements of the base slice $\mathbf{R}_0$ and the lateral slice $\mathbf{R}_s$ in terms of squared differences of the underlying variables.

Suppose that we minimize ssq $(\mathbf{X}, \mathbf{B}_s\mathbf{X})$ over (non)linear transforms of $\mathbf{X}$, then this is the same as maximizing the sum of the M s^{th} order autocorrelations. More specifically,

$$\text{ssq } (\mathbf{X}, \mathbf{B}_s\mathbf{X}) = 2\ \text{tr } (\mathbf{I} - \mathbf{R}_s) - \text{tr } \mathbf{E}, \tag{2.23}$$

provided that we normalize $\mathbf{X}$ with $\text{dg } \mathbf{X}'\mathbf{X} = \mathbf{I}$.

For the sake of completeness, if we have two sets of M variables being collected in matrices $\mathbf{Z}$ and $\mathbf{X}$ both of order N × M, off-diagonal elements of $\mathbf{R}_s$ may also be partitioned. The partial cross-autocorrelation matrix is equal to $\mathbf{R}_{sZX} = \mathbf{D}_Z^{-1/2}\mathbf{Z}'\mathbf{B}_s\mathbf{X}\mathbf{D}_X^{-1/2}$ and the decompositions are

$$\text{dg } \mathbf{R}_{sZX} = \text{dg } 0.5\ \mathbf{D}_Z^{-1/2}\ [\mathbf{D}_Z + \mathbf{D}_X - (\mathbf{Z} - \mathbf{B}_s\mathbf{X})'(\mathbf{Z} - \mathbf{B}_s\mathbf{X}) - \mathbf{E}_X]\ \mathbf{D}_X^{-1/2} \tag{2.24}$$

$$\text{dg } (\mathbf{Z} - \mathbf{B}_s\mathbf{X})'(\mathbf{Z} - \mathbf{B}_s\mathbf{X}) = \text{dg } [\mathbf{D}_Z + \mathbf{D}_X - 2\ \mathbf{D}_Z^{1/2}\mathbf{R}_{sZX}\mathbf{D}_X^{1/2} - \mathbf{E}_X], \tag{2.25}$$

where

$$\mathbf{D}_Z = \mathrm{dg}\,\mathbf{Z}'\mathbf{Z}$$
$$\mathbf{D}_X = \mathrm{dg}\,\mathbf{X}'\mathbf{X}$$
$$\mathbf{E}_X = \mathbf{X}'\mathbf{B}_{N-s}\mathbf{B}'_{N-s}\mathbf{X}.$$

If $\mathbf{Z} = \mathbf{X}$ these expressions reduce to those given before. If the variances have been normalized according to $\mathbf{D}_Z = \mathbf{D}_X = \mathbf{I}$, then the least squares function ssq $(\mathbf{Z}, \mathbf{B}_s\mathbf{X})$ may be written as

$$\mathrm{ssq}\,(\mathbf{Z}, \mathbf{B}_s\mathbf{X}) = 2\,\mathrm{tr}\,(\mathbf{I} - \mathbf{R}_{sZX}) - \mathrm{tr}\,\mathbf{E}_X, \tag{2.26}$$

so minimizing it will maximize the sum of the M correlations between $\mathbf{Z}$ and $\mathbf{B}_s\mathbf{X}$.

2.6 ARMA models

In many disciplines, autoregressive moving average models (ARMA models) have become almost synonymous to time series analysis. Based on the theoretical work of Wold (1938) and others, Box and Jenkins (1976) established an impressive time series methodology. The Box-Jenkins approach is the major time series methodology in many scientific disciplines, not in the least for the fact that many univariate series can be described by a parsimonious ARMA model. Substantial parts of the books by MacCleary and Hay (1980), Gottman (1981) and Gregson (1983) on time series analysis in the social sciences are devoted to ARMA modelling. A good introduction into ARMA models is Cryer (1986). An instructive example of ARMA modelling in the social sciences is provided by Vidgerhous (1977).

Multivariate generalizations of the ARMA model (sometimes called VARMA or MARMA models) have been discussed by Whittle (1953), Quenouille (1957), Hannan (1970) and Box and Jenkins (1976). Let $\mathbf{x}_t$ be an $M \times 1$ vector of observations at time t for t = 1...N. The stationary multivariate ARMA(P,Q) model can be written as

$$\mathbf{x}_t + \mathbf{A}_1'\mathbf{x}_{t-1} + \ldots + \mathbf{A}_P'\mathbf{x}_{t-P} = \mathbf{z}_t + \mathbf{F}_1'\mathbf{z}_{t-1} + \ldots + \mathbf{F}_Q'\mathbf{z}_{t-Q} \tag{2.27}$$

or

$$\sum_{p=0}^{P} \mathbf{A}_p'\mathbf{x}_{t-p} = \sum_{q=0}^{Q} \mathbf{F}_q'\mathbf{z}_{t-q} \qquad (t = 1\ldots N) \tag{2.28}$$

Vector $\mathbf{z}_t$ represents an M-dimensional unobserved white noise process with zero mean and finite covariance, and $\mathbf{A}_p$ and $\mathbf{F}_q$ are parameter matrices both of order $M \times M$. Equation (2.27) becomes (2.28) if

$$\mathbf{A}_0 = \mathbf{F}_0 = \mathbf{I}_M. \tag{2.29}$$

In order to identify the model it is often assumed that $\mathbf{A}_p$ and $\mathbf{F}_q$ satisfy certain stationarity and invertibility conditions. See Box and Jenkins (1976) for more details.

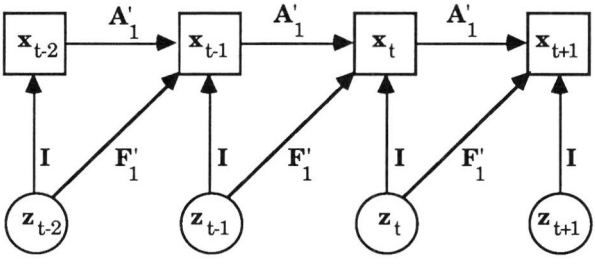


Figure 2.5 Scheme of an ARMA(1,1) model

Figure 2.5 is a schematic representation of an ARMA(P,Q) model where P = 1 and Q = 1. The idea is that $\mathbf{x}_t$ is generated by a linear combination of

- P past values $\mathbf{x}_{t-1}\ldots\mathbf{x}_{t-P}$
- one present and Q past values of random shocks: $\mathbf{z}_t$, $\mathbf{z}_{t-1}\ldots\mathbf{z}_{t-Q}$.

The linear combination is sometimes called the filter.

If we arrange $\mathbf{x}_t$ and $\mathbf{z}_t$ into the matrices $\mathbf{X}$ and $\mathbf{Z}$, both of order $N \times M$, then each row corresponds to a time point. We can write (2.28) more concisely as

$$\sum_{p=0}^{P} \mathbf{B}_p \mathbf{X} \mathbf{A}_p = \sum_{q=0}^{Q} \mathbf{B}_q \mathbf{Z} \mathbf{F}_q. \tag{2.30}$$

For fitting data purposes, (2.30) suggests the least squares loss function

$$\text{ssq} \left(\sum_{p=0}^{P} \mathbf{B}_p \mathbf{X} \mathbf{A}_p, \sum_{q=0}^{Q} \mathbf{B}_q \mathbf{Z} \mathbf{F}_q \right), \tag{2.31}$$

for which we must obtain the minimum over $\mathbf{A}_1 \ldots \mathbf{A}_P$, $\mathbf{F}_1 \ldots \mathbf{F}_Q$ and $\mathbf{Z}$.

The ARMA model is based on the assumption that each column of $\mathbf{Z}$, denoted by $\mathbf{z}$ for a moment, is generated by a white noise process, i.e.

$$E(z_t) = 0 \qquad t = 1 \ldots N \tag{2.32a}$$

$$E(z_t z_t) = \omega^2 \qquad t = 1 \ldots N \tag{2.32b}$$

$$E(z_t z_s) = 0 \qquad t \neq s \text{ and } s,t = 1 \ldots N \tag{2.33}$$

where E is the expectation operator that denotes the expected value over all possible realizations. Moreover, if $P > 0$ then x_{t-p} will be uncorrelated with z_t (Box & Jenkins, 1976, 75), i.e.

$$E(x_{t-p} z_t) = 0 \qquad p = 1 \ldots P \text{ and } t = 1 \ldots N. \tag{2.34}$$

Transforming the population white noise properties (2.33) and (2.34) to sample quantities, we expect $\mathbf{Z}$ to be orthogonal to both $\mathbf{B}_q\mathbf{Z}$ and $\mathbf{B}_p\mathbf{X}$, i.e.

$$\mathbf{Z}'\mathbf{B}_q\mathbf{Z} = \mathbf{0} \qquad q = 1 \ldots Q \tag{2.35}$$

$$\mathbf{Z}'\mathbf{B}_p\mathbf{X} = \mathbf{0} \qquad p = 1 \ldots P \tag{2.36}$$

These two conditions correspond to equations (1.4) and (1.5) of Spliid (1983). We will come back to the ARMA problem in Chapter 6.

Specifying a "correct" model from the class of multivariate ARMA class models is much more complicated than in the univariate case due to the increased number of parameters. The field of multivariate model specification has been quite active. See for example Granger and Newbold (1977), Jenkins and Alavi (1981), Tiao and Box (1981), Tiao and Tsay (1983), Tsay and Tiao (1985). In spite of this, the task is yet far from standard and not very many applications have seen the light yet.

2.7 State space models

Besides ARMA, a second important class of models for multivariate time series are the so called state space models. The state space model has been highly successful in aerospace engineering and control applications in the 1960's, for example in the NASA Apollo project for tracking the spacecraft. Since then it has rapidly migrated to other scientific disciplines as communications theory, economics, hydrology and weather forecasting. The following terms all refer to more or less the same field: state space models, Kalman filtering, optimal control, recursive estimation, random coefficients regression models and linear dynamic systems. Introductions into state space modelling are Gelb (1974), Willems (1978), Bennett (1979), O'Connell (1984), Aoki (1987) and Caines (1987).

Suppose that $\mathbf{x}_t$ is an $M \times 1$ vector of observations at time point $t = 1...N$ and that this vector can be partitioned into an $M_1 \times 1$ subvector $\mathbf{x}_{1,t}$ of system inputs (= independent or exogenous variables) and an $M_2 \times 1$ subvector $\mathbf{x}_{2,t}$ of system outputs (= dependent or endogenous variables) with $M = M_1 + M_2$. It is assumed that $\mathbf{x}_{1,t}$ transforms to $\mathbf{x}_{2,t}$ mediated by an unobserved $R \times 1$ state vector $\mathbf{z}_t$ by the linear state space model

$$\mathbf{z}_t = \mathbf{F}'\mathbf{z}_{t-1} + \mathbf{G}'\mathbf{x}_{1,t} + \mathbf{w}_t \tag{2.37}$$

$$\mathbf{x}_{2,t} = \mathbf{H}'\mathbf{z}_t + \mathbf{v}_t \tag{2.38}$$

The matrices $\mathbf{F}$ ($R \times R$), $\mathbf{G}$ ($M_1 \times R$) and $\mathbf{H}$ ($R \times M_2$) contain time invariant system parameters. The system noise $\mathbf{w}_t$ ($R \times 1$) and the measurement noise $\mathbf{v}_t$ ($M_2 \times 1$) both have zero mean and fixed variance-covariance matrices and they are not serially or cross-dependent. The state vector $\mathbf{z}_t$ serves as the memory of the system. If $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$ are known, and if in addition a priori values of $\mathbf{z}_0$ and its covariance are given, then we may apply the recursive *Kalman filter* (Kalman, 1960; Kalman & Bucy, 1961) to estimate the latent state vectors $\mathbf{z}_t$ from the observations $\mathbf{x}_{1,t}$ and $\mathbf{x}_{2,t}$.

A practical advantage of the state space model is its Markov property, i.e. the current state vector contains all relevant information on the past of the system, and this makes it possible to develop efficient finite memory recursive estimation algorithms. Moreover, the model can be easily adapted to nonstationary series by introducing time varying parameters in the system matrices $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$.

On the other hand, the Kalman filter requires very much a priori information; not only $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$ need to be known, but initial estimates of states and error covariances must also be available. Typically, in the social sciences this information will be unavailable. A suitable alternative is to generate the information from the available data. This is the complicated field of system identification (cf. Eykhoff, 1974; Goodwin & Payne, 1977; Ljung & Söderström, 1983; Ljung, 1985). A particular problem with parameter estimation in dynamical systems is the multitude of different models that can be found that have an equal fit to the same observations.

Figure 2.6 illustrates the state space model. Note that all time dependence are channeled through the state vector $\mathbf{z}_t$. Since the noises $\mathbf{w}_t$ and $\mathbf{v}_t$ have zero expectation we may write the expected value structure of the state space model as

$$\mathbf{Z} = \mathbf{BZF} + \mathbf{X}_1\mathbf{G} \tag{2.39}$$

$$\mathbf{X}_2 = \mathbf{ZH}. \tag{2.40}$$

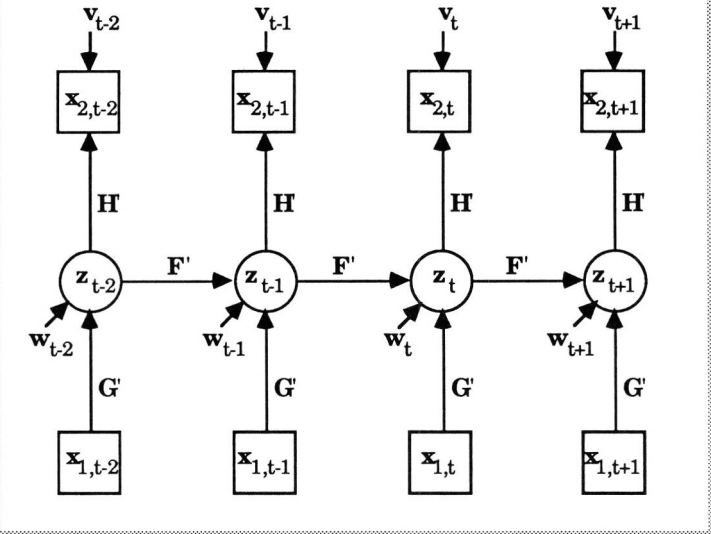


Figure 2.6 Scheme of the state space model

Fitting data to this structure is the DYNAMALS problem studied by De Leeuw and Bijleveld (1988) and Bijleveld (1989).

The state space model has many special cases. Akaike (1974) shows how to formulate the general ARMA(P,Q) model in state space form. Imposing the restriction $\mathbf{F} = \mathbf{0}$ we obtain cross-sectional regression and redundancy models. Otter (1986) discusses a state space representation of the LISREL model. MacCallum and Ashby (1986) and Oud, Van den Bercken and Essers (1986) illustrate the reverse.

2.8 Dynamic factor models

The use of factor models for analyzing multivariate time series has been repeatedly advocated by Cattell (1952, 1957, 1963). Cattell's P-technique bottles down to the application of the conventional cross-sectional factor

model on the data matrix **X**. The method has been criticized in a famous article by Anderson (1963) for a number of reasons. The main criticism is that P-technique does not take into account any time relationships of the variables, so there is a "danger of missing important and interesting characteristics which are significant because of their development in time though not because of their variability or high relationship to other variables" (Anderson, 1963, 9). Therefore, in dynamic generalizations of factor analysis several variables are tied together and the entire complex is explained in terms of its earlier history.

In general, the objective of dynamic factor analysis is to reduce the observed M-variate series to a latent R-variate series that is considered to be more fundamental. Dynamic factor models have been proposed by Priestley, Subba Rao and Tong (1973, 1974), Brillinger (1975, Ch. 9), Geweke (1977), Sargent and Sims (1977), Geweke and Singleton (1981), Engle and Watson (1981), Molenaar (1981, 1985, 1987), Immink (1986), Picci and Pinzoni (1986) and De Leeuw and Bijleveld (1988). Dynamic factor models have been applied with some success in psychometrics and econometrics.

We discriminate between two common definitions of the dynamic factor model: the "lagged factors" formulation (Brillinger, 1975; Molenaar, 1985) and the "state space" formulation (Engle & Watson, 1981; Immink, 1986).

Let $\mathbf{x}_t$ denote the $M \times 1$ vector of observed values at time t and let $\mathbf{z}_t$ be an $R \times 1$ vector of factor scores with $R \leq M$. The lagged factors dynamic factor model can be formulated as

$$\mathbf{x}_t = \sum_{q=0}^{Q} \mathbf{F}_q' \mathbf{z}_{t-q} + \mathbf{u}_t, \tag{2.41}$$

where $\mathbf{z}_t$ contains R independent common factors and where $\mathbf{u}_t$ corresponds to the unique factors, uncorrelated with the common factors. The $Q + 1$ matrices $\mathbf{F}_q$ are all of order $R \times M$ and they contain the lagged

factor loadings that connect the observed values to the latent factors over a time window of length Q + 1. In matrix notation the lagged factors model becomes

$$\mathbf{X} = \sum_{q=0}^{Q} \mathbf{B}_q \mathbf{Z} \mathbf{F}_q + \mathbf{U} \tag{2.42}$$

where $\mathbf{X}$ (N × M) represent M observed series, where $\mathbf{Z}$ (N × R) are R common latent factors and where $\mathbf{U}$ (N × M) corresponds with the unique factors. In order to identify the model it is assumed that $\mathbf{Z}$ is orthogonal and that $\mathbf{U}$ is a multivariate white noise series.

The state space dynamic factor model is equal to the state space model defined by (2.37) and (2.38) with $\mathbf{G}$ set to zero. A dynamic factor model can be seen as a linear system without inputs. In matrix notation, the state space dynamic factor model is defined by

$$\mathbf{Z} = \mathbf{B}\mathbf{Z}\mathbf{F}_1 + \mathbf{W} \tag{2.43}$$
$$\mathbf{X} = \mathbf{Z}\mathbf{F}_0 + \mathbf{V}, \tag{2.44}$$

where $\mathbf{Z}$ is an N × R matrix in which each row is an R-dimensional state vector, where $\mathbf{F}_0$ is the R × M output coefficients matrix, where $\mathbf{F}_1$ is the R × R system matrix and where $\mathbf{W}$ and $\mathbf{V}$ are system and measurement noises respectively. Immink (1986, 124) shows how the lagged factor model can be translated into a state space factor model.

Figure 2.7 is a schematic representation of the lagged factors dynamic model. Note that the structure of the scheme in 2.7 is like that of the MA(Q) part of the model in Figure 2.5. The important difference is that in the MA-model the latent variables are serially uncorrelated, whereas this is not the case for the dynamic factor model. This means that in dynamic factor analysis we are looking for latent variables that optimally summarize the time dependent structure of the observed data, whereas in the MA(Q) model we are interested in explaining the data in terms of random shocks.

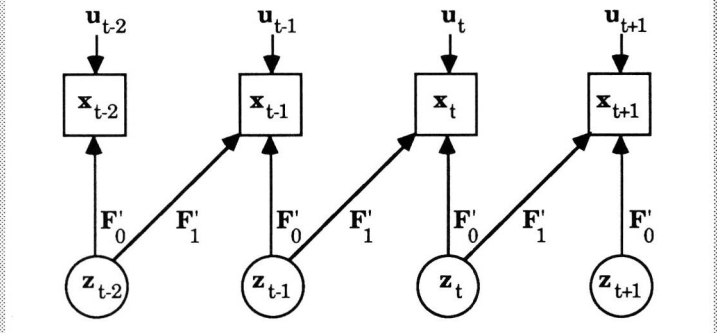


Figure 2.7 Scheme of a dynamic factor model

There are two types of unknowns in the dynamic factor model: the unobserved vector series $\mathbf{z}_t$, and the unknown factor loadings in $\mathbf{F}_0 \ldots \mathbf{F}_Q$. In order to estimate the factor loadings, it is possible to translate the dynamic factor model into a structural equations model and to estimate the loadings with LISREL (see Molenaar, 1985). Another possibility is to employ a scoring method that has the advantage that it requires only first derivatives (Engle & Watson, 1981; Immink, 1986). As a second step, we write the model in state space form and we use the Kalman filter to estimate the unobserved series $\mathbf{z}_t$ conditionally upon the obtained factor loadings.

2.9 Canonical correlation analysis and related techniques

Canonical correlation analysis (CCA) was invented by Hotelling (1936) to investigate linear relationships between two sets of variables. Statistical properties have been given by Bartlett (1947), Anderson (1958) and Kendall and Stuart (1968). Applications and relations to other multivariate techniques are discussed by McKeon (1966), Gittins (1985) and Van der Burg (1988).

CCA enjoys a modest popularity in time series analysis. Rijken van Olst (1981) contains many references to early and sometimes unintentional applications of CCA in the econometric literature. The usefulness of canonical variables in the analysis of multiple time series was already recognized by Quenouille (1957). Subsequently, Hannan (1967, 1970), Brillinger (1969, 1975), Robinson (1973), Akaike (1976), Box and Tiao (1977), Parzen and Newton (1980) and Velu et al. (1986) have used CCA for analyzing multiple series. Recently, interest has focussed on the canonical analysis of shifted univariate series as a diagnostics aid in the model specification phase (Jewell & Bloomfield, 1983; Jewell, Bloomfield & Bartmann, 1983; Tsay & Tiao, 1985; Hannan & Poskitt, 1988).

A somewhat confusing aspect of the time series literature is that canonical analysis not only refers to Hotteling's symmetric technique, but also encompasses the asymmetric reduced rank regression analysis (Izenman, 1975), also known as redundancy analysis (Van den Wollenberg, 1977) or simultaneous linear predictions (Fortier, 1966). Both the symmetric and asymmetric techniques are special cases of the general canonical correlation model discussed by Izenman (1975) and Brillinger (1975).

An interesting application of canonical analysis is the transformation proposed by Box and Tiao (1977). In the sequel we refer to their technique as the *Box-Tiao transformation* or as *predictable components*. Below we outline the method. More details can be found in Chapter 5.

Suppose we have a data matrix $\mathbf{X}$ of order $N \times M$ and that $\mathbf{X}$ can be described by a stationary multivariate AR(1) model

$$\mathbf{X} = \mathbf{BXC} + \mathbf{V} \tag{2.45}$$

with $\mathbf{C}$ of order $M \times M$. It is possible to transform $\mathbf{X}$ into M contemporaneous and temporally independent components $\underline{\mathbf{Z}}$ that follow the multivariate AR(1) process

$$\underline{\mathbf{Z}} = \mathbf{B}\underline{\mathbf{Z}}\mathbf{F} + \mathbf{E} \tag{2.46}$$

with **F** diagonal, by a linear transformation $\underline{\mathbf{Z}} = \mathbf{X}\mathbf{A}_0$. The M components $\underline{z}_1 \ldots \underline{z}_M$ can be ordered from the most to the least predictable. Let $\underline{\mathbf{z}}$ be a column from $\underline{\mathbf{Z}}$ and let $\hat{\mathbf{z}}$ denote the corresponding column in $\mathbf{B}\underline{\mathbf{Z}}\mathbf{F}$, then the predictability measure for that linear combination is

$$\gamma = \frac{\hat{\underline{\mathbf{z}}}'\hat{\underline{\mathbf{z}}}}{\underline{\mathbf{z}}'\underline{\mathbf{z}}} = 1 - \frac{\mathbf{e}'\mathbf{e}}{\underline{\mathbf{z}}'\underline{\mathbf{z}}} \tag{2.47}$$

i.e. the ratio of the predicted and the observed variance. The components that are most predictable can serve as smoothed composite indicators for overall growth, while the least predictable contain mainly noise. It is also possible to perform the transformation for higher order autoregressive processes. The estimation problem can be solved by computing canonical correlations between **X** and $[\mathbf{B}_1\mathbf{X},\ldots,\mathbf{B}_P\mathbf{X}]$.

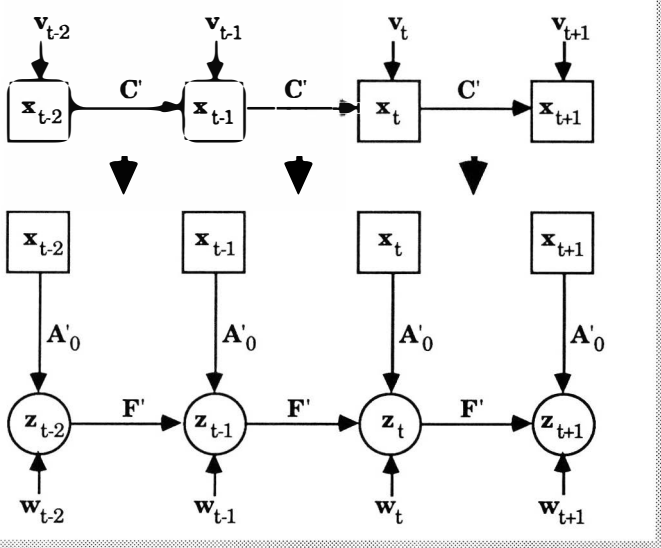


Figure 2.8 Box-Tiao canonical transformation

Figure 2.8 is a schematic representation of the predictable components technique. The Box-Tiao transform has obvious dimension reduction interpretations. Probably its major feature is the ability to separate cross-variable time relationships over independent components, resulting in a simpler time series model. Velu et al. (1986) view the transformation as a descriptive, correlational tool and they contrast it to predictor variance oriented techniques like reduced rank regression.

2.10 Graphic techniques

Graphic representations of multivariate time series are popular in the social sciences. A typical example is to plot the rows of the data as points in a low-dimensional principal subspace. Time can be included by connecting successive time points by a line so this line represents the trajectory of the series in the "state space". The resulting trajectory plot may then be inspected for regularities.

There are various ways to arrive at the low-dimensional subspace. When used in conjunction with correspondence analysis, the approach is especially popular in the French school of data analysis. Examples are given in Cabannes (1978, 1981), Saporta (1981), Deville and Saporta (1980, 1983), Greenacre (1984, cover, 267-270, 312-317), Hathout (1987), El Mousaoui (1987), Tenenhaus (1988) and Carlier et al. (1988). An alternative method for generating the subspace is to apply multidimensional scaling on a dissimilarity matrix of time points. This is illustrated in Guttman (1966) and Visser (1982, 164-170).

Inspection of the trajectory plot has some limitations. It is essentially a static method of the type criticized by Anderson (1963): the ordered structure of time is only taken into account afterwards as supplementary information. When applied to simple autoregressive series the displays can become quite unreadable. As an illustration consider the two artificial series plotted against each other in Figure 2.9. These series are mutually uncorrelated and they both follow a first order autoregressive

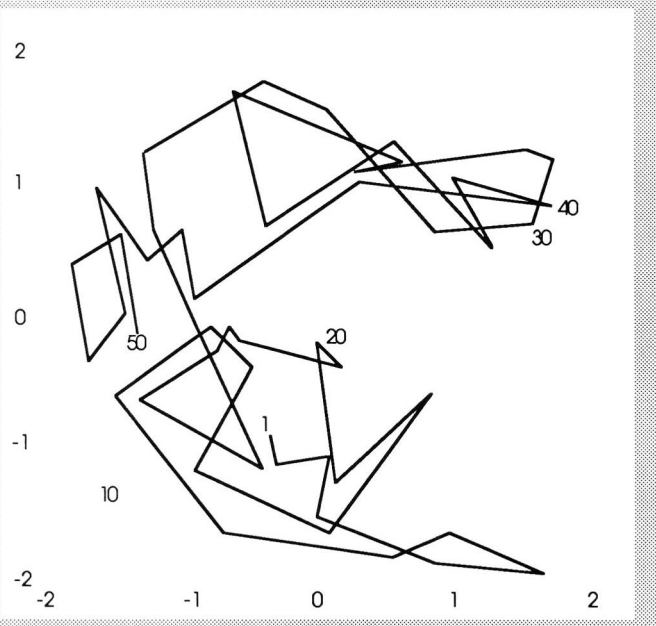


Figure 2.9 Scatterplot of two uncorrelated series with a = +0.80

process according to $x_t = 0.8\ x_{t-1} + e_t$, so they are relatively smooth. The plot shows a trajectory consisting of small steps in an apparently random direction. The small steps indicate high positive autocorrelations, but it is difficult to guess their magnitudes and to come up with a more thorough interpretation of the display.

The situation gets worse if we reverse the sign of the autoregressive parameter. In Figure 2.10 two uncorrelated series with $x_t = -0.8\ x_{t-1} + e_t$ are plotted against each other. These series behave in a very erratic way. Plotting half the number of points was enough to get the idea: the plot is a nice example of toddler drawing. No data analyst would present such a graph or would be willing to base any predictions on it. Yet, the underlying series can be succinctly described by a first order autoregressive

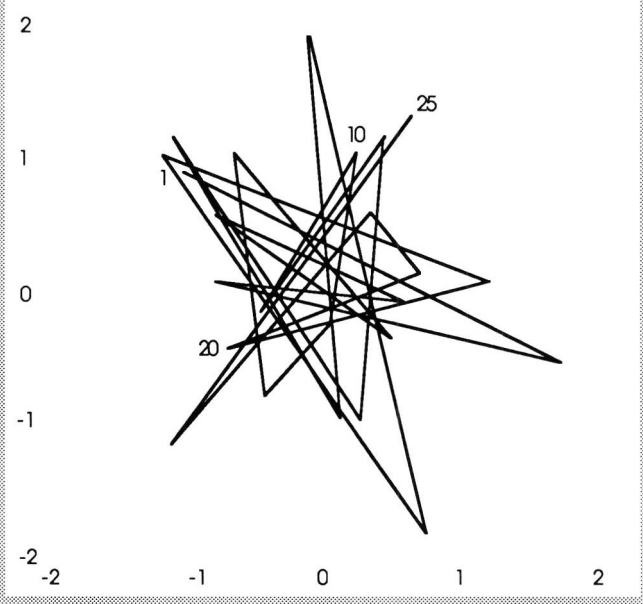


Figure 2.10 Scatterplot of two uncorrelated series with a = -0.80

model. It is also possible to accurately predict several time points ahead.

The trajectory plot has been successfully applied to short and smooth series. Many time series in the social sciences meet these requirements but as demonstrated above it is easy to think of situations in which the plot will reveal nothing but chaos.

CHAPTER 3

Optimal scaling

Data transformations have been used for a long time in time series analysis. For example, it is common practice to obtain logarithmic transforms of the variables. Smoothing and taking differences are also forms of data transformations. This chapter is concerned with yet another variety of transforms: optimal scaling. The central idea of optimal scaling is that categorical data are transformed into numerical data, such that the transform is optimal in some sense. Optimal scaling enables us to generalize many classic multivariate analysis techniques to their nonlinear, i.e. categorical, equivalents.

The main purpose of this chapter is to introduce a number of optimal scaling concepts and techniques in a systematic way. It heavily draws upon the Gifi system of nonlinear multivariate analysis (Gifi, 1981). First, we briefly discuss alternative approaches to optimal scaling. After this, homogeneity analysis is introduced as a means to find optimally scaled variables. The remainder of the chapter is concerned with restrictions. By systematically placing restrictions on the homogeneity solution it is possible to arrive at nonlinear versions of many multivariate techniques. The time series method proposed in Chapter 6 can also be considered as a restricted form of homogeneity analysis.

3.1 Optimal scaling techniques

The idea to assign numerical values to categories is quite old. Fisher (1940) and Guttman (1941) are generally recognized as predecessors of optimal scaling. Fisher was interested in obtaining an optimal scoring of the categories of a bivariate contingency table that maximizes the correlation between the quantified variables. Guttmans paper deals with the problem of reducing the information of a higher way contingency table to a single, most consistent latent variable. A well readable historical account of how their techniques have been rediscovered and reinvented over a period of three decennia is given in Van Rijckevorsel (1987). Other reviews include De Leeuw (1973, 1983a), Benzécri (1977), Nishisato (1980), Gifi (1980, 1981) and Tenenhaus and Young (1985).

The fact that the technique has been reinvented so many times not only demonstrates the lack of communication between researchers, but also the vitality and simplicity of the basic rationale. Until now we have seen American ALSOS (Young, 1981), Canadian dual scaling (Nishisato, 1980), Dutch homogeneity analysis (Gifi, 1981), French correspondence analysis (Greenacre, 1984), British optimal scoring (Fisher, 1940) and scoring systems (Healy & Goldstein, 1976), Israeli scalogram analysis (Guttman, 1941) and Japanese quantification methods (Hayashi, 1952). Although different authors stress other properties of the solution, these approaches can all be seen as attempts to find optimal transformations of the data, given some structural model.

In this chapter we will deal with only one particular variety of optimal scaling known as the Gifi system. Apart from geographical considerations, the choice for this system is based on the fact that the approach has generated a considerable amount of research.

Cornerstone of the Gifi system is homogeneity analysis. By imposing restrictions on the homogeneity solution, it is possible to generalize many linear multivariate techniques to the case of mixed measurement levels.

In this way, optimal scaling versions of principal components analysis (De Leeuw & Van Rijckevorsel, 1980; Bekker & De Leeuw, 1988), regression analysis and canonical analysis (Van der Burg & De Leeuw, 1983; Van der Burg, 1988), cluster analysis (Van Buuren & Heiser, 1989), Procrustes analysis (Van Buuren & Dijksterhuis, 1988) and path analysis (Coolen & De Leeuw, 1987) have been derived. The approach can be related to multidimensional scaling and unfolding techniques (Heiser, 1981; De Leeuw & Meulman, 1986) and many other distance based models (Meulman, 1986, Heiser & Meulman, 1987), loglinear analysis (Van der Heijden & De Leeuw, 1985; Van der Heijden, 1987), covariance structure models (De Leeuw & Mooijaart, 1987) and linear systems theory (Bijleveld, 1989).

The system relies on Alternating Least Squares (ALS) algorithms, i.e. the different sets of unknowns are approximated in an alternate fashion by least squares methods. See Young (1981) for more details on the ALS methodology, or in a more general context, Stoer and Bulirsch (1980, Ch. 8) on Gauss-Seidel and relaxation methods. Israëls (1987) formulates a considerable number of ALS techniques as eigenvalue problems which have closed form solutions. Sampling properties and stability results have been discussed in Gifi (1981), De Leeuw (1973, 1983b, 1984b) and Van der Burg (1988). Most applications have been in the social sciences, but there is a growing interest in econometrics (e.g. Keller & Wansbeek, 1983; Nijkamp et al., 1985).

3.2 Variable quantification with homogeneity analysis

This section outlines how variables are quantified, or scaled optimally, in the Gifi system. We will use the HOMALS loss function. Homogeneity analysis and its relationship to other techniques have been treated extensively elsewhere (e.g. Gifi, 1981), so we only summarize the main results here.

Suppose we have a $N \times M$ data matrix $\mathbf{H}$ of N objects measured on M

categorical variables. Each variable can be represented by an $N \times k_j$ indicator matrix $\mathbf{G}_j$. The scalar k_j denotes the number of (mutually exclusive) categories of the j^{th} variable and let S be the total number of categories summed over all variables. Each row of $\mathbf{G}_j$ contains exactly one "1" with the remaining entries being equal to zero. The inner product $\mathbf{D}_j = \mathbf{G}_j'\mathbf{G}_j$ is a $k_j \times k_j$ diagonal matrix, whose diagonal elements are equal to the number of observations per category. We define an $N \times R$ matrix $\mathbf{Z}$ of *object scores* and M matrices $\mathbf{Y}_j$, each of order $k_j \times R$, of *category quantifications*. The dimensionality R is assumed to be known and bounded by $1 \leq R \leq S - M$.

Homogeneity analysis can be formalized as the problem of minimizing the HOMALS loss function

$$\sigma(\mathbf{Z}; \mathbf{Y}_1 \ldots \mathbf{Y}_M) = M^{-1} \sum_{j=1}^{m} \text{ssq}\,(\mathbf{Z}, \mathbf{G}_j\mathbf{Y}_j), \tag{3.1}$$

over $\mathbf{Z}$ and $\mathbf{Y}_1 \ldots \mathbf{Y}_M$. To avoid the trivial solution in which $\mathbf{Z}$ and $\mathbf{Y}_j$ contain zeroes we require that $\mathbf{1}'\mathbf{Z} = \mathbf{0}$ and $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$. The solution is identified up to a rotation. We may remove this indeterminacy by rotating the solution such that the loss contribution for successive dimensions increases.

Minimization of (3.1) can be carried out by reciprocal averaging. If we start from random $\mathbf{Z}^0$ satisfying $\mathbf{1}'\mathbf{Z}^0 = \mathbf{0}$ and $\mathbf{Z}^{0\prime}\mathbf{Z}^0 = \mathbf{I}$ the following steps are computed until the solution becomes stable:

$$\mathbf{Y}_j^t \leftarrow \mathbf{D}_j^{-1}\mathbf{G}_j'\mathbf{Z}^{t-1} \qquad j = 1 \ldots M \tag{3.2a}$$

$$\mathbf{U}^t \leftarrow M^{-1} \sum_{j=1}^{m} \mathbf{G}_j\mathbf{Y}_j^t \tag{3.2b}$$

$$\mathbf{Z}^t \leftarrow \text{GRAM}(\mathbf{U}^t). \tag{3.2c}$$

Step (a) minimizes (3.1) over $\mathbf{Y}_1 \ldots \mathbf{Y}_M$ and steps (b) and (c) minimize (3.1) over $\mathbf{Z}$ under the normalization constraints. The function GRAM($\mathbf{U}$) stands for the Gram-Schmidt orthogonalization of $\mathbf{U}$ (cf. Stoer & Bulirsch, 1980). After convergence, (joint) plots of $\mathbf{Z}$ and $\mathbf{Y}_1 \ldots \mathbf{Y}_M$ may be

inspected for any regularities and patterns.

In homogeneity analysis the columns of the N × R matrices

$$\mathbf{K}_j = \mathbf{G}_j\mathbf{Y}_j \qquad j = 1...M \tag{3.3}$$

correspond to R distinct quantifications of the categorical variable $\mathbf{h}_j$. We call this *multiple* quantifications. The first column of $\mathbf{K}_j$, denoted by $\mathbf{k}_j$, contains the first optimal scaling of $\mathbf{h}_j$. Let us collect these vectors into a quantified data matrix

$$\mathbf{K} = [\mathbf{k}_1,...,\mathbf{k}_j,...,\mathbf{k}_M]. \tag{3.4}$$

It can be shown that homogeneity analysis maximizes the largest singular value of $\mathbf{K}$ over all possible quantifications (Bekker & De Leeuw, 1988). The technique can thus be viewed as a generalization of principal components analysis.

For reasons given below it is often useful to require that all $\mathbf{Y}_j$ have rank one. So if we define $\mathbf{Y}_j = \mathbf{y}_j\mathbf{a}_j'$ as a rank one matrix, where $\mathbf{a}_j$ is a R × 1 vector of loadings, then

$$\mathbf{Q}_j = \mathbf{G}_j\mathbf{y}_j\mathbf{a}_j' \tag{3.5}$$

is also of rank one and the columns of $\mathbf{Q}_j$ are proportional to each other. This is called *single* variable quantification since each variable is scaled only once. Except for R = 1, the optimal multiple and single quantifications are in general not identical. Without loss of generality we may choose the j^{th} quantified variable as $\mathbf{x}_j = \mathbf{G}_j\mathbf{y}_j$ and the quantified data matrix as

$$\mathbf{X} = [\mathbf{x}_1,...,\mathbf{x}_j,...,\mathbf{x}_M]. \tag{3.6}$$

Note that we prefer to use $\mathbf{X}$ instead of Gifi's $\mathbf{Q}$ since $\mathbf{X}$ is usually associated with the data matrix.

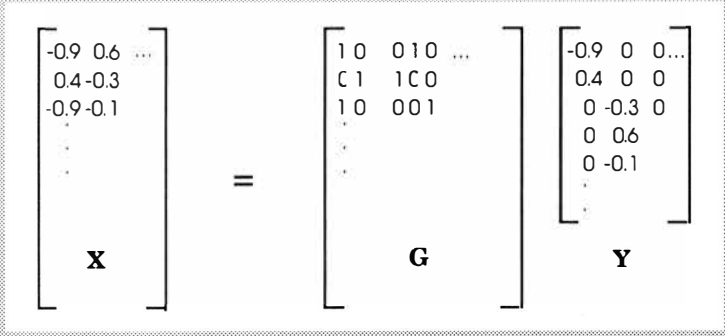


Figure 3.1 Construction of the quantified data matrix X = GY

Let $\mathbf{G} = [\mathbf{G}_1,\ldots,\mathbf{G}_j,\ldots,\mathbf{G}_M]$ of order N × S and let **Y** be the S × M block diagonal matrix that contains $\mathbf{y}_1 \ldots \mathbf{y}_M$ in its columns. We define

$$\mathbf{X} \equiv \mathbf{GY} \tag{3.7}$$

as the N × M quantified data matrix **X**. Figure 3.1 illustrates the way the matrices are organized.

It should be clear that (3.7) provides a convenient means to generalize a multivariate analysis technique to its optimal scaling equivalent. The definition expresses the quantified data matrix **X** as a linear combination of the observed indicator **G**. In our time series generalizations we will replace the data **X** in the model by the linear combination **GY**. The category quantifications in **Y** can then be estimated as additional model parameters.

It is often easier to generalize multivariate techniques to their nonlinear counterparts for single than for multiple quantifications. Other reasons why we prefer to quantify only once are the lower numbers of parameters that need to be estimated and the fact that constraints on the scale values can be more easily applied. Moreover, if desired, we can still compute multiple quantifications by copying the data matrix R times and analyze

this bigger matrix under the restriction that each copy contributes to only one dimension.

3.3 General use of restrictions

The free parameters $\mathbf{Z}$ and $\mathbf{Y}_j$ in the HOMALS loss function (3.1) may be restricted in several ways. The major uses of restricting the parameters are to

- identify the solution
- incorporate prior knowledge
- generate different models.

The identification of the solution is often carried out by placing normalization restrictions on either $\mathbf{Z}$ or $\mathbf{Y}_j$. For example, in the loss function (3.1) we chose to normalize $\mathbf{Z}$ in an orthonormal coordinate system. It is also possible to normalize $\mathbf{Y}_j$, but the choice of which normalization to apply is not always as trivial as it may seem. Although in many cases it does not matter whether one normalizes either $\mathbf{Z}$ or $\mathbf{Y}_j$, there are also problems in which the solution will depend on the specific normalization being used (cf. Goldstein, 1987).

Another use of restrictions is to incorporate prior knowledge in the analysis. For example, we could require that the analysis maintains the order of the categories if the categories are known to be ordered. Another example is in working with lagged variables. Since different lags of a given variable essentially measure the same it makes sense to constrain the category quantifications to be identical.

A third motive to restrict the solution is to arrive at interesting special cases, or different models. For example, if we restrict $\mathbf{Z}$ to be a linear combination of some external data set, the HOMALS problem becomes equivalent to the Canonical Correspondence Analysis technique proposed by Ter Braak (1986).

In general, applying restrictions will decrease the number of free parameters and so the results may become more stable, at least theoretically. In practice, computation may be difficult at times and solutions can be only locally optimal and therefore less stable. It is important to note that different types of constraints usually can not be freely intermixed without complicating the minimization problem. A lot of research has been done in minimizing the HOMALS loss function under specific combinations of constraints, leading to a range of variations on the HOMALS theme.

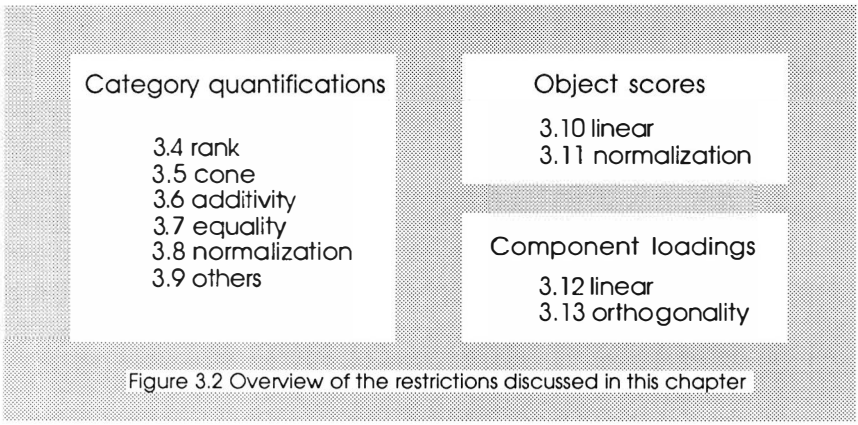


Figure 3.2 Overview of the restrictions discussed in this chapter

The remainder of this chapter classifies a number of currently employed and new restrictions into three groups according to the type of parameter the constraint applies to. The three parameter types are: the category quantifications $\mathbf{Y}_j$, the object scores $\mathbf{Z}$ and the component loadings $\mathbf{a}_j$. Figure 3.2 is a display of the restrictions we discuss.

We distinguish among six types of constraints on the quantifications. Rank restrictions determine the number of independent quantifications of a variable and cone restrictions specify relationships among categories. Additivity constraints can be used to specify the pattern of dependency among variables by assigning each variable to a specific set. Equality constraints restrict the category quantifications of different

variables to be identical. Finally, normalization constraints are needed to identify the parameters in the minimization problem.

Another way to constrain the HOMALS loss is to restrict the object scores $\mathbf{Z}$. We distinguish between linear and normalization constraints. Linear restrictions constrain the object scores to be a linear function of one or more observed or unobserved variables. Normalization constraints can be used to identify the parameters in the loss function.

If some or all $\mathbf{Y}_j$'s are subject to rank constraints it can be useful to restrict the component loadings in $\mathbf{a}_j$. We discuss linear and orthogonality restrictions. Linear restrictions either fix or constrain the range of the values of the component loadings vector of a given variable. Orthogonality constraints can be defined in conjunction with additivity restrictions in order to generalize the HOMALS loss function to Procrustes analysis.

Each restriction is discussed in a separate section. The sections on the equality constraint on the quantifications, the linear constraint on the object scores and both sections on the component loadings contain several new results and they supplement the Gifi system at some points. Each section contains hints in what circumstances to apply the restriction. The text partly parallels that of De Leeuw (1984a) and that of De Leeuw and Van Rijckevorsel (1988) who use other classification criteria.

3.4 Rank restrictions on the quantifications

Probably the most important type of constraint used in the Gifi system is to restrict the rank of $\mathbf{Y}_j$. In many cases, we want the rank to be equal to one. This rank-one restriction serves to single out that variable quantification that is optimal given a criterium model. We have seen the restriction before in section 3.2. It is very helpful in relating the HOMALS problem to a large number of linear multivariate techniques.

The rank-one restriction is represented as $\mathbf{Y}_j = \mathbf{y}_j\mathbf{a}_j'$, where $\mathbf{y}_j$ is a $k_j \times 1$ vector holding single category quantifications and where $\mathbf{a}_j$ is a $R \times 1$ vector of *component loadings*. Since $\mathbf{y}_j$ has rank one, $\mathbf{Y}_j$ also has rank one. The rank-one constraint may often be imposed on each variable separately. As said before, restricted variables are called single, unrestricted variables are known as multiple. A single variable always has only one quantification, a multiple variable has R independent quantifications, where R denotes the dimensionality of the problem.

As a general device, the singular value decomposition of $\mathbf{Y}_j$ can be used to determine $\mathbf{y}_j$ and $\mathbf{a}_j$ given $\mathbf{D}_j$ and $\mathbf{Y}_j$. If we take the singular value decomposition (SVD) of $\mathbf{Y}_j$ as

$$\mathbf{Y}_j = \mathbf{P}_j\Phi_j\mathbf{Q}_j' \tag{3.8}$$

then the $\mathbf{y}_j$ that minimizes (3.1) under rank-one constraints can be found as the first column of $\mathbf{P}_j$. It is usually convenient to scale $\mathbf{y}_j$ such that $\mathbf{1}'\mathbf{y}_j = 0$ and $\mathbf{y}_j'\mathbf{D}_j\mathbf{y}_j = 1$. The vector with component loadings $\mathbf{a}_j$ can be found by projecting $\mathbf{Y}_j$ on $\mathbf{y}_j$. Although actual estimation can be carried out more efficiently, the decomposition in (3.8) shows that it is straightforward to generalize the rank-one restriction to higher dimensional cases, a possibility discussed in De Leeuw and Van Rijckevorsel (1988), but it is unclear yet to what purpose one should want to do this.

3.5 Cone restrictions on the quantifications

Cone restrictions are used to restrict the induced category order to an a priori ordering (ordinal variables) or as a means to require that the induced scale is equal to the a priori scale (interval variables). They provide a way to incorporate prior knowledge regarding the measurement level of a variable.

Geometrically, cone restrictions constrain the feasible region for the $\mathbf{y}_j$-vector in the parameter space defined by $\mathbf{Y}_j$ of dimensionality k_j. The

quantification vector $\mathbf{y}_j$ may be located anywhere in the parameter space if the elements of $\mathbf{y}_j$ are not restricted. This is the case for (single) nominal variables. For an ordinal variable, for which the order of the categories is known, the feasible choices for $\mathbf{y}_j$ form a polyhedral convex cone (De Leeuw & Van Rijckevorsel, 1988). For interval variables the distances between categories are to be kept constant. The feasible $\mathbf{y}_j$ then makes up a line.

Finding the optimal $\mathbf{y}_j$ comes down to projecting the category points onto the cone (Gifi, 1981, 434-438). In the ordinal case the projection problem can be solved by monotone regression (cf. Kruskal, 1964, 1965). For interval variables, we use linear regression.

3.6 Additivity restrictions on the quantifications

It is sometimes meaningful to partition M variables into K sets. For the sake of illustration let M = 3 and K = 2 with the first two variables assigned to set 1 and with the third variable allocated to set 2. For the first set we may combine variable 1 and 2 into an interactive indicator matrix $\underline{\mathbf{G}_1}$ that consists of $k_1 \times k_2$ categories. Let $\underline{\mathbf{G}_2} = \mathbf{G}_3$. It is possible to perform homogeneity analysis on $\underline{\mathbf{G}_1}$ and $\underline{\mathbf{G}_2}$, resulting in category quantification matrices $\underline{\mathbf{Y}_1}$ and $\underline{\mathbf{Y}_2}$. It can be shown that if we restrict each element of $\underline{\mathbf{Y}_1}$ as a specific additive combination of some $\mathbf{Y}_1$ and $\mathbf{Y}_2$ then homogeneity analysis becomes equivalent to multiple regression analysis. More generally, we can say that K-set canonical correlation analysis can be interpreted as homogeneity analysis with linear restrictions on the quantifications. We refer to Gifi (1981, 207-208) for more detailed information on this relationship.

An equivalent though computational more efficient way to perform K-set canonical analysis is to collect all indicators and quantifications of the k^{th} set (k = 1...K) into super arrays $\mathbf{G}_k = \{\mathbf{G}_j \in C_k\}$ and $\mathbf{Y}_k = \{\mathbf{Y}_j \in C_k\}$ where C_k denotes the set of all variables belonging to set k. Then

$$\sigma(\mathbf{Z}, \mathbf{Y}_1 \ldots \mathbf{Y}_K) = \sum_{k=1}^{K} \text{ssq}\ (\mathbf{Z}, \mathbf{G}_k\mathbf{Y}_k). \tag{3.9}$$

is the OVERALS loss function, which can be minimized by an adapted version of the homogeneity analysis algorithm. Loss function (3.9) has been studied extensively by Gifi (1981, Ch. 6-8) and Van der Burg (1988). These authors show how the additivity constraint can be used to arrive at nonlinear versions of regression analysis, canonical correlation analysis, discriminant analysis, redundancy analysis and MANOVA.

3.7 Equality restrictions on the quantifications

A less common constraint is to require the quantifications of two or more variables to be equal. Gifi (1981, 264-273), De Leeuw et al. (1985) and Van der Lans and Heiser (1988) discuss some applications of the equality constraint. The main application so far has been in the field of classical scaling techniques.

For many techniques developed in next chapters, we will use the constraint to equate the quantifications of different lags of the same variable. The purpose for doing so is to make the scale values independent of time. There seems to be little value in obtaining different scalings for exactly the same score only because it has shifted in time. We remove this artifact by applying equality constraints. This has the additional advantage that less parameters need to be estimated.

Let us take a look at the restriction. Suppose that we want to restrict the multiple quantification matrices to be equal, i.e.

$$\mathbf{V} = \mathbf{Y}_1 = \ldots = \mathbf{Y}_j = \ldots = \mathbf{Y}_M. \tag{3.10}$$

This is only possible if all variables have an equal number of categories. The loss function of the restricted problem becomes

$$\sigma(\mathbf{Z}, \mathbf{V}) = \sum_{j=1}^{m} \text{ssq}\ (\mathbf{Z}, \mathbf{G}_j\mathbf{V}), \tag{3.11}$$

which must be minimized under $\mathbf{1}'\mathbf{Z} = \mathbf{0}$ and $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$. One way of solving this problem is to partition the loss as

$$\sigma\ (\mathbf{Z}, \mathbf{V}) = \sum_{j=1}^{m} \text{ssq}\ (\mathbf{Z}, \mathbf{G}_j\tilde{\mathbf{Y}}_j) + \sum_{j=1}^{m} \text{tr}\ (\tilde{\mathbf{Y}}_j - \mathbf{V})'\mathbf{D}_j(\tilde{\mathbf{Y}}_j - \mathbf{V}), \tag{3.12}$$

which is valid if we define $\tilde{\mathbf{Y}}_j = \mathbf{D}_j^{-1}\mathbf{G}_j'\mathbf{Z}$ for $j = 1...M$ so that all cross-product terms vanish. The second part of (3.12) is minimized over $\mathbf{V}$ given $\tilde{\mathbf{Y}}_j$ by

$$\mathbf{V} = (\Sigma_j\ \mathbf{D}_j)^{-1}(\Sigma_j\ \mathbf{D}_j\tilde{\mathbf{Y}}_j). \tag{3.13}$$

If we have single variables, we must deal with M subproblems of minimizing

$$\sum_{j=1}^{m} \text{tr}\ (\tilde{\mathbf{Y}}_j - \mathbf{y}\mathbf{a}_j')'\mathbf{D}_j(\tilde{\mathbf{Y}}_j - \mathbf{y}\mathbf{a}_j') \tag{3.14}$$

over $\mathbf{y}$ and $\mathbf{a}_j'$. We may solve this problem by computing the SVD of

$$[\tilde{\mathbf{Y}}_1,...,\tilde{\mathbf{Y}}_j,...,\tilde{\mathbf{Y}}_M] = \mathbf{P}\Phi\mathbf{Q}',$$

as in PRINCALS, and take the first column of $\mathbf{P}$ as an estimate of $\mathbf{y}$. This approach will only work if the marginal frequencies for all variables are proportional to each other, as is the case in for example ranking order data. In other cases we can iterate over $\mathbf{y}$ and $\mathbf{a}_j$ by

$$\mathbf{y} = (\Sigma_j\ \mathbf{a}_j'\mathbf{a}_j\mathbf{D}_j)^{-1}\ (\Sigma_j\ \mathbf{D}_j\tilde{\mathbf{Y}}_j\mathbf{a}_j) \tag{3.15}$$

and

$$\mathbf{a}_j' = (\mathbf{y}'\mathbf{D}_j\mathbf{y})^{-1}\ (\mathbf{y}'\mathbf{D}_j\tilde{\mathbf{Y}}_j). \tag{3.16}$$

If desired, additional cone restrictions may be imposed after the **y**-step. It is straightforward to generalize to situations where only some variables are required to obtain equal scaling.

Note that the procedure outlined in this section will produce scaled variables that are not in deviations from their means. It is in general not possible to require $\mathbf{1}'\mathbf{D}_j\mathbf{y} = 0$ for all $j = 1...M$ simultaneously because the marginal frequencies $\mathbf{D}_j$ may be unequal for different variables. If a variable does not have zero mean, this will introduce an "unnecessary" loss component. One way to evade this is to take each indicator matrix in deviations from its column mean so any linear combination of the columns will also have zero mean. As noted by Van de Geer (1986) all rows will add up to zero, which causes $\mathbf{G}_j'\mathbf{G}_j$ to be singular. A remedy is to delete an arbitrary column from the matrix. The deleted category will be treated as if it had obtained a quantification of zero, and the results can be recomputed for the original categories by simple arithmetic. Another way is to include a centering operator into the loss function. This is done by Van der Lans and Heiser (1988). A drawback of either method is that the diagonality property of $\mathbf{D}_j$ is destroyed so computation becomes more demanding. Since the differences in marginal frequencies for lagged variables are small, we prefer not to correct for the mean at all.

If the variables are partitioned into sets, application of the equality constraint is more difficult because the loss function cannot be split anymore as in (3.12). We not only have to restrict the quantifications between sets, as above, but within sets as well. A general method to do this is presented in Chapter 6.

3.8 Normalization restrictions on the quantifications

Normalization restrictions have little intrinsic interest as they only act to prevent degenerate solutions. If all variables are multiple, two options to standardize the solution are available. The first is

$$\mathbf{1}'\mathbf{D}_j\mathbf{Y}_j = \mathbf{0}$$
$$\mathbf{Y}_j'\mathbf{D}_j\mathbf{Y}_j = \mathbf{I} \tag{3.17}$$

for j = 1...M. This strong orthogonality restriction orthonormalizes all variables. A second option is

$$\mathbf{1}'\mathbf{D}_j\mathbf{Y}_j = \mathbf{0}$$
$$\sum_{j=1}^{m} \mathbf{Y}_j'\mathbf{D}_j\mathbf{Y}_j = \mathbf{I}. \tag{3.18}$$

This is weak orthogonality: only the sum of the quantified variables need to be orthonormal. As strong orthogonality is more restrictive than weak orthogonality the solutions under both constraints will be different.

Usually it is preferable to normalize the object scores instead, but in some cases, e.g. when the object scores are being subjected to additional restrictions, this can be quite inconvenient. The weak orthogonality constraint has the advantage that it will provide a solution that is proportional to the solution with the more common normalization $\mathbf{1}'\mathbf{Z} = \mathbf{0}$ with $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$.

3.9 Other restrictions on the quantifications

De Leeuw (1973, Ch. 4) and Gifi (1981, Ch. 10) discuss a variety of other linear and nonlinear constraints on the quantifications. These restrictions can be used to connect homogeneity analysis to paired comparison scaling, latent structure models, nonmetric factor analysis and three way models. There is not much experience with these options, but according to Gifi "their flexibility and generality guarantees the possibility of an unending stream of programs and publications in the appropriate journals" (Gifi, 1981, 278).

3.10 Linear restrictions on the object scores

It is sometimes interesting to restrict the object scores. We represent the object scores as N points in R-dimensional space. Some examples of restrictions are: the location of an object is fixed, the distance between two objects is known in advance, or the objects are ordered according to some scale or grouping criterion.

In this section we study the linear constraint $\mathbf{Z} = \mathbf{PU}$ for any nonsingular "design-matrix" $\mathbf{P}$. The HOMALS loss function (3.1) then becomes

$$\sigma(\mathbf{Z}; \mathbf{Y}_1 \ldots \mathbf{Y}_M) = M^{-1} \sum_{j=1}^{m} \text{ssq}\,(\mathbf{PU}, \mathbf{G}_j\mathbf{Y}_j). \tag{3.19}$$

We define

$$\tilde{\mathbf{Z}} = M^{-1} \sum_{j=1}^{m} \mathbf{G}_j\mathbf{Y}_j, \tag{3.20}$$

and insert $\mathbf{PU} = \tilde{\mathbf{Z}} - (\tilde{\mathbf{Z}} - \mathbf{PU})$ into (3.19). Then

$$\begin{aligned} \sigma(\mathbf{U}; \mathbf{Y}_1 \ldots \mathbf{Y}_M) &= M^{-1} \sum_{j=1}^{m} \text{ssq}\,((\tilde{\mathbf{Z}} - \mathbf{G}_j\mathbf{Y}_j) - (\tilde{\mathbf{Z}} - \mathbf{PU})) \\ &= M^{-1} \sum_{j=1}^{m} \text{ssq}\,(\tilde{\mathbf{Z}}, \mathbf{G}_j\mathbf{Y}_j) + \text{ssq}\,(\tilde{\mathbf{Z}}, \mathbf{PU}). \end{aligned} \tag{3.21}$$

The total loss can be split into two additive components. So, for given $\tilde{\mathbf{Z}}$ the total loss decreases by minimizing ssq $(\tilde{\mathbf{Z}}, \mathbf{PU})$ over $\mathbf{U}$ by projection. For given $\mathbf{Z} = \mathbf{PU}$ the loss may be lowered by finding $\mathbf{Y}_1 \ldots \mathbf{Y}_M$ with the usual procedures.

The nonmetric cluster analysis method GROUPALS (Van Buuren, 1986; Van Buuren & Heiser, 1989) contains an example of a linear restriction on the object scores. In GROUPALS, each object is restricted to be located at only one of K points by the restriction $\mathbf{Z} = \mathbf{G}_c\mathbf{Y}_c$, where $\mathbf{G}_c$ is an $N \times k_c$

indicator matrix of group memberships, and where the $k_c \times R$ matrix $\mathbf{Y}_c$ holds the cluster means. The clustering problem is slightly more complicated than the linear constraint problem because not only $\mathbf{Y}_c$, but also design matrix $\mathbf{G}_c$ is unknown, and so the second part of (3.21) must be minimized over both $\mathbf{G}_c$ and $\mathbf{Y}_c$. One way to do this is by using an iterative relocation algorithm such as K-means (Hartigan, 1975). After convergence, $\mathbf{G}_c$ is the indicator matrix of the most homogeneous with respect to the variables and the most discriminating with respect to the objects k_c-category variable. Another application of the linear restriction is Canonical Correspondence Analysis (Ter Braak, 1986), where $\mathbf{P}$ is a matrix of background variables.

Another type of linear restriction can be used to fix the locations of some of the object scores, or to create specific (e.g. order) patterns among the objects.

Let $\mathbf{z} = \text{vec } \mathbf{Z}$, i.e. $\mathbf{z}$ is an $NR \times 1$ column vector in which the columns of $\mathbf{Z}$ are stacked, let $\mathbf{P}$ be any $N \times NR$ consistent design matrix, and let $\mathbf{r}$ be a known $N \times 1$ vector. The linear constraint $\mathbf{Pz} = \mathbf{r}$ specifies N linear combinations of the object scores coordinates. For example, $\mathbf{P} = \mathbf{I}$ fixes the entire object space. Alternatively, if $\mathbf{P}$ is a transposed indicator matrix, groups of objects are forced to be located at specific points.

Again we must solve the problem of minimizing $\text{ssq}(\tilde{\mathbf{Z}}, \mathbf{Z}) = \text{ssq}(\tilde{\mathbf{z}}, \mathbf{z})$, but now satisfying $\mathbf{Pz} = \mathbf{r}$. We can approximate this $\mathbf{z}$ by restricted least squares (Magnus & Neudecker, 1988, 233) as

$$\mathbf{z} = \mathbf{N}^+\tilde{\mathbf{z}} + \mathbf{N}^+\mathbf{P}'(\mathbf{P}\mathbf{N}^+\mathbf{P})^+(\mathbf{r} - \mathbf{P}\mathbf{N}^+\tilde{\mathbf{z}}), \tag{3.22}$$

where $\mathbf{N} = \mathbf{I} + \mathbf{P}'\mathbf{P}$ and where $\mathbf{N}^+$ denotes the Moore-Penrose inverse of $\mathbf{N}$.

3.11 Normalization restrictions on the object scores

A common normalization used in the Gifi system is $\mathbf{1}'\mathbf{Z} = \mathbf{0}$ with $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$, i.e. each object scores variate has zero mean, unit sums of squares and the variates are mutually uncorrelated. In some cases both $\mathbf{Z}$ and $\mathbf{Y}_j$ are subjected to any other restrictions beyond normalization. In this case we must simultaneously deal with two or more types of restrictions, and this may lead to computational problems. A way out is to apply a transfer of normalization procedure. The idea is that a normalization on $\mathbf{Z}$ can be transferred to a normalization on $\mathbf{Y}_1 \ldots \mathbf{Y}_M$, and the other way around, while preserving the loss. See Van der Burg and De Leeuw (1983) and Van Buuren and Heiser (1989) for more details on this procedure.

Other normalizations can also be made. For example, Healy and Goldstein (1976) and Goldstein (1987) suggest to fix the location of two object points, which they call endpoints, at 0 and 1. Since their normalization constraint is linear, the solution will be different from the one corresponding to the quadratic constraint $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$.

3.12 Linear restrictions on the loadings

For single variables, the component loadings indicate in what way a variable contributes to the solution in a particular dimension. An obvious use of restricting the loadings is to set some elements of $\mathbf{a}_j$ to zero, so that the j^{th} variable does not load the corresponding dimensions. More generally, they allow to regulate the influence of the j^{th} variable on the analysis using a priori information. See for example Tiao and Box (1981) on how parameter constraints can be used in multiple time series analysis.

Suppose that we constrain the component loadings $\mathbf{a}_j$ according to the linear restriction $\mathbf{P}_j\mathbf{a}_j = \mathbf{r}_j$ for a consistent design matrix $\mathbf{P}_j$ and a known vector $\mathbf{r}_j$. Starting from the rank-one loss function, the loss for variable j may be partitioned as

$$\sigma(\mathbf{Z}; \mathbf{y}_j;\mathbf{a}_j) = \text{ssq}\,(\mathbf{Z}, \mathbf{G}_j\mathbf{y}_j\mathbf{a}_j') = \text{ssq}\,(\mathbf{Z}, \mathbf{G}_j\mathbf{y}_j\tilde{\mathbf{a}}_j') + \mathbf{y}_j'\mathbf{D}_j\mathbf{y}_j\,(\tilde{\mathbf{a}}_j - \mathbf{a}_j)'(\tilde{\mathbf{a}}_j - \mathbf{a}_j), \tag{3.23}$$

with $\tilde{\mathbf{a}}_j = \mathbf{Z}'\mathbf{G}_j\mathbf{y}_j / \mathbf{y}_j'\mathbf{D}_j\mathbf{y}_j$. The total loss decreases if we minimize the second loss component $(\tilde{\mathbf{a}}_j - \mathbf{a}_j)'(\tilde{\mathbf{a}}_j - \mathbf{a}_j)$ over $\mathbf{a}_j$. For a consistent design matrix $\mathbf{P}_j$ this can be done using the restricted least squares approximation (Magnus & Neudecker, 1988)

$$\mathbf{a}_j = \mathbf{a}_j^0 + \mathbf{N}^+\mathbf{P}_j'(\mathbf{P}_j\mathbf{N}^+\mathbf{P}_j')^+(\mathbf{r}_j - \mathbf{P}_j\mathbf{a}_j^0), \tag{3.24}$$

where $\mathbf{a}_j^0 = \mathbf{N}^+\tilde{\mathbf{a}}_j$, $\mathbf{N} = \mathbf{I} + \mathbf{P}_j'\mathbf{P}_j$ and $\mathbf{N}^+$ is the Moore-Penrose inverse of $\mathbf{N}$. In the special case that $\mathbf{r}_j = \tilde{\mathbf{a}}_j$ and $\mathbf{P}_j = \mathbf{I}$, i.e. $\mathbf{a}_j$ is unrestricted, it can be easily verified that $\mathbf{a}_j^0 = 0.5\ \tilde{\mathbf{a}}_j$ and $\mathbf{a}_j = 0.5\ \tilde{\mathbf{a}}_j + 0.5\ \tilde{\mathbf{a}}\tilde{}_j = \tilde{\mathbf{a}}_j$.

3.13 Orthogonality restrictions on the loadings

Suppose we partition KR variables into K sets, so that each set contains R variables. An example is that we have K judges that rate N stimuli on R attributes. Gower (1975) proposed to analyze such data by means of Procrustes rotation for K sets. Extensions and further results are given in Ten Berge (1977), Ten Berge and Knol (1984) and Peay (1988).

Procrustes analysis can be formulated in terms of homogeneity analysis by using additivity restrictions on the quantifications and orthogonality constraints on the loadings. Let $\mathbf{X}_k = [\mathbf{G}_{k1}\mathbf{y}_{k1},\ldots,\mathbf{G}_{kR}\mathbf{y}_{kR}]$ be the $N \times R$ matrix containing the quantified variables in the k^{th} set ($k = 1\ldots K$) and let $\mathbf{A}_k = [\mathbf{a}_{k1},\ldots,\mathbf{a}_{kR}]'$ denote the $R \times R$ matrix of corresponding loadings. The within-set orthogonality constraint is defined as $\mathbf{A}_k'\mathbf{A}_k = \mathbf{A}_k\mathbf{A}_k' = \mathbf{I}$ for all k. The discrete data Procrustes loss function is

$$\sigma(\mathbf{Z}; \mathbf{X}_1\ldots\mathbf{X}_K; \mathbf{A}_1\ldots\mathbf{A}_K) = K^{-1}\sum_{k=1}^{K}\text{ssq}\,(\mathbf{Z}, \mathbf{X}_k\mathbf{A}_k). \tag{3.25}$$

For fixed $\mathbf{X}_k$ (or equivalently $\mathbf{y}_j$) the loss can be minimized over $\mathbf{A}_1 \ldots \mathbf{A}_K$ by a standard K-sets Procrustes rotation algorithm (cf. Ten Berge, 1977).

The problem of minimizing the loss over $\mathbf{X}_k$ becomes easy if we rewrite the loss into a more convenient form. Since $\mathbf{A}_k\mathbf{A}_k' = \mathbf{I}$ we can reformulate (3.25) as

$$\sigma(\mathbf{Z}; \mathbf{X}_1 \ldots \mathbf{X}_K; \mathbf{A}_1 \ldots \mathbf{A}_K) = K^{-1} \sum_{k=1}^{K} \text{ssq}\ (\mathbf{Z}\mathbf{A}_k', \mathbf{X}_k). \tag{3.26}$$

Let $\tilde{\mathbf{z}}_r$ denote the r^{th} column of $\mathbf{Z}\mathbf{A}_k'$. We can now minimize the loss over each $\mathbf{y}_{kr}$ by solving the simple problem of minimizing ssq $(\tilde{\mathbf{z}}_r, \mathbf{G}_{kr}\mathbf{y}_{kr})$ for all $k = 1 \ldots K$ and $r = 1 \ldots R$. Additional cone restrictions can be applied without a problem. The discrete Procrustes rotation problem has been described in more detail by Van Buuren and Dijksterhuis (1988).

CHAPTER 4

Univariate time series analysis with optimal scaling

In this chapter we study in what way univariate time series techniques can be combined with optimal scaling. We describe three simple filters: a sum filter, a difference filter and an exponential smoothing filter, and we show how the autocorrelations of the series is affected by optimal scaling under these three filters. Next, the filters are related to regression analysis with one lagged variable. It is shown that both the filters and the regression methods either maximize or minimize the first-order autocorrelation of the series as an outcome of the optimal transformation. The chapter continues with two useful extensions of the simple regression model. The first one is seasonal autoregression. It can be used to describe periodic components in a series. The second extension, multiple autoregression, incorporates multiple lagged predictor series, and it is a generalization of the Box-Jenkins AR(P) model to the case of mixed measurement levels. All methods are illustrated with examples.

4.1 Univariate transformations

Sections 4.2 to 4.4 deal with the behavior of a number of elementary univariate time series transformations. These simple transforms have been widely used for many years and they provide a perfect means to investigate the effect of optimal scaling on the solution.

In general, time series transforms are undertaken to extract a certain kind of information from the observations. A transformation, also called a *filter*, usually has an input and an output. For example, a radio tuner filters out the signal of one specific station, while discarding those of thousands of others as being irrelevant noise. In statistics, a familiar type of transformation is to subtract the mean from a given series, so the new series will have zero mean. Here, the raw series is the input, and the zero mean series represents the output. In general, the function of a filter is to change the input into the output in a predictable way.

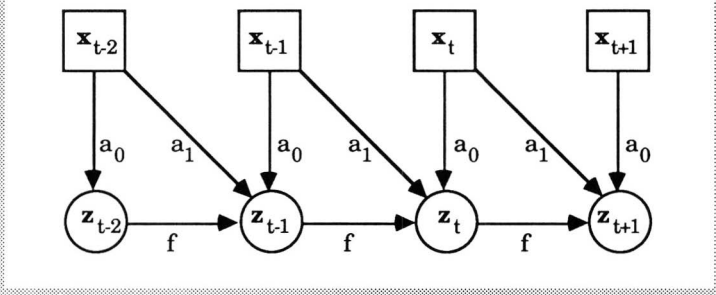


Figure 4.1 Scheme of the general transform (4.1)

We will be concerned with three filters that are all special cases of the general transform

$$\mathbf{z} = f\,\mathbf{B}\mathbf{z} + a_0\mathbf{x} + a_1\mathbf{B}\mathbf{x}, \tag{4.1}$$

where $\mathbf{x}$ is the input, $\mathbf{z}$ is the output and where f, a_0 and a_1 are fixed filter coefficients. The general transform (4.1) is represented schematically in Figure 4.1.

We distinguish among the following cases:

• sum filter	$\mathbf{z} = \mathbf{x} + \mathbf{Bx}$	$f = 0, a_0 = a_1 = 1$
• difference filter	$\mathbf{z} = \mathbf{x} - \mathbf{Bx}$	$f = 0, a_0 = 1, a_1 = -1$
• exponential smoothing filter	$\mathbf{z} = f\mathbf{Bz} + \mathbf{x}$	$a_0 = 1, a_1 = 0$

Studying these filters is useful to get an idea in what way the input differs from the output for some simple choices of the filter coefficients. This will facilitate the interpretation of results in which the filter coefficients are being determined by the data, as in regression analysis.

4.2 Sum filter

Suppose that we construct a new series by adding the previous observation to the current observation. This is the *sum filter* $\mathbf{z} = \mathbf{x} + \mathbf{Bx}$, where $\mathbf{x}$ is the input and $\mathbf{z}$ is the output. The structure of the filter is schematically represented in Figure 4.2.

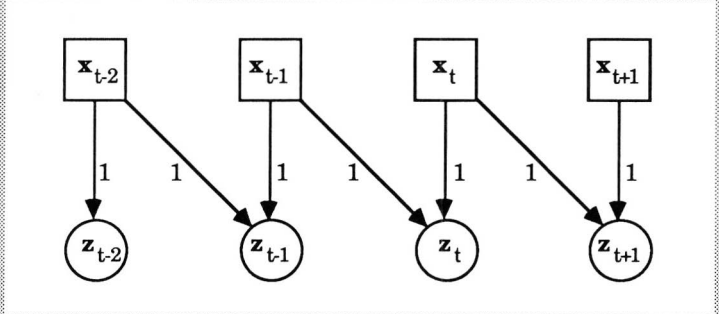


Figure 4.2 Scheme of the sum filter

The sum filter is an example of a linear smoothing filter. The aim of smoothing is to discard fast fluctuations in order to make an underlying trend more visible. The moving average filter $\mathbf{z} = 0.5\ (\mathbf{x} + \mathbf{Bx})$ is, except for scaling, equivalent to the sum filter. A moving average filter requires that all filter coefficients are positive and add up to unity.

If $\mathbf{x}$ is a numerical series the application of the sum filter to $\mathbf{x}$ is simple. However summing a categorical series is more problematical since addition of categories is not defined. Nevertheless, adopting the optimal scaling approach, it is possible to apply the sum filter not to the categorical series itself, but to an optimal transformation of it. This problem can be formalized in a least squares context as minimizing

$$\sigma_{sum}(\mathbf{z}; \mathbf{x}) = \text{ssq}\ (\mathbf{z}, \mathbf{x}) + \text{ssq}\ (\mathbf{z}, \mathbf{Bx}) \tag{4.2}$$

over $\mathbf{x} = \mathbf{Gy}$ and $\mathbf{z}$ under constraints $\mathbf{1}'\mathbf{z} = \mathbf{1}'\mathbf{x} = 0$ and $\mathbf{z}'\mathbf{z} = 1$. Matrix $\mathbf{G}$ is an indicator matrix and vector $\mathbf{y}$ contains the optimal category quantifications. The sum filter is an example of a low pass filter, i.e. high frequencies are being discarded as irrelevant noise, so the filtered series looks smoother than the original. Low pass filters are therefore of considerable interest for dimension reduction.

It is interesting to study which effect minimizing (4.2) has on the correlation between $\mathbf{x}$ and $\mathbf{Bx}$. If we define $\mathbf{z}^* = 0.5\ (\mathbf{x} + \mathbf{Bx})$ then (4.2) can be partitioned as follows

$$\begin{aligned}\sigma_{sum}(\mathbf{z}; \mathbf{x}) &= 2\ \text{ssq}\ (\mathbf{z}^*, \mathbf{z}) + \text{ssq}\ (\mathbf{z}^*, \mathbf{x}) + \text{ssq}\ (\mathbf{z}^*, \mathbf{Bx}) \\ &= 2\ \text{ssq}\ (\mathbf{z}^*, \mathbf{z}) + 0.5\ \text{ssq}\ (\mathbf{x}, \mathbf{Bx}).\end{aligned} \tag{4.3}$$

The first component ssq $(\mathbf{z}^*, \mathbf{z})$ measures the deviation between the best estimate $\mathbf{z}^* = 0.5\ (\mathbf{x} + \mathbf{Bx})$ and its normalized version (i.e. $\mathbf{1}'\mathbf{z} = 0$, $\mathbf{z}'\mathbf{z} = 1$). The second part of the function ssq $(\mathbf{x}, \mathbf{Bx})$ is more exciting since it expresses the loss value as a function of the covariance, and thus also of the correlation, between $\mathbf{x}$ and $\mathbf{Bx}$. If we write out the loss components by the correlation decomposition theorem and substitute for $\mathbf{z}^*$ then it is not

difficult to derive the approximate relationship

$$\sigma_{sum}(\mathbf{z}; \mathbf{x}) \approx 1 - r_{1x}, \tag{4.4}$$

for the first-order autocorrelation r_{1x} of $\mathbf{x}$.

The inexactness of (4.4) is due the fact that $\mathbf{Bx}$ is not precisely in deviations from its mean. In the remainder we sometimes use the approximate equality sign "≈" in this situation. It is also possible to derive exact relationships, but the formulas will then become embellished with rather complex correction terms. Since the contribution of the correction terms will be usually small, we ignore them here and prefer simplicity above exactness.

The result clearly shows that minimizing $\sigma_{sum}(\mathbf{z}; \mathbf{x})$ over nonlinear transforms of $\mathbf{x}$ maximizes the first-order autocorrelation r_{1x} of the input series. Stated conversely, highly autocorrelated input series will result in a better fit. Of course, when $\mathbf{x}$ is numerical its autocorrelation is constant, so there is not much to minimize then, except for setting $\mathbf{z} = \mathbf{x} + \mathbf{Bx}$ and normalizing the result.

It is also possible arrive at an approximate relationship between the loss $\sigma_{sum}(\mathbf{z}; \mathbf{x})$ and the squared canonical correlations r_{0zx}^2 and r_{1zx}^2. Under the normalizations we use, it can be established that after convergence $\mathbf{x}'\mathbf{x} = r_{0zx}^2$ and $\mathbf{x}'\mathbf{B}'\mathbf{Bx} = r_{1zx}^2$ if all variables have zero mean. Gifi (1981, 96) calls these quantities discrimination measures. After some algebraic manipulation and ignoring end effects we obtain

$$\sigma_{sum}(\mathbf{z}; \mathbf{x}) \approx 2 - r_{0zx}^2 - r_{1zx}^2. \tag{4.5}$$

This equation shows that $\mathbf{z}$ will be as close as possible to the input series $\mathbf{x}$ and the lagged series $\mathbf{Bx}$. Obviously, we achieve this by taking $\mathbf{z}$ as the mean of $\mathbf{x}$ and $\mathbf{Bx}$. Since we have two sets here, the loss will be equally distributed over both sets, so the discriminations measures will be identical, i.e. $r_{0zx}^2 = r_{1zx}^2$. Hence in our case, the formula reduces to

$$\sigma_{sum}(\mathbf{z}; \mathbf{x}) \approx 2\ (1 - r_{0zx}^2) \tag{4.6}$$

If we substitute for $\sigma_{sum}(\mathbf{z}; \mathbf{x})$ the relationship between the different types of correlations is

$$r_{1x} \approx 2\ r_{0zx}^2 - 1 \tag{4.7}$$

and conversely

$$r_{0zx} \approx 1/4\ (r_{1x} + 1)^{1/2}. \tag{4.8}$$

Since high autocorrelations correspond to smooth series optimal nonlinear transforms derived under sum filter remove high frequency components. Consequently, optimal scaling works into the same direction as the sum operation itself. The optimal quantifications will be found such that smoothest possible input series emerges. Of course, this also results in even smoother output series.

TABLE 4.1 Artificial series A (Read across and down)

1	1	1	3	3	3	2	2	2	3	3	1	3	2	1
3	2	3	1	3	3	3	3	2	2	2	2	3	3	2
3	2	3	2	2	3	1	2	3	2	3	1	3	2	2
2	1	1	1	3										

TABLE 4.2 Artificial series B (Read across and down)

1	1	1	2	2	2	3	3	3	2	2	1	2	3	1
2	3	2	1	2	2	2	2	3	3	3	3	2	2	3
2	3	2	3	3	2	1	3	2	3	2	1	2	3	3
3	1	1	1	2										

As an illustration of what the sum filter does when applied to categorical data consider the artificial categorical series A and B which are plotted in Figure 4.3. They both consists of 50 observations in three categories. The categories have been assigned the category numbers 1, 2 and 3. The (first order) autocorrelation of series A is -0.03, so it is near white noise. The series are given in Table 4.1 and Table 4.2.

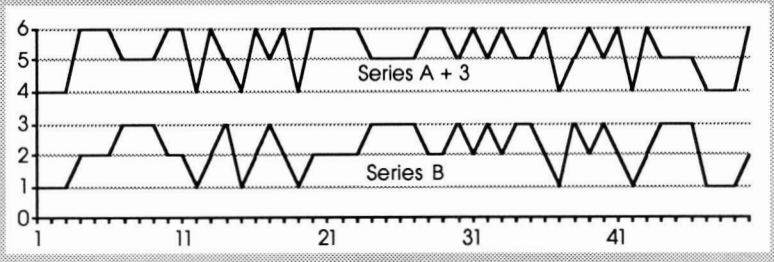


Figure 4.3 Artificial series A and B

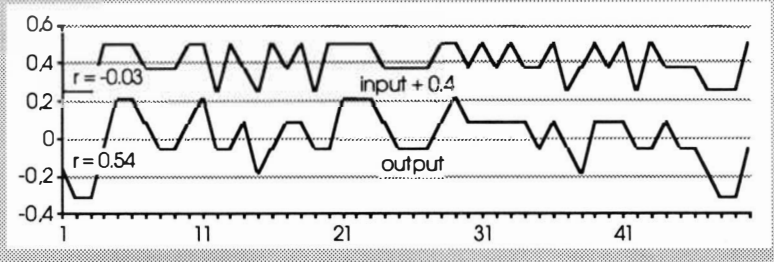


Figure 4.4 Sum filter: numerical

For the numerical case, where the category quantifications are a linear function of the category numbers, the input and the output of the sum filter are plotted in Figure 4.4. It can be seen that, except for scaling, the input series is equal to series A. The output series is proportional to the result we obtain if we had used the direct sum filter without any minimization. It is a smoother version of series A. See for example time points 29 till 33. The filter has flattened out fast fluctuations there. The autocorrelation rises from -0.03 to 0.54, a rather spectacular change that however is impossible to detect by visual inspection alone. This change illustrates that the autocorrelation may change dramatically under even a very simple operation like addition of two adjacent values. It also shows that we can make something (an autocorrelated series) from nothing (a white noise series).

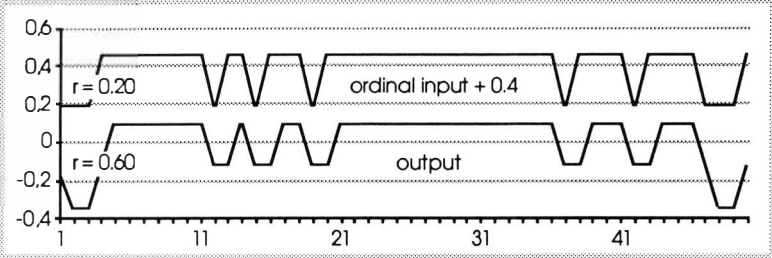


Figure 4.5 Sum filter: ordinal

Things become quite different if we allow for nonlinear transformations of the data. For example, in the ordinal case we require the relationship between the category numbers and the quantified categories to be monotone rather than linear. The result of minimizing (4.2) under this relaxed condition is depicted in Figure 4.5. The input series is now substantially different from the categorical series. The plot shows a somewhat peculiar pattern of long sequences of no change. This is caused by the minimization procedure, which has given categories 2 and 3 equal weights. It is smoother than the numerical input series. Its autocorrelation is now 0.20. The same holds for the output series that has an autocorrelation of 0.60.

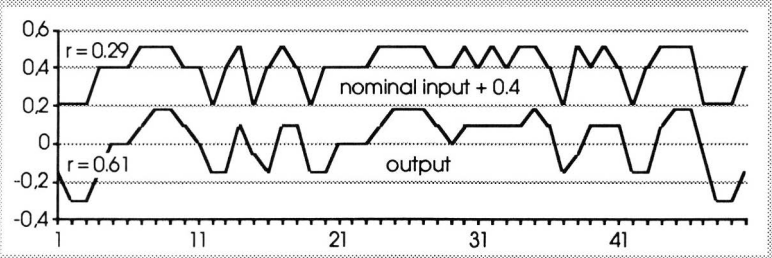


Figure 4.6 Sum filter: nominal

It is very hard to recognize series A from the shape of the nominal input series plotted in Figure 4.6. The main reason is that the categories 2 and

3 have been reversed by the minimization procedure, so that the peaks now appear at other places. Judging from the autocorrelation of 0.29, the categories are now in "smoothest order". A nice property of the nominal sum filter transform is that it recovers the smoothest category order irrespective of the coding of the original data.

The category quantifications are plotted against their original category numbers in Figure 4.7. As required, the quantifications of the numerical option are located on a straight line. For the ordinal case, they are monotone in the category numbers, and in the nominal case the quantifications are free.

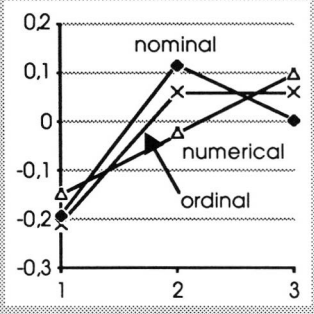


Figure 4.7 Category weights

TABLE 4.3 Analysis results for the sum filter

	r_{1x}	r_{1z}	r_{0zx}	r_{1zx}	σ_{sum}	$2-r_{0zx}^2-r_{1zx}^2$	$1-r_{1x}$
numerical	-0.0347	0.5374	0.6987	0.6829	1.0363	1.0454	1.0347
ordinal	0.2041	0.5992	0.7776	0.7739	0.7954	0.7987	0.7959
nominal	0.2908	0.6069	0.8034	0.8033	0.7092	0.7092	0.7092

Table 4.3 provides some statistics of the three analyses. The entries confirm the relationships $\sigma_{sum}(\mathbf{z}; \mathbf{x}) \approx 1 - r_{1x} \approx 2 - r_{0zx}^2 - r_{1zx}^2$. We conclude that optimal scaling tends to amplify the low pass properties of the sum filter.

4.3 Difference filter

The *difference filter* $\mathbf{z} = \mathbf{x} - \mathbf{B}\mathbf{x}$ works exactly the other way around. This filter differs from the sum filter only in its sign of the last term, but, as we will see below, the results these two filters produce are very dissimilar. Differencing is often used to eliminate trend from the series so that the filtered series $\mathbf{z}$ approaches stationarity.

We define the difference filter for categorical series as the problem of minimizing

$$\sigma_{dif}(\mathbf{z}; \mathbf{x}) = ssq(\mathbf{z}, \mathbf{x}) + ssq(\mathbf{z}, -\mathbf{B}\mathbf{x}) \tag{4.9}$$

over $\mathbf{x} = \mathbf{G}\mathbf{y}$ and $\mathbf{z}$ under normalizations $\mathbf{1}'\mathbf{z} = \mathbf{1}'\mathbf{x} = 0$ and $\mathbf{z}'\mathbf{z} = 1$. If $\mathbf{x}$ is continuous, then minimizing (4.9) is equivalent to setting $\mathbf{z} = 0.5(\mathbf{x} - \mathbf{B}\mathbf{x})$, provided that the normalization constraints are fulfilled. The resulting output $\mathbf{z}$ is just a standardized version of the direct first differences filter output. In other cases, when also categories must be quantified, we can iterate over $\mathbf{x}$ and $\mathbf{z}$. The filter is an example of a high pass filter. This means that it extracts fast fluctuations and removes low frequency components such as a linear trend.

The effect of minimizing (4.9) on the correlation between $\mathbf{x}$ and $\mathbf{B}\mathbf{x}$ is exactly the opposite of that of the sum filter. Let $\mathbf{z}^* = 0.5(\mathbf{x} - \mathbf{B}\mathbf{x})$ then

$$\begin{aligned}\sigma_{dif}(\mathbf{z}; \mathbf{x}) &= 2\, ssq(\mathbf{z}^*, \mathbf{z}) + ssq(\mathbf{z}^*, \mathbf{x}) + ssq(\mathbf{z}^*, -\mathbf{B}\mathbf{x}) \\ &= 2\, ssq(\mathbf{z}^*, \mathbf{z}) + 0.5\, ssq(\mathbf{x}, -\mathbf{B}\mathbf{x}).\end{aligned} \tag{4.10}$$

The corresponding approximate relationship between the first-order autocorrelation of $\mathbf{x}$, denoted by r_{1x}, and $\sigma_{dif}(\mathbf{z}; \mathbf{x})$ becomes

$$\sigma_{dif}(\mathbf{z}; \mathbf{x}) \approx 1 + r_{1x} \tag{4.11}$$

i.e. differencing a series, while simultaneously searching for an optimal nonlinear transform of $\mathbf{x}$, comes down to minimizing the first-order

autocorrelation of $\mathbf{x}$. If we interpret a low autocorrelation as indicative of the roughness of a series, we see again that optimal nonlinear transforms amplify the characteristics of the filter. Quantifications will be found such that low frequency components are removed as much as possible. The relationship

$$\sigma_{dif}(\mathbf{z}; \mathbf{x}) \approx 2 - r_{0zx}^2 - r_{1zx}^2 \tag{4.12}$$

is identical to that of the sum filter.

Differencing series A turned out to yield almost identical transformations under the numerical, the ordinal and the nominal options. In order to show the effect of quantification, series A is recoded into series B that is somewhat smoother. Series B is also plotted in Figure 4.3. We saw that reversing categories 2 and 3 leads to a smoother A-series, so the categories were recoded by $1 \leftarrow 1$, $2 \leftarrow 3$ and $3 \leftarrow 2$. The first lag autocorrelation r_{1x} of the B-series is 0.28.

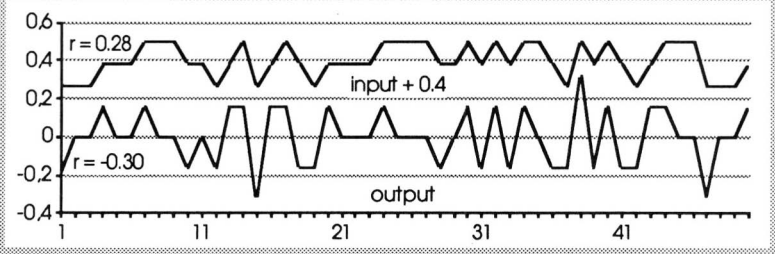


Figure 4.8 Difference filter: numerical

The results of analyzing series B using the difference filter are plotted in Figures 4.8, 4.9 and 4.10. For the numerical case, the output series is scaled according to $\mathbf{1}'\mathbf{z} = 0$ and $\mathbf{z}'\mathbf{z} = 1$, and it is, except for scaling, equal to the direct differences of the categorical series. In the ordinal case, the categories 2 and 3 have been tied to an equal quantification. The input series now displays relatively long sequences of no change, but nevertheless it is less smooth than the numerical series. Its autocorrelation

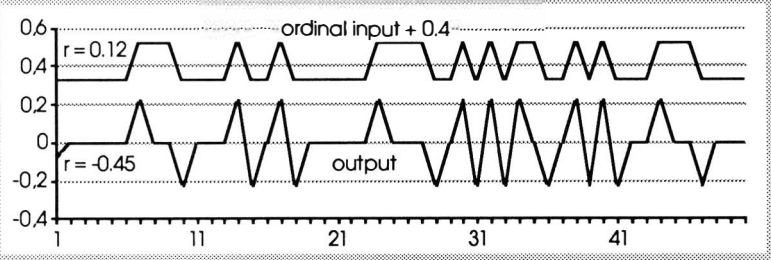


Figure 4.9 Difference filter: ordinal

r_{1x}(ord) = 0.12 is lower than r_{1x}(num) = 0.28. The output series follows more or less the same pattern. It is also more rough than its numerical counterpart (r_{1z}(num) = -0.30 and r_{1z}(ord) = -0.45).

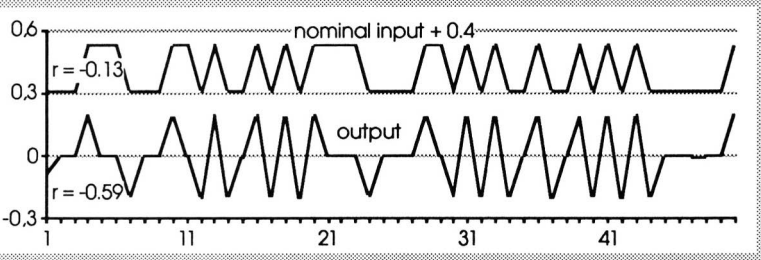


Figure 4.9 Difference filter: nominal

The increase of roughness in both series is amplified in the nominal case. Here, we also drop the restriction that the mapping function between the category numbers and quantification should be monotone. The results are represented in Figure 4.10. Both categories 1 and 3 have obtained (almost equal) positive quantifications, while the scaling of category 2 is negative, so it is very hard to recognize the original categorical series from its nominal version. Both input and output are substantially more rough than those from the previous plots (respectively. r_{1x}(nom) = -0.13 and r_{1z}(nom) = -0.59).

For the sake of completeness, various statistics of the three analyses are listed in Table 4.4, and the quantifications are plotted against their category numbers in Figure 4.11. All in all, it appears that optimal scaling tends to reinforce the high pass characteristics of the difference filter.

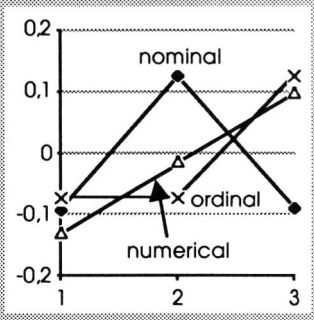


Figure 4.11 Category weights

TABLE 4.4 Analysis results for the difference filter

	r_{1x}	r_{1z}	r_{0zx}	r_{1zx}	σ_{dif}	$2-r_{0zx}^2-r_{1zx}^2$	$1+r_{1x}$
numerical	0.2805	-0.2973	0.5999	-0.5993	1.2807	1.2809	1.2805
ordinal	0.1207	-0.4465	0.6650	-0.6567	1.1215	1.1264	1.1207
nominal	-0.1294	-0.5880	0.7561	-0.7375	0.8691	0.8845	0.8706

4.4 Exponential smoothing filter

The sum and difference filters are both examples of linear filters. We now discuss a nonlinear variety, namely the *exponential smoothing filter* defined by $\mathbf{z} = f\,\mathbf{B}\mathbf{z} + \mathbf{x}$ for a given f. The characteristic feature of the filter is that the output $\mathbf{z}$ depends on previous outputs $\mathbf{B}\mathbf{z}$. See Figure 4.12 for a schematic representation of the filter.

If we solve recursively for $\mathbf{x}$ the filter is equivalent to

$$\mathbf{z} = \mathbf{x} + f\,\mathbf{B}\mathbf{x} + f^2\,\mathbf{B}_2\mathbf{x} + \ldots + f^{N-1}\,\mathbf{B}_{N-1}\mathbf{x}, \tag{4.13}$$

which displays $\mathbf{z}$ as an exponentially weighted combination of the past

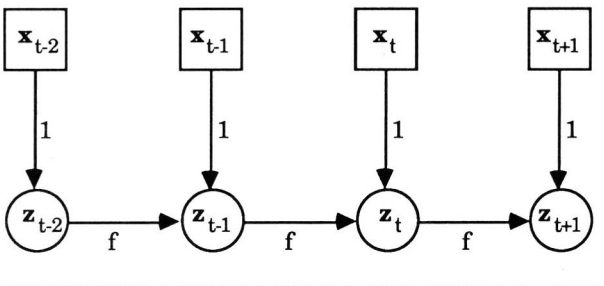


Figure 4.12 Scheme of the exponential filter

values of **x**. If -1 < f < 1 the influence of observations further away in time will diminish exponentially. We define the corresponding loss function for categorical data as

$$\sigma_{exp}(\mathbf{z}; \mathbf{x}) = \text{ssq}(\mathbf{z}, f\mathbf{Bz}) + \text{ssq}(\mathbf{z}, \mathbf{x}), \tag{4.14}$$

which must be minimized for a given f over **z** and **x** = **Gy** under normalizations $\mathbf{1}'\mathbf{z} = \mathbf{1}'\mathbf{x} = 0$ and $\mathbf{z}'\mathbf{z} = 1$. The exponential filter can be either a high pass, or a low pass filter, depending on the sign of f. If f < 0 it is high pass, for f > 0 it is low pass. If f = 0, then **z** = **x**. The magnitude of the filter coefficient indicates the influence of past observations. If |f| is large then past observations will exert a large influence on the present one, i.e. the filter has a substantial memory.

The correlation properties of the exponential filter are rather straightforward. The term ssq (**z**, f**Bz**) maximizes the first-order autocorrelation r_{1z} of **z**, while the component ssq (**z**, **x**) maximizes the correlation r_{0zx} between **z** and **x**. We may thus interpret the filter output as a com- promise between the data at time t and the past output values at times t-1, t-2 and so on. The output **z** can be seen as a latent variable that represents both the contemporary data **x** and its own past **Bz**. The ratio between past and present is modelled by the weight f.

Applying the correlation decomposition theorem (2.15) to ssq ($\mathbf{z}$, f$\mathbf{Bz}$) and ssq ($\mathbf{z}$, $\mathbf{x}$) we find that for $-1 \leq f \leq 1$

$$\text{ssq}(\mathbf{z}, f\mathbf{Bz}) = 1 + f^2 - 2 f r_{1z} - f^2 \mathbf{z}'\mathbf{B}_{N-1}\mathbf{B}_{N-1}'\mathbf{z} \tag{4.15}$$

$$\text{ssq}(\mathbf{z}, \mathbf{x}) = 1 + \mathbf{x}'\mathbf{x} - 2 (\mathbf{x}'\mathbf{x})^{1/2} r_{0zx}. \tag{4.16}$$

After convergence it is true that $\mathbf{x}'\mathbf{x} = r_{0zx}^2$ (cf. Gifi, 1981, 96). Disregarding the end effects we then obtain the approximate relationship

$$\sigma_{exp}(\mathbf{z}; \mathbf{x}) \approx 2 + f^2 - 2 f r_{1z} - r_{0zx}^2 \tag{4.17}$$

for the exponential smoothing filter. For example, if we choose f = 0, then $\mathbf{z} = \mathbf{x}$ and $\sigma_{exp}(\mathbf{z}; \mathbf{x}) \approx 1$. If we choose $f < 0$, then r_{1z} will become small, preferably negative.

The exponential filter works out very nicely for categorical data analysis. In contrast to the linear filters described before, the minimum least squares solution for the numerical case is not proportional to the corresponding direct recursive solution. It will be shown below that the minimization solution has superior correlational properties.

We analyzed series A under numerical, ordinal and linear transformation functions with f chosen as f = 0.8. The resulting series are plotted in Figures 4.13, 4.14 and 4.15.

For the numerical case, three series are plotted: one input series (used by both a direct recursive filter and a least squares filter), and two output series. The least squares filter is defined by (4.14). Judging from the autocorrelations, the least squares output is smoother than the direct filter output series. The respective autocorrelations are 0.84 for the least squares filter and 0.73 for the direct filter. Also, the cross correlation between $\mathbf{z}$ and $\mathbf{x}$ is higher for the least squares solution: r_{0zx}(num) = 0.70 for least squares filter and r_{0zx}(num) = 0.62 for the other. Thus, it appears that the least squares filter not only yields a smoother output,

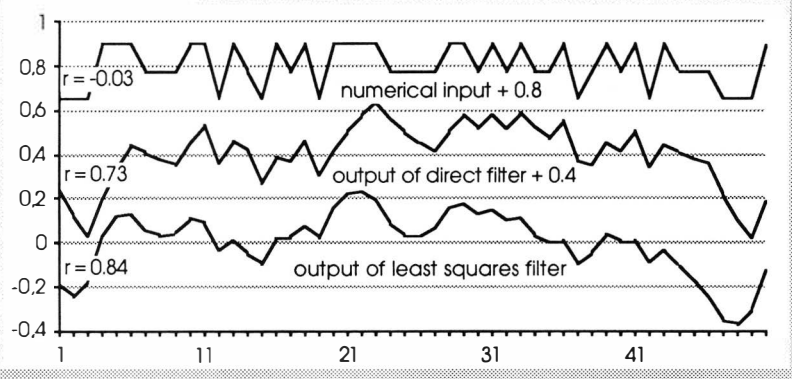


Figure 4.13 Exponential filter: numerical

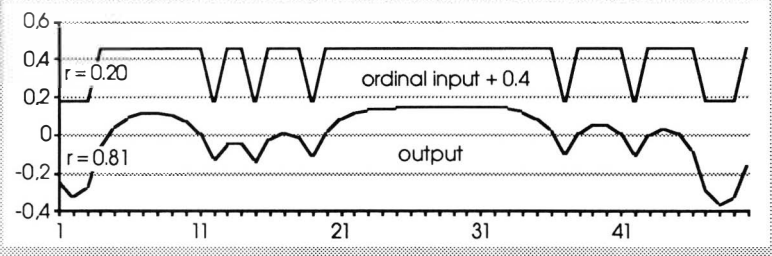


Figure 4.14 Exponential filter: ordinal

but also better represents the input.

For ordinal data, we obtain a peculiar bumpy output series which closely follows the optimally scaled input (r_{0zx} = 0.81). Note that the ordinal input series is identical to its sum filter equivalent. It seems that it is the smoothest possible input series under ordinal transforms.

In the nominal case, the filtering has been carried through so far that the output series resembles a typical numerical stock market series. Suppose for example that the categories of the input stand for 1 = oil price

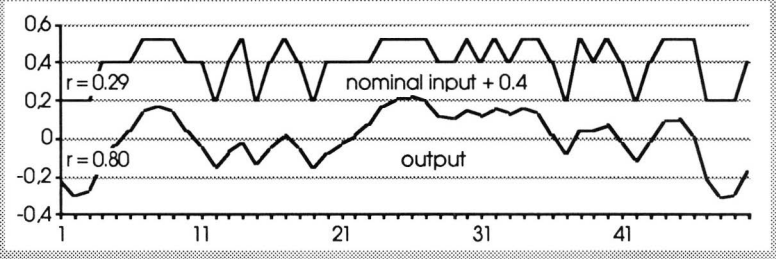


Figure 4.15 Exponential filter: nominal

goes down, 2 = oil price goes up, 3 = oil price is constant, then the output series could be a hypothetical stock market price of a major oil company share. The output series can then be seen as absorbing the relatively fast fluctuations in oil pricing. Again, the input series becomes equivalent to its sum filter counterpart.

The quantifications, plotted on the right, are proportional to the sum filter quantifications. The exponential and the sum filter have in common that both maximize the input autocorrelation. In reverse, the same relation holds between the exponential filter with f = -0.8 and the difference filter. It seems that the quantification function has only two basic shapes: one for positive and one for negative autocorrelations.

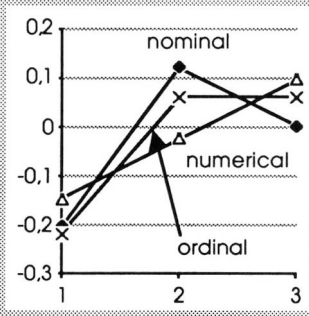


Figure 4.16 Category weights

Table 4.5 lists a number of analysis results. We can use these values to check the loss value approximation formula 4.15. For example, $\sigma_{exp}(num) = 0.8003 \approx 2 + (0.8)^2 - 2 \times 0.8 \times 0.8372 - (0.7014)^2 = 0.8085$.

TABLE 4.5 Analysis results for the exponential filter

	r_{1x}	r_{1z}	r_{0zx}	σ_{exp}	$2+f^2-2fr_{1z}-r_{0zx}^2$
numerical	-0.0347	0.8372	0.7014	0.8003	0.8085
ordinal	0.2041	0.8095	0.8132	0.6694	0.6835
nominal	0.2907	0.7965	0.8462	0.6305	0.6495

4.5 Autoregression with optimal scaling: lag-1 predictor

In the preceding section we have considered the effect of optimal scaling on the autocorrelation properties of three simple filters, and we have seen that scaling amplifies the band pass properties of the filter. So far, it was assumed that the system dynamics were known beforehand, i.e. the filter coefficients are known constants. Typically, in soft sciences like economics and psychology one often does not know these system parameters. This makes it worthwhile to investigate a more general situation in which the system parameters are not known a priori, but depend on the data at hand. We then deal with regression models with lagged predictor variables.

As we will see below, the mathematics of the univariate filters we discussed are intimately connected with those of regression, but the interpretations of the vectors $\mathbf{z}$ and $\mathbf{x}$ differ. In filtering, $\mathbf{x}$ is the system input and $\mathbf{z}$ is the system output. In (auto)regression, $\mathbf{x}$ is the dependent variable, which is to be predicted by the independent variable $\mathbf{Bx}$ of past observations, and $\mathbf{z}$ is a canonical variate somewhere in between $\mathbf{x}$ and $\mathbf{Bx}$ which in itself has no direct function.

In this section, we restrict our attention to the simple first-order autoregressive model, also known as the lag-1 predictor model,

$$\mathbf{x} = \mathbf{Bx}b_1 + \mathbf{e} \tag{4.18}$$

where $\mathbf{x}$ is an N × 1 vector of observations on N time points. Vector $\mathbf{Bx}$ is the first lag of $\mathbf{x}$ and $\mathbf{e}$ is a residual vector $\mathbf{x} - \mathbf{Bx}b_1$.

The lag-1 predictor problem $\mathbf{x} = \mathbf{Bx}b_1 + \mathbf{e}$ can be translated into the least squares problem of minimizing

$$\sigma_{ard}(\mathbf{x}; b_1) = ssq\,(\mathbf{x}, \mathbf{Bx}b_1) \tag{4.19}$$

over b_1 and $\mathbf{x} = \mathbf{G}\mathbf{y}$ under constraints $\mathbf{1}'\mathbf{x} = 0$ and $\mathbf{x}'\mathbf{x} = 1$. The "ard" subscript stands for "AutoRegression Direct". It is not difficult to solve this problem directly by an iterative algorithm, but in order to stay in line with previous sections, we prefer to formulate the regression problem in terms of canonical analysis. Suppose we have two sets of variables. The first set contains the dependent variable $\mathbf{x}$, and the second set consists of the independent variable $\mathbf{Bx}$. We may then maximize the canonical correlation between $\mathbf{x}$ and $\mathbf{Bx}$ by minimizing

$$\sigma_{arc}(\mathbf{z}; \mathbf{x}; a_0; a_1) = ssq\,(\mathbf{z}, \mathbf{x}a_0) + ssq\,(\mathbf{z}, \mathbf{Bx}a_1) \tag{4.20}$$

over $\mathbf{z}$, $\mathbf{x}$, a_0 and a_1 under normalizations $\mathbf{1}'\mathbf{z} = \mathbf{1}'\mathbf{x} = 0$ and $\mathbf{z}'\mathbf{z} = \mathbf{x}'\mathbf{x} = 1$. Subscript "arc" refers to "AutoRegression Canonical".

It can be seen that minimizing σ_{ard} and σ_{arc} are equivalent problems if we partition (4.20) as

$$\begin{aligned}\sigma_{arc}(\mathbf{z}; \mathbf{x}; a_0; a_1) \quad &= 2\ ssq\,(\mathbf{z}^*, \mathbf{z}) + 0.5\ ssq\,(\mathbf{x}a_0, \mathbf{Bx}a_1) \\ &= 2\ ssq\,(\mathbf{z}^*, \mathbf{z}) + 0.5\ a_0^2\ ssq\,(\mathbf{x}, \mathbf{Bx}a_1/a_0),\end{aligned} \tag{4.21}$$

where $\mathbf{z}^* = 0.5\,(\mathbf{x}a_0 + \mathbf{Bx}a_1)$. The regression weight b_1 is of course equal to r_{1x}, the first-order autocorrelation of the optimally scaled series $\mathbf{x}$.

It is of quite some interest to note that, thanks to the normalizations we used, the sum and difference filters are exactly identical to the regression problem. In fact, if $r_{1x} \geq 0$ then

$$\sigma_{sum}(\underline{z}; \underline{x} \mid \underline{z}'\underline{z} = 1) = \sigma_{arc}(\mathbf{z}; \mathbf{x}; a_0; a_1 \mid \mathbf{z}'\mathbf{z} = \mathbf{x}'\mathbf{x} = 1), \quad (4.22)$$

and if $r_{1x} < 0$ then

$$\sigma_{dif}(\underline{z}; \underline{x} \mid \underline{z}'\underline{z} = 1) = \sigma_{arc}(\mathbf{z}; \mathbf{x}; a_0; a_1 \mid \mathbf{z}'\mathbf{z} = \mathbf{x}'\mathbf{x} = 1). \quad (4.23)$$

The vectors $\underline{x}$ and $\underline{z}$ correspond to the filter solutions. The relation between the regression model and the sum filter is easy to see if one thinks of the unstandardized filter input $\underline{x}$ as a scaled down version of the regression variable $\mathbf{x}$, i.e. $\underline{x} = \mathbf{x}a_0$. Since $\mathbf{x}'\mathbf{x} = 1$ it follows that $\underline{x}'\underline{x} = a_0^2$.

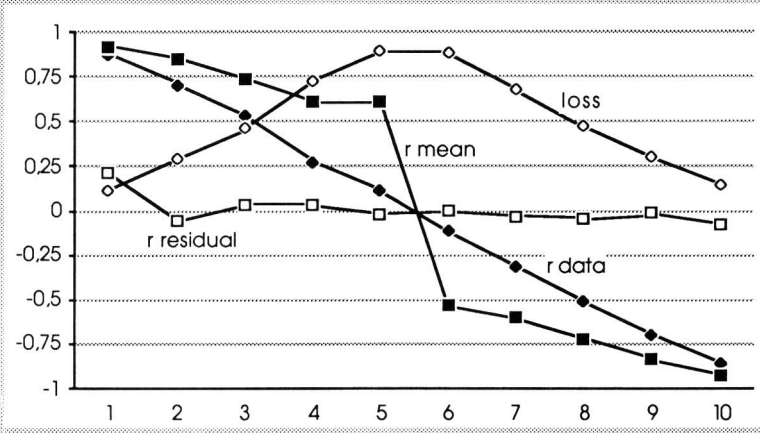


Figure 4.17 Analysis results of 10 simulated series

The sign of the autocorrelation r_{1x} determines whether a sum filter analysis or a difference filter analysis will be used. Figure 4.17 contains the results of a regression analysis on 10 simulated series, each with 50 observations. The series contain varying amounts of autocorrelation. The autocorrelations of the 10 series are labeled by "r data", and they gradually decrease from 0.90 to -0.90. The figure also plots the loss values σ_{arc} for the 10 runs. These values form a bell-shaped curve, indicating that series with extreme autocorrelations fit the autoregressive model much

better. A third statistic, labeled "r mean", is the autocorrelation r_{1z} of the canonical variate $\mathbf{z}$. The values for runs 1 to 5 (i.e. the runs that correspond with positive autocorrelations in the data) are all positive, while the opposite is true for runs 6 to 10. This indicates that on the left side, autoregression behaves like the sum filter, and that on the right side it is equal to the difference filter. There is a jump from +0.50 to -0.50 in r_{1z} at point $r_{1x} = 0$. A fourth statistic, labeled "r residual", is the autocorrelation in the least squares residuals $\mathbf{e} = \mathbf{x} - \mathbf{B}\mathbf{x}b_1$. Except for run 1, these values are very close to zero, meaning that the regression model adequately describes the (first-order) time dependencies in the data.

It is now also possible to investigate the effect of optimal scaling on the regression solution. The process of minimizing the autoregressive loss σ_{arc} comes down to finding the lowest point on the bell curve of losses. Since the smaller loss values are located in the tails, the first-order autocorrelation of the optimally scaled data is *either* maximized *or* minimized. Thus, optimal scaling leads to either an optimally smooth or an optimally rough data transformation. The direction in which the data will be transformed is determined by the location of the global minimum of the loss value. If this minimum lies on the left side of the loss curve, then the data will be smoothed. If it lies on the right, then the optimal transformation will make the series more erratic.

Before presenting an example, we conclude with a word on the interpretation of the autoregressive weight b_1. If the weight turns out to be positive, then we are dealing with the sum filter. A negative weight corresponds to the difference filter. In the univariate case, the weight is often close to the first-order autocorrelation coefficient of the data. The magnitude of the regression weight indicates how much a series depends on its own past. The square of the weight may be interpreted as the proportion of variance that is being explained by the autoregression. Note that we have not made the assumption $-1 < b_1 < +1$ that is usually associated in the first-order autoregressive Box-Jenkins model. However, in general the weight turns out to be located in that interval. The weight may also be used in producing the least squares forecast $b_1 x_t$.

4.6 An example of the lag-1 predictor analysis

The Box and Jenkins (1976) series D consists of 310 observations of hourly viscosity readings of a chemical process. The series takes on values between 7.4 and 10.4. Box and Jenkins found that both an autoregressive AR(1) and a first difference model fit the series rather well and they estimated the regression weight for the AR(1) as $b_1 = 0.87$.

We fitted equation (4.20) to series D using linear and monotone transformations of the data. In the linear analysis we found a minimum loss of $\sigma_{arc} = 0.1385$ with $a_0 = a_1 = 0.96$ and $b_1 = 0.87$. As expected, the regression weight b_1 is thus equal to the one obtained by Box and Jenkins (1976, 239) in this case. The value of the Box-Pierce statistic[1] is equal to $\psi_{int} = 10.21$ with df = 24. This value is not significant, which implies that the residuals may be regarded as white noise, and so we may conclude that the model provides a reasonable description of the data.

We also analyzed the series under an optimal monotone transformation of the data. The 310 observations assume 26 distinct values, so no order information is lost by recoding the series into 26 consecutive categories. The frequency distribution over the 26 categories is plotted Figure 4.18, which shows that the majority of observations falls into the right side of the histogram. The results for the ordinal analysis are $\sigma_{arc} = 0.0975$ with $a_0 = a_1 = 0.98$ and $r_{1x} = b_1 = 0.91$. Since there are more free parameters, the fit between the transformed series and the model is somewhat better compared to the linear analysis, although this change is not spectacular since we already started with a high autocorrelation of $r_{1x} = 0.87$.

Note: A popular way to check the adequacy of the model is to perform a *Portmanteau test*. This test examines whether the model leaves any time dependent information in the residuals $\mathbf{e} = \mathbf{x} - \mathbf{B}\mathbf{x}b_1$. For a correctly specified ARMA(P,Q) model, the Box-Pierce statistic

$$\psi = N \Sigma_s r_{se}^2, \qquad s=1...S$$

is approximately chi-squared distributed with df = S - P - Q degrees of freedom. Parameter S is the number of residual autocorrelations included in the computation. It is typically chosen as 25.

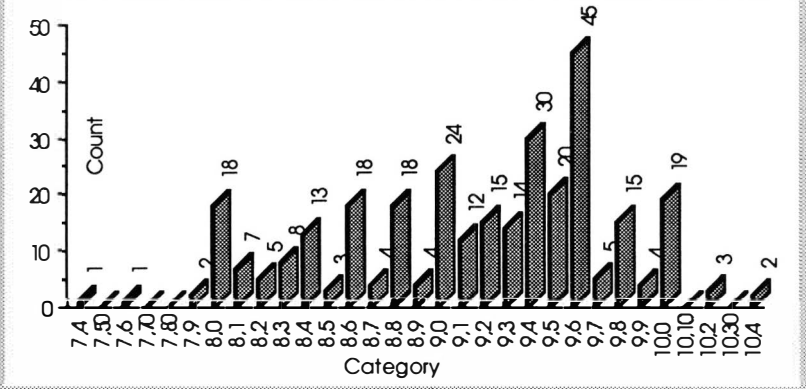


Figure 4.18 Category frequencies for the Box-Jenkins series D

Differences between the ordinal and the linear analysis can best be seen from the transformation plot, which graphs the observed values against the optimally scaled counterparts. Figure 4.19 is such a plot. This plot clearly demonstrates the monotonic increasing score pattern that always accompanies an ordinal analysis. In a standard numerical analysis all these scores are located on a straight line. The transformation tends to cluster the extremes of the scale. This effect is especially visible on the lower side: scores 7.2 to 8.2 obtain identical quantifications. This implies that given a first-order autoregressive model, the extremes of the scale do not discriminate very

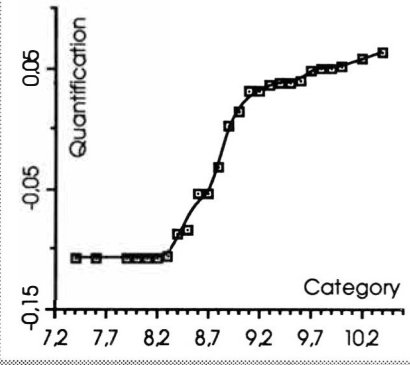


Figure 4.19 Transformation plot for the ordinal analysis

much among the measurements, i.e. it does not matter whether we observe a score of 7.2 or a score of 8.2. A possible interpretation of the phenomenon is that physical process moves back and forward between two points of attraction, located at about 8.2 and 9.6. An alternative explanation is the optimal scaling blows up the agreement at the end of the scale in order to make the autocorrelation r_{1x} larger. These effects are further demonstrated in Figures 4.20 and 4.21.

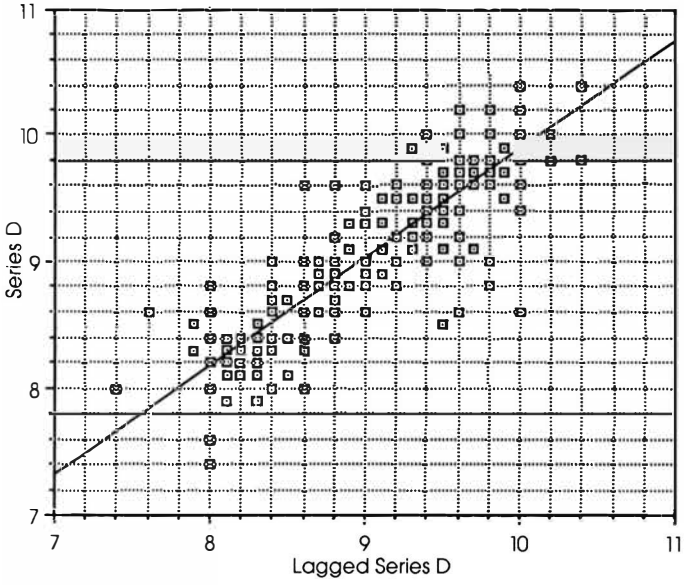


Figure 4.20 Scattergram of series D versus lagged series D

Figure 4.20 is a combination of a scatter plot and a contingency table of the Box-Jenkins series D (Box & Jenkins, 1976). Each of the axes contains 40 categories. The first-order correlation ($r_{1x} = 0.87$) of the series can thus be completely represented in the plot. Since the series is treated at an interval level here, the gridlines are equally spaced on the axes (note: only half of the number of gridlines is actually present in Figure 4.20).

Each point is located on the intersection of two gridlines, and the points indicate how an observation at time t-1 is followed by an observation at time t (start reading at the horizontal axis for time t-1 and pick the value for time t at the vertical axis). If the series is highly autocorrelated the points fall near or on a regression line. The plot becomes a 40 × 40 contingency table, often called a transition table, if we label each point by its number of occurrences.

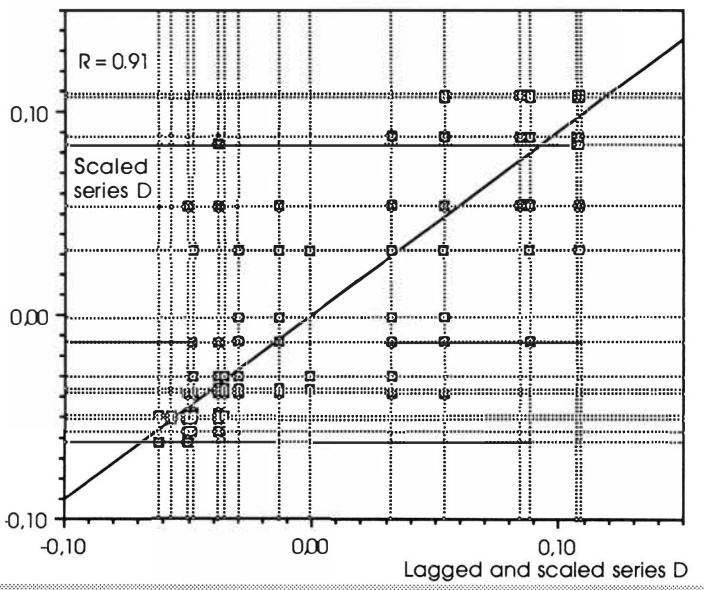


Figure 4.21 Scattergram of optimally scaled series D versus its first lag

Figure 4.21 is identical to Figure 4.20, except for the fact that the inter-category distances have been rescaled by means of the optimal monotone transformation plotted in Figure 4.19. The categories have been shifted simultaneously along both axes. The extreme categories tend to cluster, while the middle categories spread out. The autocorrelation $r_{1x} = 0.91$ is at a maximum for all possible monotone transformations. Even a regres-

sion analysis on the original data with 4 or 5 polynomial degrees does not produce a multiple correlation that exceeds this autocorrelation. According to the Portmanteau test, the residuals do not correlate (the Box-Pierce statistic is 19.54 with df = 24).

It is not easy to say whether the ordinal analysis is "better" than the linear analysis. The differences in terms of explained variance are not very large, however these are likely to grow for less autocorrelated series. For one thing, if the relationship between x_t and x_{t-1} in the data is linear (and most time series models only describe linear relationships), the transformation plot will show a straight line. Here, this is clearly not the case. It is difficult to determine from these data whether the grouping effect indicates a real world physical process, or merely results from a statistical artifact.

4.7 Seasonal autoregression with optimal scaling: lag-p predictor

Until now we have only considered the case in which we have a single predictor variable, and where this predictor variable is the first lag of the series. In this section we study a simple extension: *seasonal autoregression*.

Seasonal autoregression is identical to single first-order autoregression described in section 4.5, except for the choice of the predictor variable. In the first-order model we choose the first lagged variable $\mathbf{Bx}$ as the predictor, in the seasonal model we pick one from the whole range of lagged variables $\mathbf{B}_p\mathbf{x}$ for some $p > 1$. The loss functions of the seasonal and of the first-order model only differ in their subscript of the backshift matrix $\mathbf{B}$, so all results of the preceding section can be readily applied. We only substitute "p^{th} order" for "first-order".

Seasonal models are convenient for describing periodic fluctuations in the data. For example, monthly unemployment figures tend to be related to the month in which they are measured. In this case, one may try to

describe the observations by a seasonal autoregressive model with a 12-month lagged predictor. Seasonal series are characterized by spikes at regular intervals of the autocorrelation plot.

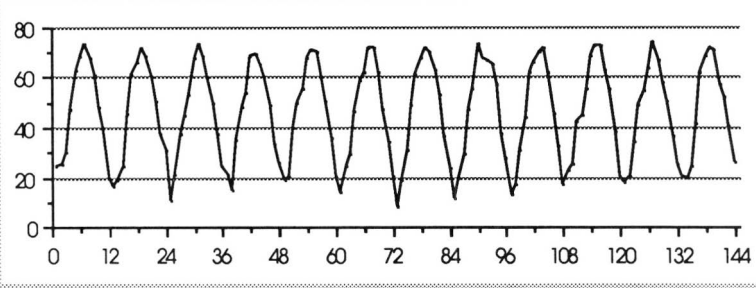


Figure 4.22 IOWA mean monthly temperature for 144 months

A good candidate to illustrate a seasonal model is the Iowa temperature series listed in Cryer (1986). The series records the monthly average temperature in Dubuque, Iowa for 144 consecutive month between January 1964 and Decembre 1975. It is plotted in Figure 4.22 and it clearly demonstrates a cyclical pattern of 12 months.

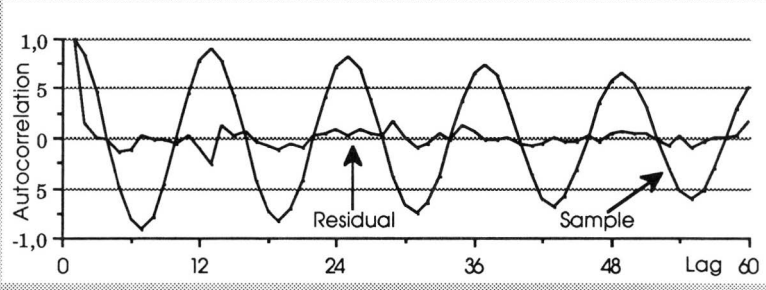


Figure 4.23 Autocorrelations of the IOWA series before and after regression

The autocorrelation plot is also very regular (cf. Figure 4.23). After fitting a 12^{th} lag autoregressive model to the series we found a regression

weight of $b_{12} = 0.93$ and a loss of $\sigma_{arc} = 0.0675$. The data are thus fitted very well by to the equation

$$\mathbf{x} = 0.93\, \mathbf{B}_{12}\mathbf{x}. \tag{4.24}$$

The autocorrelations of the residuals $\mathbf{e} = \mathbf{x} - 0.93\, \mathbf{B}_{12}\mathbf{x}$, computed with the first 12 elements deleted, are also plotted in Figure 4.23. They are all close to zero: the Box-Pierce statistic is 27.8 with df = 24, so the residuals do not contain any time relevant information anymore.

4.8 Multiple autoregression with optimal scaling

A second extension of the lag-1 predictor model is to include more than one lagged variable as a predictor. We are then dealing with multiple autoregression. Examples of multiple autoregressive models are combined first-order and seasonal autoregressive models and Box-Jenkins AR(P) models, where consecutive lags serve as predictors.

The multivariate autoregressive problem is to predict a time series $\mathbf{x}$ by a linear combination of one or more lags $\mathbf{B}_p\mathbf{x}$, with $p \in \{1 \ldots P\}$, i.e.

$$\begin{aligned} \mathbf{x} &= \mathbf{B}\mathbf{x}b_1 + \mathbf{B}_2\mathbf{x}b_2 + \mathbf{B}_3\mathbf{x}b_3 + \ldots + \mathbf{B}_P\mathbf{x}b_P + \mathbf{e} \\ &= \sum_{p=1}^{P} \mathbf{B}_p\mathbf{x}b_p + \mathbf{e}. \end{aligned} \tag{4.25}$$

Some of the b_p-weights may be restricted to zero if we do not wish to include the p^{th} lag in the analysis. We assume that $\mathbf{x}$ is partitioned into two independent components such that

$$(\sum_{p=1}^{P} \mathbf{B}_p\mathbf{x}b_p)'\mathbf{e} = 0. \tag{4.26}$$

The goal is to maximize the multiple correlation between $\mathbf{x}$ and the predictor variables over the weights b_p and over the category quantifications $\mathbf{y}$ in $\mathbf{x} = \mathbf{Gy}$. This problem can be formulated in terms of canonical analysis as minimizing the least squares loss

$$\sigma_{mar}(\mathbf{z}; \mathbf{x}; a_0; a_p) = ssq\,(\mathbf{z}, \mathbf{x}a_0) + ssq\,(\mathbf{z}, \sum_{p=1}^{P} \mathbf{B}_p\mathbf{x}a_p) \qquad (4.27)$$

over $\mathbf{z}$, $\mathbf{x} = \mathbf{Gy}$, a_0 and a_p ($p = 1 \ldots P$) under normalizations $\mathbf{1}'\mathbf{z} = \mathbf{1}'\mathbf{x} = 0$ and $\mathbf{z}'\mathbf{z} = \mathbf{x}'\mathbf{x} = 1$, and possibly restrictions $a_p = 0$ for some $p > 0$. The "mar" subscript stands for "Multiple AutoRegression". After convergence, the regression weights may be found by applying standard projection techniques to the optimally scaled data.

The question how much and which lags to include in the analysis can be handled in more or less the same way as the iterative Box-Jenkins strategy of model building that is based on autocorrelation functions (ACF) and partial autocorrelation functions (PACF). There are two complications however.

First, we do not require the included lags to be contiguous as in the Box-Jenkins ARMA(P,Q) model. This has consequences for the computation of the PACF, since the variables that are partialled out are not necessarily all lower order lags, so the standard way of computing the PACF's must be adapted.

The second problem is more serious and has to do with the effect of optimal scaling on the autocorrelation function. Because ordinal and nominal transformations of the variables will change the autocorrelation properties of the series, the ACF's may not be comparable across different models that use different optimal transformations, and so the value of using these ACF's as an identification tool is questionable. In the first-order autoregressive case, we found that the first autocorrelation is either minimized or maximized. As we will see below, this property does not hold anymore in higher order models.

TABLE 4.6 Swedish Harvest Index (1749 - 1850)(Read across and down)

3	10	7	7	10	7	7	2	2	7	10	7	2	1	2
2	7	7	7	10	10	7	1	2	7	7	2	7	10	7
10	2	2	2	2	7	2	7	10	9	7	10	9	7	9
9	9	7	7	2	2	2	7	7	9	7	4	7	5	4
9	9	3	2	5	8	9	3	4	3	8	10	8	5	9
9	7	3	9	9	7	5	5	9	7	5	8	7	4	8
7	8	3	5	5	6	4	6	7	8	6	7			

As an example we use the Swedish Harvest Index series listed in MacCleary and Hay (1980). The series records the annual Swedish grain harvest between 1749-1850 on a ten point scale. MacCleary and Hay argue that the series is *not* a time series since by definition they simply rule out any series that is not measured on an interval scale (see pp. 21 and 124). However, for our purposes the series is a perfect example of a categorical time series. It is listed in Table 4.6. It is slightly different from the series listed in MacCleary and Hay since we took the integer fraction of some entries that, for some unclear reason, were not whole numbers. The number of categories is 10.

TABLE 4.7 Analysis results of four multivariate autoregression models

Lag	Autocorrelations				Regression weights			
1	.5038	.5016	.5022	.5015	.5041	.5691	.5630	.5665
2	.1661	.1509	.1532	.1486		-.1344	-.1077	-.1060
3	.0050	-.0128	-.0114	-.0176			-.0433	-.0735
4	-.0112	-.0133	-.0148	-.0127				.0499
5	.0192	.0072	.0073	.0070				
Loss	.4960	.4851	.4835	.4813	.4960	.4851	.4835	.4813
r	.5040	.5150	.5164	.5187	.5040	.5150	.5164	.5187
BP	20.92	17.32	18.38	17.77	20.92	17.32	18.38	17.77
MBP	24.89	20.67	21.91	21.22	24.89	20.67	21.91	21.22

We fitted a hierarchy of models to the series, starting from lag 1 up to 4, under optimal monotone transformations of the data. The results are summarized in Table 4.7. The columns in the first block of the table contain the first five autocorrelations of optimally scaled series for each analysis. These entries are very similar across the four analyses, indicating

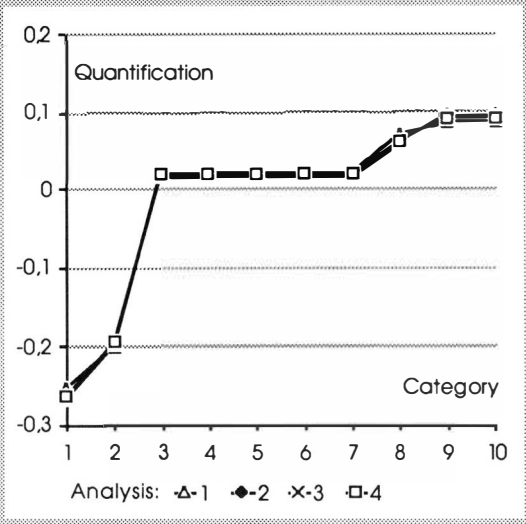


Figure 4.24 Transformation plot: 4 multiple analyses

that all models show up with the same basic data transformation. This is also illustrated in Figure 4.24 which plots the category quantifications against the original category numbers. All analyses find the same large tie-block in the middle categories. The second part of Table 4.7 lists the regression weights under the four models. The first lag turns out to be the most important one in predicting the series. We also see this from the loss values and multiple correlations of the four analyses that do not change very much any more after lag 1 has been included. Finally, the Box-Pierce statistic (BP) and the modified Box-Pierce statistic (MBP) are listed. None of the models leaves any substantial autocorrelation in the residuals, so we conclude that the simple first-order model is adequate here.

TABLE 4.8 Analysis results of four univariate autoregression models

Lag	Autocorrelations				Regression weights			
1	.5038	.4870	.3012	.2276	.5041			
2	.1661	.1877	-.0796	-.0445		.1882		
3	.0050	.0376	-.2113	-.0617			-.2127	
4	-.0112	-.0176	-.1007	-.2574				-.2575
5	.0192	.0341	-.0617	-.1529				
Loss	.4960	.8121	.7881	.7426	.4960	.8121	.7881	.7426
r	.5040	.1880	-.2120	-.2574	.5040	.1880	-.2120	-.2574
BP	21.97	45.01	27.62	35.44	21.97	45.01	27.62	35.44
MBP	26.14	51.54	31.15	41.07	26.14	51.54	31.15	41.07

Obviously, the relatively large influence of the first lag forces the optimal data transformation in a particular direction, so that the first lag will come out even better. This is a kind of natural selection process among the lags: they fight over their influence in the transformation function. If we remove the first lag different transformations will be found. See for example Table 4.8, which contains the results of four one-predictor autoregressions for lags 1 up to 4. The autocorrelations for lags 1 and 2 are rather similar, but those for lags 3 and 4 are entirely different. This is because the second analysis maximizes the second-order autocorrelation, the third maximizes (the absolute value of) the third-order autocorrelation, and so on. The transformation functions for the four analyses are plotted in Figure 4.25.

Optimal transformations are not only data-dependent, as is the standard autocorrelation, but also model-dependent. A good way to proceed seems therefore to work in a hierarchical way. First, we select the most important lags, possibly based on the ACF and PACF of the data treated at an interval level, and fit a model on them. Then by inspecting the autocorrelations of the residual less informative lags may be dropped, and more informative lags may be included, and so on. Working this way, it is unlikely that abrupt changes will appear in the transformation function since we preserve "good" predictors.

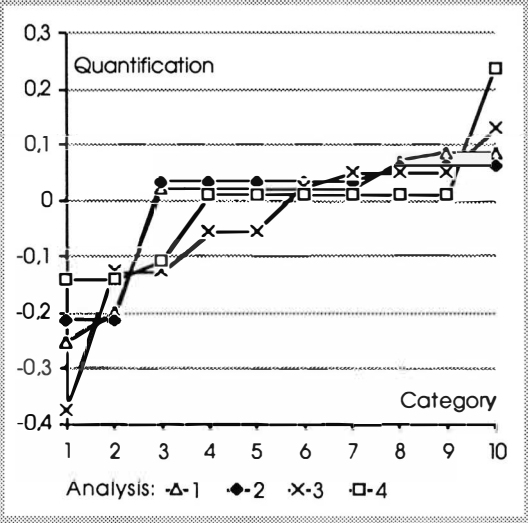


Figure 4.25 Transformation plot: 4 univariate analyses

4.9 Summary and conclusion

We started out to study the effect of optimal scaling on the autocorrelations of the series in the context of a number of simple filtering techniques. Using the correlation partioning theorem given in Chapter 2, we found that the first-order autocorrelation is either maximized or minimized under the optimal transformation. The sum filter and the exponential smoothing filter may be used as smoothers for categorical series, while the difference filter does the opposite action of filtering out low frequency components. For the numerical case, both the sum filter and the difference filter will produce output series that are proportional to the ones yielded by direct summing or differencing, so these filters may be regarded as generalizations of the direct methods to the case of nominal and ordinal series.

The properties of the simple first-order autoregression are almost equivalent to those of the sum and difference filters. It is mainly in the interpretation where both methods differ. The emphasis in filtering is on transforming the input to the output, while in regression analysis it is on modelling the series. The goal of regression analysis is usually to find a model that absorbs the autocorrelations, resulting in regression residuals that are almost white noise.

Next, two straightforward extensions of the first-order model are described. The seasonal autoregressive model is convenient for modelling series with a regular periodic component by just one lagged variable. Multiple autoregression is very much related to the Box-Jenkins AR(P) model and it allows us to describe more complex categorical time series.

A problem associated with optimal scaling in time series analysis is that the autocorrelations are model-dependent, so a major model identification tool like the correlogram is also dependent on the model. For the autoregression problem with optimal scaling there is not yet a proved strategy as to how many and which lags to include in the model.

A common characteristic of the filtering and regression techniques is that all are forms of canonical correlation analysis with lagged variables. The methods are formalized within a least squares framework as loss functions. We will elaborate on this concept in the next chapter. This will deal with multivariate categorical time series.

CHAPTER 5

Multivariate time series analysis with optimal scaling

In contrast to Chapter 4, this chapter deals with the analysis of multivariate time series. We discuss four techniques: intervention analysis, the Box-Tiao canonical transformation, dynamic components analysis and multiset dynamic components analysis. The chapter is divided into four sections. Each section deals with one of the techniques. A section typically starts with a short presentation on the goal of the technique and renders situations in which it can be applied. This is followed by a technical account on the minimization problem, the effect on the correlation box and the effect on the optimal category weights. Finally, we provide an example analysis. It is not the intent of this chapter to provide rigorous definitions. Rather it should give a basic understanding in what way the omnibus function of Chapter 6 can be molded in order to arrive at useful data analysis techniques.

5.1 Intervention analysis

A common goal in time series analysis is to determine whether some external event influences the level or shape of an observed series. Box and Tiao (1975) phrase the problem as follows: "Given a known intervention, is there evidence that change in the series of the kind expected actually occurred, and, if so, what can be said of the nature and magnitude of the change?".

A typical example comes from the N = 1 research designs that are popular in clinical psychology. Here the question is whether a therapeutic treatment has an effect on the client. A number of intervention schemes have been proposed, the most usual of which is the A-B design. This design specifies that during N_1 time points treatment A has been administered and during N_2 time points treatment B is in effect. The major methodological foundations of what is called *interrupted time series experiments* or *intervention analysis* have been laid down by Campbell and Stanley (1963) and Box and Tiao (1965, 1975). Many books on the topic are available, including those of Glass, Willson and Gottman (1975), Hersen and Barlow (1976) and Cook and Campbell (1979). The scientific platform for this kind of applications in the behavioral sciences is the *Journal of Applied Behavior Analysis*.

The main worry in intervention analysis is serial dependency in the observed series. Serial dependency invalidates the use of the t-test for checking the difference between the pre-intervention mean and the post-intervention mean. As a solution one may fit an ARMA model to the series and derive the residual, which ideally should be free of serial dependency. Subsequently, the change in level can be assessed by applying traditional cross-sectional procedures like the t-test on this residual.

In the present framework the treatment effect is viewed as prescribing a second variable with a limited number of states, i.e. a categorical variable. We call it the intervention variable. Figure 5.1 portrays three hypothetical intervention variables.

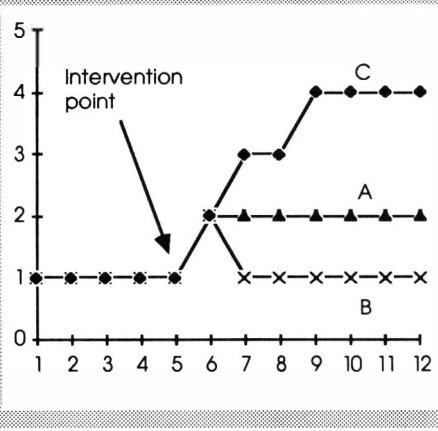


Figure 5.1 Hypothetical intervention variables

The simplest possible intervention variable is shown as series A. It consists of only two states and it specifies that the time series immediately changes at the intervention point and that the change is permanent. For an abrupt but temporary change the indicator is like series B. Series C represents a gradual constant change. The intervention variable defines the intervention model. The data analyst is expected to base the choice for a specific model on the shape of the intervention.

Let us denote the intervention variable by $\mathbf{x}_i$ and the observed time series by $\mathbf{x}_0$. We define the intervention model as

$$\mathbf{x}_0 = \mathbf{x}_i b + \mathbf{e}, \tag{5.1}$$

where $\mathbf{e}$ is assumed to be independent of the predictors and not autocorrelated. Vector $\mathbf{y}_i$ in $\mathbf{x}_i = \mathbf{G}_i\mathbf{y}_i$ contains the k_i category quantifications of the intervention variable. The entries of the product vector $\mathbf{y}_i b$ are simply the means of the corresponding elements of $\mathbf{x}_0$ and they indicate which treatments or intervention levels exert most influence on the analysis. Larger absolute values correspond to larger influences. In principle, differences between these means can be examined with a t-test or an F-test, but if the residuals $\mathbf{e}$ are autocorrelated these differences will be overestimated, which eventually leads to erroneous conclusions. Therefore, in intervention analysis one often includes a "noise model" in the regression equation. This noise model can be a plain first-order autoregressive

model, but also a complicated ARMA model. In the remainder, we restrict our attention to autoregressive noise models like

$$\mathbf{x}_0 = \mathbf{x}_i b + \mathbf{B}\,\mathbf{x}_0 b_1 + \mathbf{B}_2\,\mathbf{x}_0 b_2 + \ldots + \mathbf{B}_P\,\mathbf{x}_0 b_P + \mathbf{e}. \tag{5.2}$$

In general, adding a lagged variable will decrease the differences among the elements of $\mathbf{y}_i$, while increasing the randomness in $\mathbf{e}$. It is instructive to write (5.2) as

$$\mathbf{x}_0 - (\mathbf{B}\,\mathbf{x}_0 b_1 + \mathbf{B}_2\,\mathbf{x}_0 b_2 + \ldots + \mathbf{B}_P\,\mathbf{x}_0 b_P) = \mathbf{x}_i b + \mathbf{e}, \tag{5.3}$$

which shows that we are actually examining the residuals of the autoregressive model for a systematic change in level. Equation (5.2) is the multiple autoregressive model discussed in Section 4.8, but with one extra exogenous variable.

Optimal scaling maximizes the multiple correlation between the dependent variable $\mathbf{x}_0$ and predictors over the regression weights $b, b_1 \ldots b_P$ and over optimal transforms of $\mathbf{x}_i$ and/or $\mathbf{x}_0$. If $b_1 = b_2 = \ldots = b_P = 0$ this reduces to maximizing the bivariate correlation between $\mathbf{x}_i$ and $\mathbf{x}_0$, so optimal scaling tends to maximize treatment differences then.

The estimation problem can be written in a canonical correlation analysis framework as minimizing

$$\sigma(\mathbf{z}; \mathbf{x}_i, \mathbf{x}_0; a_0, a, a_1 \ldots a_P) = \mathrm{ssq}(\mathbf{z}, \mathbf{x}_0 a_0) + \mathrm{ssq}(\mathbf{z}, \mathbf{x}_i a + \sum_{p=1}^{P} \mathbf{B}_p \mathbf{x}_0 a_p) \tag{5.4}$$

under normalization restrictions

$$\mathbf{1}'\mathbf{z} = \mathbf{1}'\mathbf{x}_0 = \mathbf{1}'\mathbf{x}_i = 0$$
$$\mathbf{z}'\mathbf{z} = \mathbf{1}'\mathbf{x}_0 = \mathbf{1}'\mathbf{x}_i = 1.$$

The regression weights can be found by projecting $\mathbf{x}_0$ on the predictor space after the optimal variable transforms have been found.

TABLE 5.1 Hyde Park Purse Snatchings data (Read across and down)

10	15	10	10	12	10	7	17	10	14	8	17	14	18	3
9	11	10	6	12	14	10	25	29	33	33	12	19	16	19
19	12	34	15	36	29	26	21	17	19	13	20	24	12	6
14	6	12	9	11	17	12	8	14	14	12	5	8	10	3
8	8	7	12	6	10	8	10	5	7					

As a simple example we use the Hyde Park Purse Snatchings series listed in MacCleary and Hay (1980). The series consists of 71 counts of purse snatchings in the Hyde Park, Chicago during the period of January 1969 to September 1973. The series is listed in Table 5.1, and it is plotted in Figure 5.2. At time point 42 Operation Whistlestop started, a community crime prevention program. Amongst others, the project involved distributing whistles to citizens which could be used to alarm the police. Figure 5.2 shows that after the intervention point the number of purse snatchings decreases, so the intervention seems to have a desirable effect.

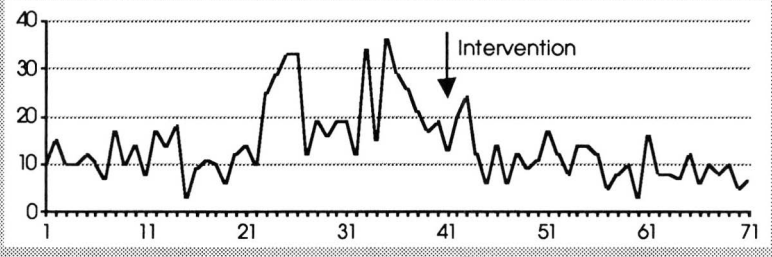


Figure 5.2 Purse Snatchings Series (N=71)

Using a t-test we found that the difference between the pre- and post-intervention means is statistically significant beyond $\alpha = 0.001$. The significance of this five-star result is debatable however, since it is hard to maintain that the observations are independent replications of each other. Look for example at the autocorrelation plot of the series in Figure 5.3: it is likely that the next observation will depend on the present and some past observations. This implies that for some time point with a

high number of offences we expect the surrounding values also to be high, thereby increasing the average of the series.

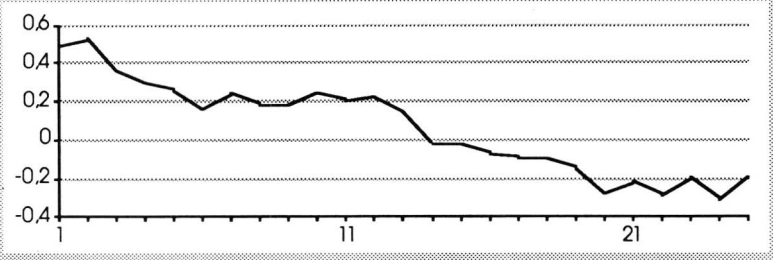


Figure 5.3 Autocorrelations for the Hyde Park series

MacCleary and Hay (1980) found that an AR(2) model fitted the series rather well. We analyzed the series three times with zero, one and two lags included in the regression equation. The results are summarized in Table 5.2.

TABLE 5.2 Results of three intervention analyses for the Purse Snatchings series.

Lag	Loss	r	b	b_1	b_2	BP	MBP	t
0	0.6048	0.3952	0.3952			69.33	85.29	3.57
1	0.4574	0.5429	0.2396	0.4059		27.86	35.49	1.23
2	0.3769	0.6237	0.1714	0.2562	0.3647	17.97	24.06	0.90

The zero-lag analysis corresponds to equation (5.1). The intervention weight b for this analysis is 0.3952. As noted before, this value is significant different from zero, however the residuals are very much autocorrelated as can be seen from the high values for the Box-Pierce (BP) and the Modified Box-Pierce statistic (MBP) (df = 25). When first and second lags of the series are included, these values drop dramatically. But simultaneously the intervention weight b and its associated t-statistic (df = 69) fade rapidly. So the more lags are included, the less residual autocorrelation, but also the less intervention effect we see. In this example, controlling for autocorrelation makes the intervention effect disappear. It is

however quite possible that controlling for other types of intervention will yield different results.

We have limited ourselves to the case of one intervening variable with only two levels. It is of course possible to extend the model to multivariable-multilevel MANOVA-like situations, but we will not explore that direction any further since such models will become very complex and difficult to interpret in the presence of autocorrelation. In the remainder of the chapter we focus on more exploratory oriented techniques to represent the interrelations among multiple time series.

5.2 The Box - Tiao transformation: predictable components

Box and Tiao (1977) propose a canonical analysis that extract predictable components from a multivariate time series. A predictable component is a linear combination of an observed multiple series and it is characterized by its ability to forecast itself with a high degree of precision. An obvious use of the technique is to identify those components that can serve as smoothed indicators of overall growth in for example the stock market. Alternatively, the technique can be used as a dimension reduction device to bring out the major time dependent characteristics of a multivariate data set.

Suppose we have a data matrix $\mathbf{X}$ of order $N \times M$ that contains scores on M variables or series recorded on N time points. The goal of the Box-Tiao transform is to find M linear combinations

$$\underline{Z} = \mathbf{X}\mathbf{A}_0, \tag{5.5}$$

that are contemporaneously independent and that are ordered according to their respective predictive powers. The predictability measure reflects how much a component can predict itself by a P^{th} order autoregressive model

$$\underline{z} = \sum_{p=1}^{P} \mathbf{B}_p \underline{z} f_p + \mathbf{e} \tag{5.6}$$

($\underline{z}$ denotes a column from $\underline{\mathbf{Z}}$) and it is defined as

$$\gamma = \frac{\hat{\underline{z}}'\hat{\underline{z}}}{\underline{z}'\underline{z}} = 1 - \frac{\mathbf{e}'\mathbf{e}}{\underline{z}'\underline{z}} \tag{5.7}$$

i.e. the proportion of variance of $\underline{z}$ explained by the predictor combination

$$\hat{\underline{z}} = \sum_{p=1}^{P} \mathbf{B}_p \underline{z} f'_p. \tag{5.8}$$

For the first predictable component, the goal is thus to find the weight vector $\mathbf{a}_0$ such that the linear combination $\underline{z} = \mathbf{X}\mathbf{a}_0$ has maximum predictability γ. Next, a second predictable component, orthogonal to the first, can be identified, and so on. The matrix version of (5.6) is

$$\underline{\mathbf{Z}} = \sum_{p=1}^{P} \mathbf{B}_p \underline{\mathbf{Z}} \mathbf{F}_p + \mathbf{E}, \tag{5.9}$$

with diagonal $\mathbf{F}_p$. Since $\underline{\mathbf{Z}} = \mathbf{X}\mathbf{A}_0$ we may express (5.9) in terms of the observed data if we substitute for $\underline{\mathbf{Z}}$ so that

$$\begin{aligned} \mathbf{X}\mathbf{A}_0 &= \sum_{p=1}^{P} \mathbf{B}_p \mathbf{X}\mathbf{A}_0\mathbf{F}_p + \mathbf{E} \\ &= \sum_{p=1}^{P} \mathbf{B}_p \mathbf{X}\mathbf{A}_p + \mathbf{E} \end{aligned} \tag{5.10}$$

is identical to (5.9). It is now easy to see that the Box-Tiao transform for identifying series of maximum predictability is equivalent to a canonical correlation analysis between a set of the observed series $\mathbf{X}$ and a set of lagged $\mathbf{X}$. Velu et al. (1986) found that γ is equal to the squared canonical correlation. Using more complicated and rigorous proofs, this relationship has been pointed out before by Parzen and Newton (1980) and Velu et al. (1986).

It is straightforward to translate the Box-Tiao transform into a least squares loss function. For a given multivariate categorical series $\mathbf{X}$ we must minimize

$$\sigma(\mathbf{Z}; \mathbf{X}; \mathbf{A}_0 \ldots \mathbf{A}_P) = \text{ssq}\,(\mathbf{Z}, \mathbf{X}\mathbf{A}_0) + \text{ssq}\,(\mathbf{Z}, \sum_{p=1}^{P} \mathbf{B}_p\mathbf{X}\mathbf{A}_p) \qquad (5.11)$$

over $\mathbf{Z}$, $\mathbf{X} = \mathbf{G}\mathbf{Y}$ and $\mathbf{A}_0 \ldots \mathbf{A}_P$ under constraints $\mathbf{1}'\mathbf{Z} = \mathbf{0}$, $\mathbf{1}'\mathbf{X} = \mathbf{0}$, $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$ and $\text{dg}\,\mathbf{X}'\mathbf{X} = \mathbf{I}$. The predictable components are equal to $\underline{\mathbf{Z}} = \mathbf{X}\mathbf{A}_0$.

It can be shown that minimizing (5.11) comes down to maximizing the sum of the M canonical correlations between $\mathbf{X}$ and $[\mathbf{B}\mathbf{X}, \mathbf{B}_2\mathbf{X}, \ldots, \mathbf{B}_P\mathbf{X}]$. If we define

$$\mathbf{Z}^* = 0.5\,(\mathbf{X}\mathbf{A}_0 + \sum_{p=1}^{P} \mathbf{B}_p\mathbf{X}\mathbf{A}_p)$$

then (5.11) partitions into

$$\sigma(\mathbf{Z}; \mathbf{X}; \mathbf{A}_0 \ldots \mathbf{A}_P) = 2\,\text{ssq}\,(\mathbf{Z}, \mathbf{Z}^*) + 0.5\,\text{ssq}\,(\mathbf{X}\mathbf{A}_0, \sum_{p=1}^{P} \mathbf{B}_p\mathbf{X}\mathbf{A}_p)\,, \qquad (5.12)$$

so by (2.26) we find that the second part of the loss function corresponds to obtaining the maximum of the sum of the M canonical correlations.

From a dimension reduction point of view, the information contained in the first few dimensions will be of primary importance. Since the lower dimensions capitalize on high autocorrelations, the first few predictable components will exhibit very smooth behavior. In contrast, the last few components will approach irregular white noise patterns. In this respect, the Box-Tiao transform can be regarded as a multivariate smoother with the most predictable component being an optimally smoothed linear combination of the observed series.

In the conventional Box-Tiao transform, where all variables are numerical, the solution will be nested. This means that dimensions are invariant under the number of computed predictable components. The optimal scaling version is not nested: with a different number of dimensions, different solutions may emerge, although in general the discrepancy will not be very large. This non-nesting property is common to nearly all ALS-techniques discussed in Chapter 3.

As an example, let us consider the dairy series in Table 5.3. The series records a number of psychological and medical factors and some daily activities of a woman in her mid-twenties for 131 consecutive days. The intent of the data collection was to examine what factors might influence the occurrences of eruptive fever.

Treating the data at an interval level, we computed the correlation box for 10 lags. Unfortunately most elements not located on the diagonal slice of the box turned out to be near zero, indicating that the time-related influence between series is limited. It seems that we are stuck with more or less cross-independent series. It could well be however that while the cross-dependency among the individual series is not very high, some linear combination of the series is highly autocorrelated. We may then inspect the composition of this most predictable component to see which series contribute most to the multivariate time-dependencies in the data. To this end, we computed the first two predictable components of the dairy data using a mix of optimal transformations. The used measurement level of each series is given in the legend of Table 5.3. We chose to include five lags in the predictor set.

The first predictable component is plotted in Figure 5.4. The component has a very strong periodic tendency corresponding to the 28-day menstruation cycle. The first canonical correlation is 0.86, so the predictability of the component is equal to 0.74. The second component is plotted in Figure 5.5. The canonical correlation is 0.76 so the predictability of this series equals 0.58.

TABLE 5.3 Dairy Data (N = 131)

1	1211131	45	2111151	89	2211242	Description
2	3211111	46	2111151	90	3211343	
3	3211121	47	2111131	91	2211133	Series 1: Emotional State *
4	3211131	48	2111151	92	2231112	1: down
5	3211121	49	3111132	93	3211222	2: normal
6	4221111	50	2112112	94	3221252	3: good
7	4211151	51	3222121	95	2211113	4: active-hysteric
8	3121151	52	3212141	96	3211343	
9	4211111	53	3212252	97	2211342	Series 2: Physical State ✎
10	4111121	54	2212123	98	2211153	1: ill
11	1221121	55	2212151	99	2211132	2: healthy
12	1211141	56	2211131	100	2211343	
13	2211112	57	2111212	101	2211313	Series 3: Sexual Activity ✎
14	2211122	58	2211131	102	3211323	Nothing (1) - Much (3).
15	2221142	59	2211111	103	3211323	
16	2211122	60	2211123	104	2211323	Series 4: Indisposed *
17	3211122	61	2211121	105	3211323	1: no
18	3211122	62	1211122	106	2211353	2: yes
19	3211232	63	1221112	107	3222353	
20	2221112	64	2221143	108	3222131	Series 5: Smoking ✎
21	2211131	65	2211111	109	3212131	1: none, missing
22	2212111	66	3221151	110	3212152	2: some (1- 10 cigarettes)
23	3112122	67	1211111	111	4212313	3: much (>10 cigarettes)
24	2112122	68	2211151	112	1212141	
25	3112122	69	2211343	113	1211112	Series 6: Food *
26	1112142	70	1221123	114	2221141	1: Italian
27	1111212	71	1211312	115	1211152	2: Dutch
28	2131113	72	2211122	116	2211253	3: Bread
29	2111113	73	2221122	117	2111242	4: Snacks
30	3221252	74	2211143	118	2211343	5: Other foreign
31	4211221	75	4221153	119	3231313	
32	2211122	76	2211312	120	2211111	Series 7: Alcohol ✎
33	2211123	77	2211132	121	2111212	1: none, missing
34	2211121	78	3222113	122	2111232	2: some 1-3 beer/wine
35	2211131	79	2212222	123	1211133	3: much > 3
36	2221133	80	3212222	124	1121152	
37	2221222	81	3212333	125	2111333	
38	2211223	82	3212223	126	2111121	Measurement levels:
39	1211222	83	3222242	127	2211113	* nominal
40	2211121	84	3211323	128	2221222	✎ ordinal
41	2211111	85	3211233	129	2211353	↔ numerical
42	2221112	86	4231213	130	3211252	
43	1211111	87	2211212	131	1121232	
44	2131253	88	2221312			

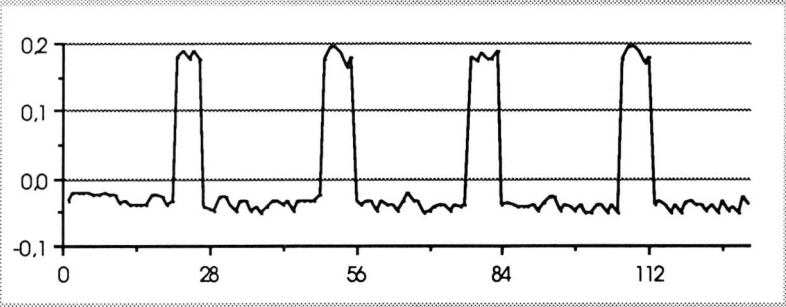


Figure 5.4 First predictable component of the Dairy data

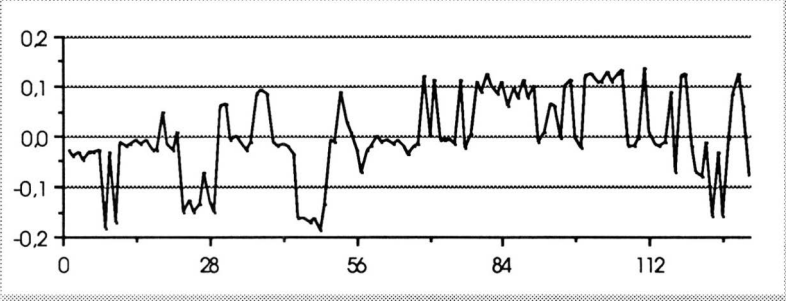


Figure 5.5 Second predictable component of the Dairy data

In order to interpret the components it is useful to look at the correlations between the predictable components and the series from which they were constructed. There is a total of 6 (lags) × 7 (variables) × 2 (dimensions) = 84 correlations to look for. Figure 5.6 contains 7 two-dimensional plots of the correlations between the six first lags of each variable with the two predictable components.

The position of each point indicates the correlation with the first (horizontal) and second (vertical) component. The points are labeled by their lag numbers.

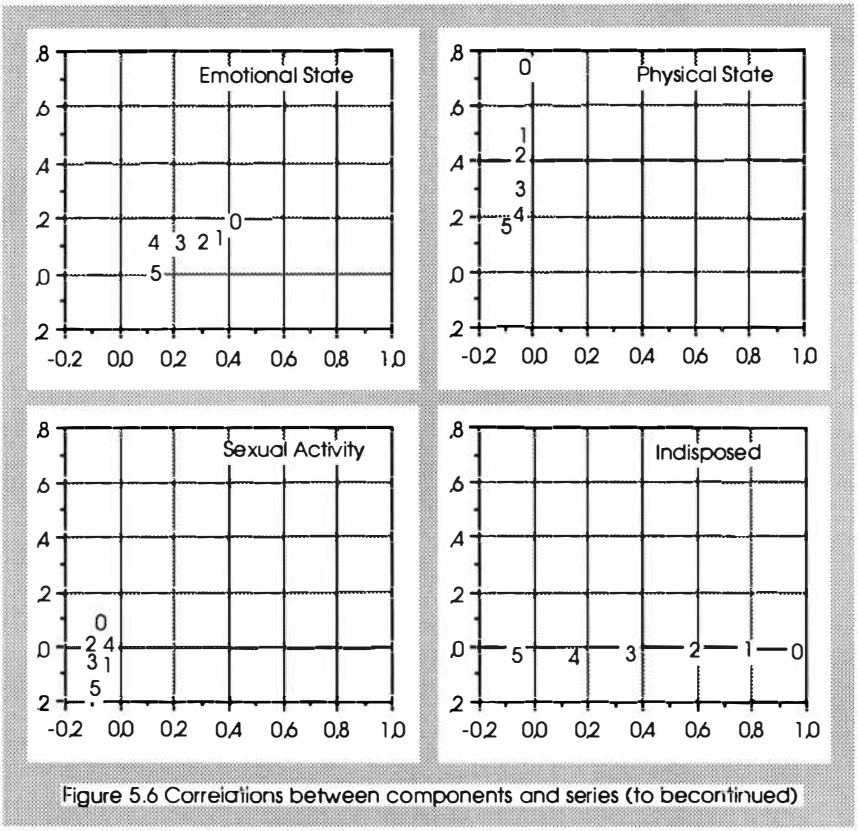
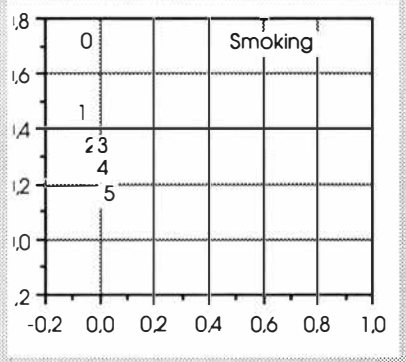
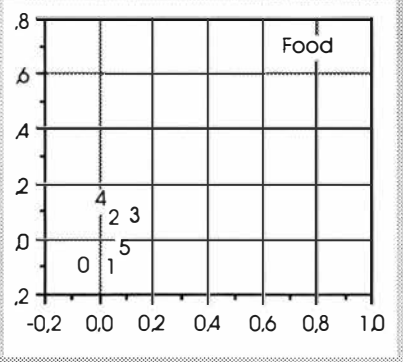


Figure 5.6 Correlations between components and series (to becontinued)

The first dimension is dominated by the menstruation cycle: the correlation between the period series and the component is 0.99. A second variable that "loads" on this component is emotional state but its contribution is only rather slight.

The second component depends on a combination of physical state, smoking and alcohol consumption. The most important contributor is the physical state series. The dips in the second component in Figure 5.5 exactly match the periods of illness. The relationship between smoking and drinking behavior is fairly obvious. Both are likely to be moderated by social events such as parties, visits and holidays.





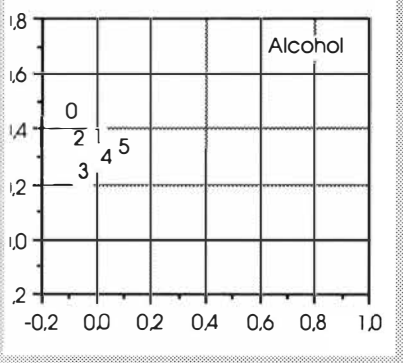


Figure 5.6 Correlations between components and series (continued)

The peaks in Figure 5.5 correspond to periods with much drinking and smoking. See for example the periods between days 80-91 and 100-107. These two periods are both located within a longer period of physical well-being. In general, it seems that the dips are caused by illness and that the peaks are caused by many cigarettes (10 or more) and much wine. As we will see below, these two appear to be relatively independent processes. Because of this, the relationship between health and smoking/ drinking is not necessarily positive, in case you planned to rush out to the nearest pub to consume lots of medicinal spirituals.

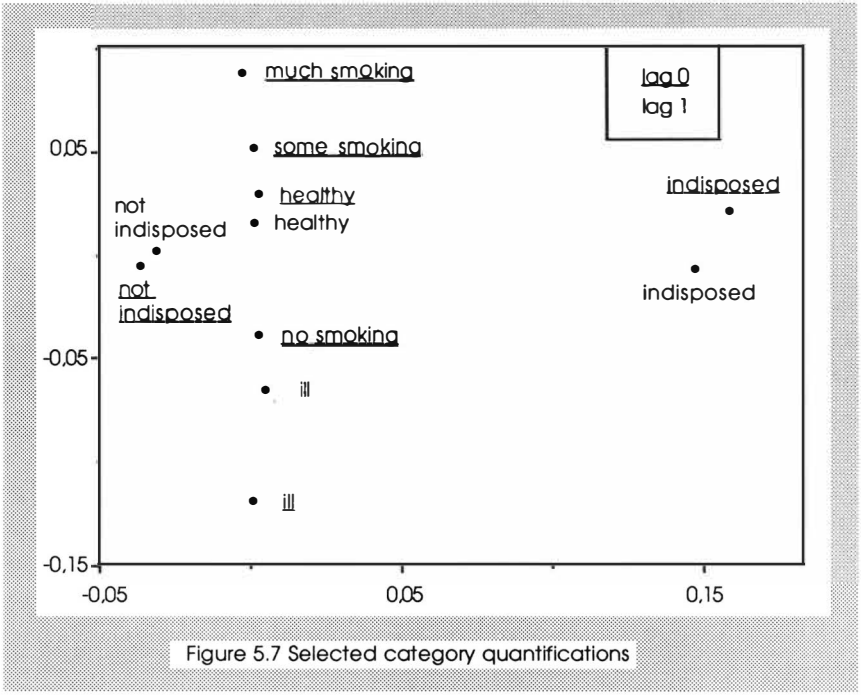


Figure 5.7 Selected category quantifications

The correlations for the higher lags diminish for nearly all seven series. This not only indicates that first-order time relationships dominate higher order influences in the series themselves, but also leads to the conclusion that there are few, if any, clear dynamic cross-relationships among the series. For example, if drinking invariably leads to gross headaches on the next day, then we expect the correlations between lag-0 of health and lag-1 of drinking to be rather large. The dairy series do not reveal such patterns.

The optimally scaled categories $\mathbf{Y}$ have been standardized such that $dg(\mathbf{Y'G'GY}) = \mathbf{I}_M$. To examine the relative influence each category has on the solution, we multiply the quantification matrix by the obtained weights $\mathbf{A}$, i.e. $\mathbf{V}_0 = \mathbf{YA}_0$, $\mathbf{V}_1 = \mathbf{YA}_1$ and so on. In Figure 5.7 we plotted the most important categories, where the rows of $\mathbf{V}$ are taken as coordinates.

High scores on the first component are caused by the menstruation category. The quantifications of the other series are hardly different from zero, so the menstruation cycle is a relatively independent process without too many side effects. Note that the distance between both "not indisposed" categories is smaller than the distance between both "indisposed" categories. One way to interpret this is that it is easier to predict no period from a current no period observation than it is to predict a period from the current period category. The same holds for the physical state series (ill-healthy) on the second component. As said before, high scores on component 2 correspond to much smoking and drinking, low scores by the illness category. Obviously, one does not smoke over 10 cigarettes if one is sick.

Let us make a final observation on the predictable components technique. The first few components tend to single out the slowest varying series. In the example, these are the menstruation series followed by the physical state and smoking series. We expect the higher components to capture faster moving series like sexual activity, type of food and emotional state, but these will be less predictable of course.

5.3 Dynamic components analysis

The goal of dynamic components analysis and dynamic factor analysis is to reduce M observed series to R latent series (R < M) without discarding too much time relevant information. See Section 2.4 for references.

As explained in Section 2.4, there are two common definitions of the dynamic factor model: the "lagged factors" and the "state space" form. The lagged factors model was written as

$$\mathbf{X} = \sum_{q=0}^{Q} \mathbf{B}_q \mathbf{Z} \mathbf{F}_q + \mathbf{U}, \tag{5.13}$$

and the state space model was defined by

$$\mathbf{Z} = \mathbf{B}\mathbf{Z}\mathbf{F}_1 + \mathbf{W} \tag{5.14}$$
$$\mathbf{X} = \mathbf{Z}\mathbf{F}_0 + \mathbf{V}. \tag{5.15}$$

The dynamic components analysis proposed below borrows some concepts of both formulations. Like the lagged factors model, it utilizes the concept of lagged factors. Like the state space model, it has a system equation akin of (5.14) for describing the dynamic behavior of the series. Since we are more interested in dimension reduction than modelling, we do not provide a model for the error structure. This makes the method a principal components analysis instead of a factor analysis.

The approximation structure of the dynamic components model is

$$\mathbf{Z} = \mathbf{X}\mathbf{A} \tag{5.16}$$
$$\mathbf{Z} = \sum_{q=1}^{Q} \mathbf{B}_q\mathbf{Z}\mathbf{F}_q \tag{5.17}$$

with $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$. Since both equations cannot be satisfied simultaneously in general, some loss will occur when trying to fit the structure to the data $\mathbf{X}$. This loss is most conveniently measured by the function

$$\sigma(\mathbf{Z}; \mathbf{X}; \mathbf{A}, \mathbf{F}_1 \ldots \mathbf{F}_Q) = \text{ssq}\,(\mathbf{Z}, \mathbf{X}\mathbf{A}) + \text{ssq}\,(\mathbf{Z}, \sum_{q=1}^{Q} \mathbf{B}_q\mathbf{Z}\mathbf{F}_q) \tag{5.18}$$

which can be minimized over $\mathbf{Z}$, $\mathbf{X} = \mathbf{G}\mathbf{Y}$, $\mathbf{A}$ and $\mathbf{F}_1 \ldots \mathbf{F}_Q$ under normalizations $\mathbf{1}'\mathbf{Z} = \mathbf{0}$, $\mathbf{1}'\mathbf{X} = \mathbf{0}$, $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$ and $\text{dg}(\mathbf{X}'\mathbf{X}) = \mathbf{I}$. Minimizing (5.18) will result in a dynamic component series $\mathbf{Z}$ that combines current observation from $\mathbf{X}\mathbf{A}$ with the past observations of itself. Thus, the system equation (5.17) serves as the memory of the system.

The dynamic components method is a multivariate generalization of the exponential smoothing filter discussed in Chapter 4. We may regard the component series $\mathbf{Z}$ as a filtering output that compromises between the

current multivariate observation $\mathbf{x}_t$ and the past states of the system $\mathbf{z}_{t-1} \ldots \mathbf{z}_{t-Q}$.

The relationship with the Box-Tiao transform is easy to see: if we restrict $\mathbf{Z}$ to be an exact linear combination of $\mathbf{X}$ instead of an approximation, then ssq($\mathbf{Z}$, $\mathbf{XA}$) = 0 and the dynamic components model reduces to the Box-Tiao transform. The two methods only differ in the weighting of the loss function part ssq($\mathbf{Z}$, $\mathbf{XA}$).

The Hutt series render a popular data set for demonstrating the use of dynamic factor models. The series consists of five physiological variables. They are:

1. amplitude electro-encephalogram (EEG)
2. speed heartbeam
3. speed breathing
4. variability heartbeam
5. variability breathing.

Hutt, Lenard and Prechtl (1969) measured these variables on a 8-days old baby on 124 time points. The interval between two subsequent measurements was 3 minutes. The data can be found in Molenaar (1981) and Immink (1986). The series are plotted in Figure 5.8.

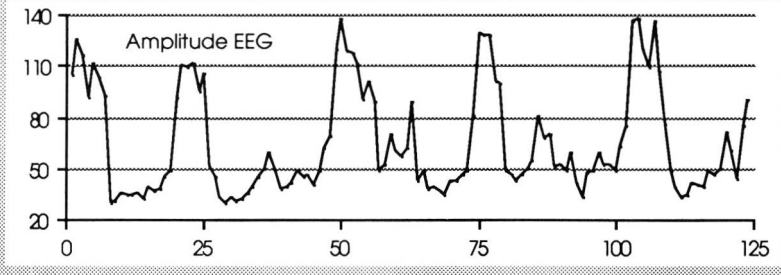


Figure 5.8 Hutt series (to be continued)

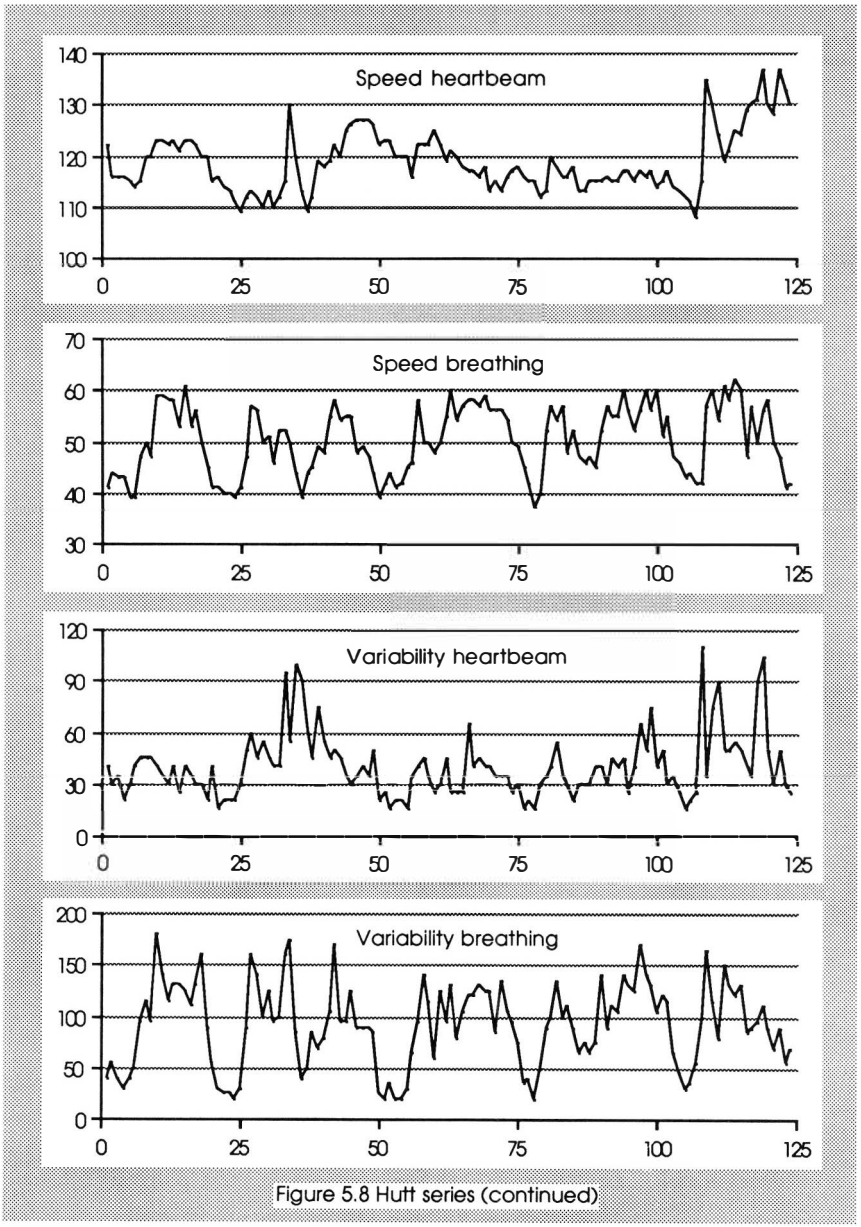


Figure 5.8 Hutt series (continued)

We analyzed the Hutt series four times with the dynamic components technique. We varied the number of latent lags Q (1 or 2) and the number of components R (also 1 or 2) so we have a total of four solutions. Let us first take a look at a partitioning of the loss values given in Table 5.4.

TABLE 5.4 Loss value partitioning of four dynamic components analyses

	ONE COMPONENT, ONE LAG			ONE COMPONENT, TWO LAGS		
Dimension	ssq(**Z**,**XA**)	ssq(**Z**, **BZF**$_1$)	total	ssq(**Z**,**XA**)	ssq(**Z**, Σ**BZF**)	total
1	0.0411	0.1461	0.1871	0.0427	0.0377	0.0804

	TWO COMPONENTS, ONE LAG			TWO COMPONENTS, TWO LAGS		
Dimension	ssq(**Z**,**XA**)	ssq(**Z**, **BZF**$_1$)	total	ssq(**Z**,**XA**)	ssq(**Z**, Σ**BZF**)	total
1	0.0580	0.0719	0.1299	0.0402	0.0294	0.0697
2	0.0570	0.1317	0.1887	0.0723	0.0560	0.1283
	0.1150	0.2036	0.3187	0.1125	0.0854	0.1980

For the one-component solutions the loss ssq(**Z**,**BZF**) decreases from 0.1461 to 0.0377 as a result of the inclusion of the second lag of **Z**, while ssq(**Z**, **XA**) remains more or less constant at 0.04. The same phenomenon occurs if we compare the two-component solution: ssq(**Z**,**BZF**) now drops from 0.2036 to 0.0854. This suggest that the component(s) can better be described by an AR(2) rather than an AR(1) model. Including a third lag does not lead to a substantial improvement of the fit so using the elbow criterion we decide for two lags.

If we compare the one-lag solutions, we may notice a rather odd thing: the first component of the two-component solution *fits better* than the one of the one-component solution (loss values 0.1299 versus 0.1871). One might say that this is theoretically impossible. After all, why doesn't the method find the better component for the one-dimensional analysis? The answer is simple. We should realize that an extra predictor sneaks in. Although the components are independent so that $\mathbf{z}_1'\mathbf{z}_2 = 0$ it is not true

that $\mathbf{z}_1'\mathbf{B}\mathbf{z}_2$ is also zero. Thus the first component may not only depend on its own past, but also on the past of the second dimension. As a result, the prediction error becomes less and the fit increases. If we do not want this kind of strange behavior we may restrict $\mathbf{F}_q$ to be diagonal. The other components are effectively ruled out as predictors then.

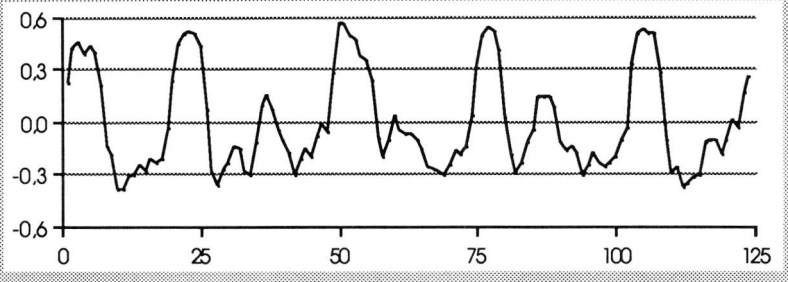


Figure 5.9 Immink's dynamic factor

Immink (1986) found that the five variables can be described by a general arousal factor which follows an ARMA(2, 1) model. His dynamic factor is plotted in Figure 5.9.

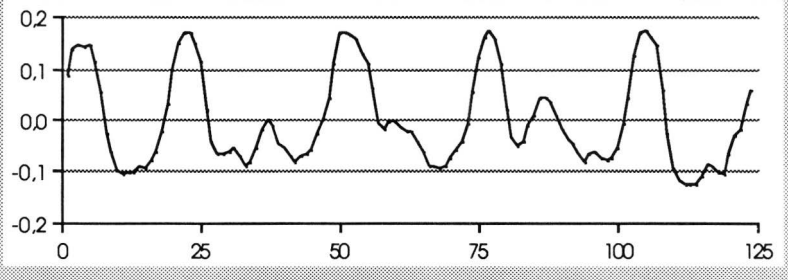


Figure 5.10 Our dynamic component

The dynamic component of the one-component one-lag analysis is plotted in Figure 5.10. This component was identified by all four analyses. The correlation with the Immink factor is nothing less than 0.97. Despite the

substantial technical differences between Immink's method and our dynamic components analysis, the series are virtually identical.

A close visual inspection of Figures 5.9 and 5.10 reveals that the dynamic component is smoother. This also becomes evident from the autocorrelations of both series. Table 5.5 compares the first 12 autocorrelations.

TABLE 5.5 Twelve autocorrelations from the Factor and the Component solution

	1	2	3	4	5	6	7	8	9	10	11	12
F	.88	.63	.35	.08	-.16	-.33	-.42	-.44	-.43	-.38	-.33	-.26
C	.92	.73	.49	.22	-.03	-.23	-.36	-.44	-.47	-.47	-.44	-.38

The component autocorrelations are on the average more extreme. They start higher, indicating that the component series has less abrupt changes, and they are lower around lag 9 and this is a sign of the stronger periodicity of the component series.

The weight coefficients in **A** can be inspected to infer in what way the components relate to the observed series **X**. For the two-components two-lags analysis the coefficients are summarized in Table 5.6.

TABLE 5.6 Coefficient weights in **A**

	Component 1	Component 2
1. amplitude EEG	.55	.01
2. speed heartbeam	.22	-.91
3. speed breathing	-.31	.05
4. variability heartbeam	-.07	-.29
5. variability breathing	-.21	.12

The first component is largely determined by the EEG series and in a somewhat lower extend and in the opposite direction by both breathing series. The series can be interpreted as a general arousal factor (cf. Immink, 1986, 153). If we compare Figure 5.10 with the EEG series in Figure 5.8 we see that the peaks occur at almost the same points.

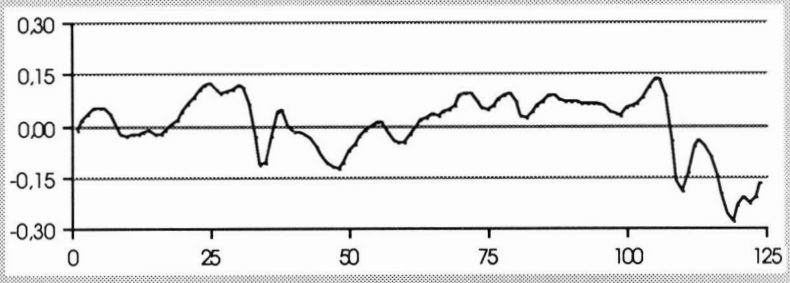


Figure 5.11 Second dynamic component (two-components two-lags)

The second component turns out to be governed by the heartbeam series. It is plotted in Figure 5.11. It is very much like a smoothed version of speed heartbeam depicted upside down.

The time dependency of the latent series is described by the system equation (5.17). The system matrices $\mathbf{F}_1$ and $\mathbf{F}_2$ are given in Table 5.7. The first component series $\mathbf{z}_{1,t}$ is thus modelled by

$$\mathbf{z}_{1,t} = -1.76\,\mathbf{z}_{1,t-1} + 0.90\,\mathbf{z}_{1,t-2} + 0.11\,\mathbf{z}_{2,t-1} - 0.07\,\mathbf{z}_{2,t-2}.$$

TABLE 5.7 System matrices $\mathbf{F}_1$ and $\mathbf{F}_2$

$\mathbf{F}_1$		$\mathbf{F}_2$	
-1.76	-0.11	0.90	0.12
0.11	-1.60	-0.07	0.69

We finally note that the value of the Box-Pierce statistic varies between 50 and 150 with df = 23 and df = 24. This means that, although most time related information has been captured by the analyses, the residuals are still autocorrelated. We may further refine the analysis if we include more lags of $\mathbf{Z}$. A good candidate for the first component is the 25^{th} lag. Including this lag will filter out most of the periodicity of the first component. We then end up with a seasonal components model.

5.4 Multiset dynamic components analysis

Until now we have considered time series analysis as a two-set canonical correlation problem. For example, in multiple autoregression the first set contains the dependent variables and the second set holds the predictors. In the same manner, the first set in dynamic components analysis consists of lagged objects scores **Z** and the second set is equal to the series **X**.

In some cases it is useful to partition the observed series into more than two sets. In time series analysis, doing this is useful when each set can be considered as a replication of the same process. For example, if five raters observe a mother interacting with her child in a laboratory experiment, the scores can be assigned to five sets. Each set corresponds to one rater. It is then possible to investigate not only the time-dependency of one or more aggregated series (e.g. the mean scores over the five raters), but we may also examine inter-rater agreement. Conversely, suppose we have data on a social group stratified according to age, geographic location and other relevant background variables. Then it would be interesting to determine a common latent process that accounts for the behavior of each individual member. Each member is regarded as a replication then.

It is not difficult to generalize the multiset technique from the two-set dynamic components analysis. We simply add more sets. Let the columns of the N × M data matrix **X** be partitioned in K sets, i.e.

$$\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_K],$$

where M_k (k=1,...,K) denotes the number of variables in the k^{th} set. Analogous to (5.16) and (5.17) the approximation structure of the multiset model becomes

$$\mathbf{Z} = \mathbf{X}_1\mathbf{C}_1 = \mathbf{X}_2\mathbf{C}_2 = \ldots = \mathbf{Z} = \mathbf{X}_K\mathbf{C}_K \qquad (5.19)$$

and

$$\mathbf{Z} = \sum_{q=1}^{Q} \mathbf{B}_q \mathbf{Z} \mathbf{F}_q. \tag{5.20}$$

We may also write this as

$$\mathbf{Z} = \mathbf{X}_1\mathbf{C}_1 = \mathbf{X}_2\mathbf{C}_2 = \ldots = \mathbf{X}_K\mathbf{C}_K = \sum_{q=1}^{Q} \mathbf{B}_q \mathbf{Z} \mathbf{F}_q. \tag{5.21}$$

We are thus looking for R linear combinations of the series $\mathbf{X}_1 \ldots \mathbf{X}_K$ that result in variates that are close to each other and that follow a low-order autoregressive process. The object scores $\mathbf{Z}$ denote the common latent series. The loss is measured by

$$\sigma(\mathbf{Z}; \mathbf{X}_k; \mathbf{C}_k; \mathbf{F}_q) = \sum_{k=1}^{K} \text{ssq}\,(\mathbf{Z}, \mathbf{X}_k\mathbf{C}_k) + \text{ssq}\,(\mathbf{Z}, \sum_{q=1}^{Q} \mathbf{B}_q \mathbf{Z} \mathbf{F}_q) \tag{5.22}$$

which must be minimized over $\mathbf{Z}$, $\mathbf{X}_k = \mathbf{G}\mathbf{Y}_k$, $\mathbf{C}_k$ and $\mathbf{F}_q$ under normalization constraints $\mathbf{1}'\mathbf{Z} = \mathbf{0}$, $\mathbf{1}'\mathbf{X}_k = \mathbf{0}$, $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$ and $\text{dg}(\mathbf{X}'\mathbf{X}) = \mathbf{I}$ for all k. We prefer to reformulate (5.22) a little bit so that the k subscript drops out for $\mathbf{X}$ and $\mathbf{Y}$. This is done by defining $\mathbf{A}_k = [\mathbf{0}, \ldots, \mathbf{C}_k', \ldots, \mathbf{0}]'$ for all k, i.e. setting the rows of $\mathbf{A}_k$ corresponding to the series that do not belong to set k to zero. The rewritten loss is then equal to

$$\sigma(\mathbf{Z}; \mathbf{X}; \mathbf{A}_k; \mathbf{F}_q) = \sum_{k=1}^{K} \text{ssq}\,(\mathbf{Z}, \mathbf{X}\mathbf{A}_k) + \text{ssq}\,(\mathbf{Z}, \sum_{q=1}^{Q} \mathbf{B}_q \mathbf{Z} \mathbf{F}_q) \tag{5.23}$$

with $\mathbf{X} = \mathbf{G}\mathbf{Y}$. The advantage of using (5.23) instead of (5.22) is that it is easier to relate (5.23) to loss function (6.6), but both will yield the same result.

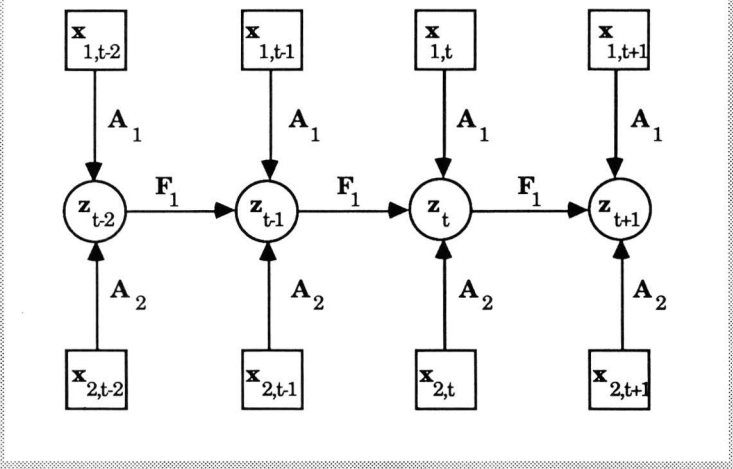


Figure 5.12 Scheme of multiset dynamic components analysis

A schematic representation of the technique for K = 2 and Q = 1 is given in Figure 5.12. The time dependencies in the data are modelled by the dynamic component **Z**. Suppose that the correlation box partitions into K × K blocks. The technique then aims to reduce this box to an R × R correlation box of **Z** in which only the diagonal slice is nonzero.

As an example, we use memory scores from 18 seniors collected on 7 points of time. The data were collected as a part of the Dutch Longitudinal Study among the Eldery (cf. Deeg et al., 1985, 1989) and they were kindly provided by Dorly Deeg. In a 20-year follow-up study, the short-term memory ability of 18 individuals was measured. Scores range from 0 (nothing remembered) to 25 (perfect score). The interval between two occasions is approximately 3 years. Deeg et al. (1989) use this type of data to determine whether or not cognitive memory functioning has anything to say about life-expectancy. Here we use the data to explore major memory development patterns. The 18 series are plotted in Figure 5.13.

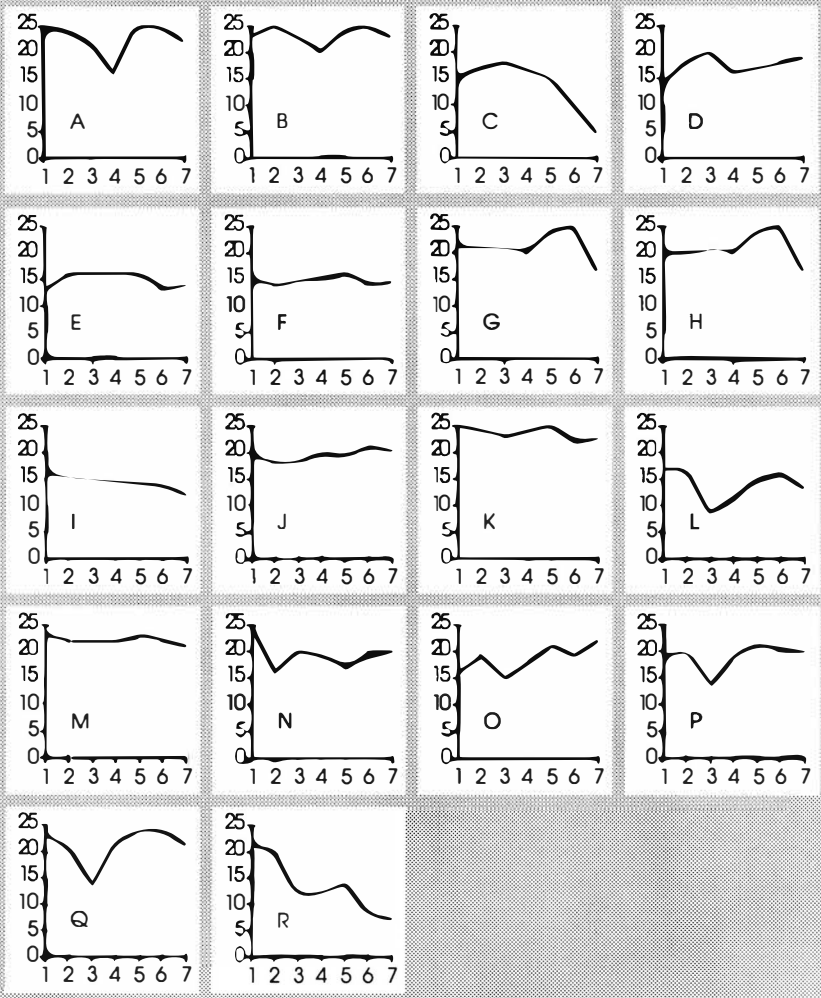


Figure 5.13 Memory test scores of 18 individuals on 7 time points

We analyzed the data with the multiset technique with K = 18, Q = 1 and R = 2. The total loss for this analysis is equal to 27.8. This loss divides into 13.6 for dimension 1 and 14.2 for dimension 2. For this analysis we restricted $\mathbf{F}_k$ to be diagonal so that it is assumed that a latent series

depends on its own past only. The obtained system matrix is

$$\mathbf{F}_k = \begin{bmatrix} .92 & .00 \\ .00 & .16 \end{bmatrix}$$

which shows that the first component is very smooth with f = 0.92.

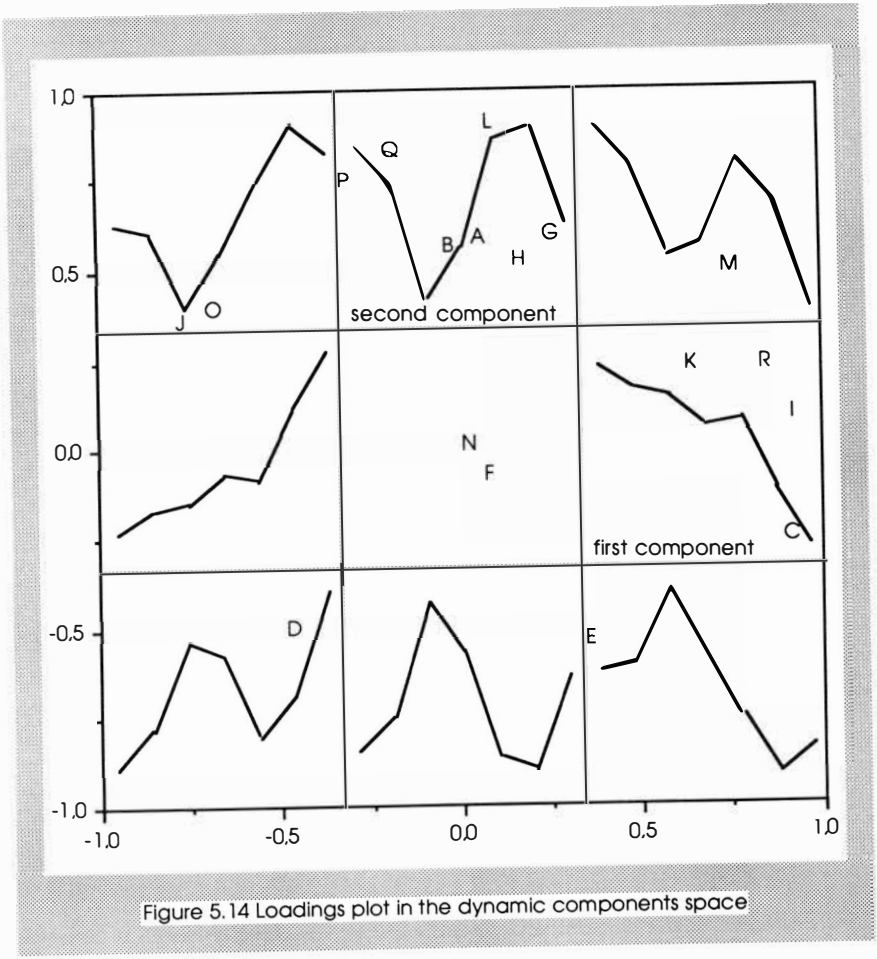


Figure 5.14 Loadings plot in the dynamic components space

It is interesting to look at the component loadings in $\mathbf{A_k}$. Because each set contains just one variable, the the loadings are equal to the correlations between the variable and the two latent components. Figure 5.14 shows the loadings plotted in the space of the two dynamic components.

The first component is plotted at the three o'clock position. This component shows a clear downward trend. The plot of the second component is at twelve o'clock. This series has a severe dip at the third observation. All other series are combinations of these two components. For example, the series located on the upper-right side is the average of the two components. Series situated at opposite ends are reciprocals.

Each letter refers to an individual. On the right side object I is positioned at coordinates (0.92, 0.12). These coordinates are the loadings on the first respectively second component. The most outspoken object on the top is individual L, with position (0.14, 0.90). If we compare the raw data with the components for these two objects we see the large degree of similarity between them. In the same way the raw scores of the other objects can be matched against the component profiles. Two major groups show up. The first is located near the first component (objects C, I, K, M, and R). This group is characterized by a decrease in memory functioning, probably as a result of ageing. People were about 90 years old at the end of the follow-up. The second group consists of persons A, B, G, H, L, P and Q. All members have a dip in the middle of the series. We do not have an explanation for this. A possibly relevant factor is that both A and B and P and Q live together, but we have no further data available that might help here. The other individuals are scattered over the plot. Persons F and N are located towards the centre. This indicates that none of the eight profiles fits their series very well.

We evaluate the quality of the solution by assessing of the amount of explained variance. The total variance in a set can be partitioned into an explained and an unexplained part. We have solved equations of the type

$$\mathbf{z} = \mathbf{x}a + \mathbf{e} \tag{5.24}$$

for minimum variance of $\mathbf{e}$. Since $\mathbf{z}'\mathbf{z} = \mathbf{x}'\mathbf{x} = 1$ and $\mathbf{x}'\mathbf{e} = 0$ we have that

$$1 = a^2 + \mathbf{e}'\mathbf{e}. \tag{5.25}$$

The squared component loading a^2 is equal to the proportion of explained variance. After computing these proportions for all set and both dimensions we may display them in a variance bar chart like Figure 5.15.

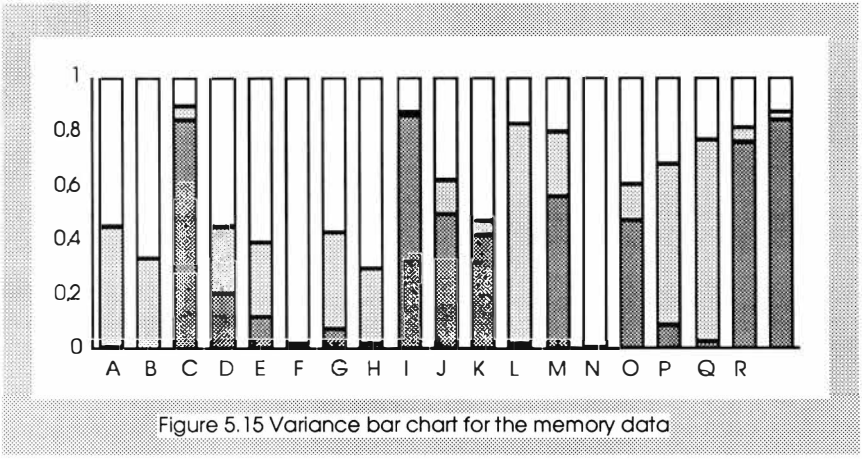


Figure 5.15 Variance bar chart for the memory data

Pattern ● corresponds to the variance explained by dimension 1. As expected, the objects of the first group (C, I, K, M and R) take on high values here. The last column refers to the extra set of the lagged component series. The major time dependency is in the first dimension. Pattern ▒ represents dimension 2. The transparent pattern ◯ renders the proportion of unexplained variance of each set. Almost no variance is explained of the badly fitting objects F and N. The variance bar chart provides a graphic illustration of the constituents of the loss value. The aim of the multiset dynamic components analysis is to minimize the sum of transparent bar space.

CHAPTER 6

Integration and minimization

Chapters 4 and 5 present various techniques that unify optimal scaling and time series analysis. The material covered in this chapter goes one step further as it concentrates the individual techniques into one general approach. This chapter borrows the concept of lagged variables from time series analysis and it borrows the concept of optimal scaling from nonlinear multivariate analysis. From a time series point of view, the method discussed here is a least squares data fitting procedure of a time series model that contains additional scaling parameters. From an optimal scaling perspective, the approach involves the analysis of supplementary, lagged variables that are subject to equality constraints on the quantifications.

We start with the introduction of an omnibus loss function. We distinguish among three classes of minimization problems: the canonical class, the ARMA class and the state space class. The methods discussed in Chapter 4 and 5 can all be seen as special cases of the canonical class, and therefore this class receives most attention. Minimization of the canonical class function is based on a double majorization procedure. We derive two algorithms. The first algorithm is the most universal. It integrates optimal scaling and the analysis of data with general dependency patterns among the observations. The second one specializes to time series analysis and this one was used for all computations. We conclude with two short accounts of the ARMA class and the state space class.

6.1 An omnibus loss function

This section introduces a general loss function for analyzing multivariate time series under optimal scaling of variables. This loss function forms the backbone of our formulation of all time series techniques discussed in this book.

Suppose we have observed M categorical variables on N consecutive time points. Let $\mathbf{X} = \mathbf{GY}$ of order N × M be a quantified data matrix as defined by (3.7). Let $\mathbf{A}_{k0}$ be an M × R real valued matrix. Then $\mathbf{XA}_{k0}$ is a matrix of order N × R containing R linear combinations of the columns of $\mathbf{X}$. If we restrict certain rows of $\mathbf{A}_{k0}$ to be zero, then the corresponding columns in $\mathbf{X}$ do not enter the linear combinations in any way and we are effectively dealing with a subset of $\mathbf{X}$. Let k = 1...K be the number of sets. Applying the zero row constraint for K weight matrices $\mathbf{A}_{10}...\mathbf{A}_{k0}...\mathbf{A}_{K0}$ allows us to construct linear combinations of any K column subsets of $\mathbf{X}$. The choice K = 1 or K = 2 is often convenient, but there are also problems in which it is useful to construct more than two sets of variables.

If $\mathbf{B}_p$ is the N × N backshift matrix of order p, then $\mathbf{B}_p\mathbf{X}$ is the p^{th} order matrix of lagged variables. Let $\mathbf{A}_{kp}$ be an M × R matrix of weights with proper zero row constraints. The product $\mathbf{B}_p\mathbf{XA}_{kp}$ then defines R linear combinations of any subset of all p^{th} order lagged variables. By adding these linear combinations over $p = 0...P_k$, i.e.

$$\underline{\mathbf{X}}_k = \sum_{p=0}^{P_k} \mathbf{B}_p\mathbf{XA}_{kp}, \tag{6.1}$$

the resulting $\underline{\mathbf{X}}_k$ can be any linear combination of any lags of any series. In exactly the same way, we define

$$\underline{\mathbf{Z}}_k = \sum_{q=0}^{Q_k} \mathbf{B}_q\mathbf{ZF}_{kq} \tag{6.2}$$

as a result of any linear combination of any lags of any latent series.

The only difference between $\mathbf{X}$ and $\mathbf{Z}$ is that $\mathbf{X}$ is observed, though possibly incomplete, while the values for $\mathbf{Z}$ are all unknown. Without any loss of generalization we may take $\mathbf{Z}$ as an orthogonal basis of a space with dimensionality R.

Geometrically, $\underline{\mathbf{Z}}_k$ defines R vectors that span a subspace in $\Re^N$ of at most R dimensions. Similarly, each of the lag products $\mathbf{B}_q\mathbf{Z}$ (q = 0...Q) also defines R vectors that span a subspace in $\Re^N$ of at most R dimensions. In general, these subspaces will be different from each other. If we construct $\underline{\mathbf{BZ}} = [\mathbf{B}_0\mathbf{Z},\ldots,\mathbf{B}_Q\mathbf{Z}]$ then $\underline{\mathbf{BZ}}$ defines R(Q+1) vectors that span a hyperspace in $\Re^N$ that unite all lag product subspaces $\mathbf{B}_q\mathbf{Z}$. Since $\underline{\mathbf{Z}}_k$ is a linear combination of $\underline{\mathbf{BZ}}$ it is also a subspace of this hyperspace $\underline{\mathbf{BZ}}$. In the same way we find that $\underline{\mathbf{X}}_k$ corresponds to an R-dimensional subspace of the united hyperspace $\underline{\mathbf{BX}} = [\mathbf{B}_0\mathbf{X},\ldots,\mathbf{B}_P\mathbf{X}]$.

The idea is that we must find the closest match in R dimensions between the observable hyperspace $\underline{\mathbf{X}}_k$ and the unobservable hyperspace $\underline{\mathbf{Z}}_k$ over all k = 1...K. This immediately translates into the *omnibus loss function*

$$\sigma(\underline{\mathbf{Z}}_k; \underline{\mathbf{X}}_k) = \sum_{k=1}^{K} \text{ssq}\,(\underline{\mathbf{Z}}_k, \underline{\mathbf{X}}_k). \tag{6.3}$$

By imposing various types of restrictions on the relations between the total of K × 2 hyperspaces we will able to derive many multivariate analysis techniques as special cases.

If we substitute for $\underline{\mathbf{X}}_k$ and $\underline{\mathbf{Z}}_k$ we obtain (6.3) as

$$\sigma(\mathbf{Z}; \mathbf{X}; \mathbf{A}_{kp}; \mathbf{F}_{kq}) = \sum_{k=1}^{K} \text{ssq}\,\Big(\sum_{q=0}^{Q_k} \mathbf{B}_q\mathbf{Z}\mathbf{F}_{kq}, \sum_{p=0}^{P_k} \mathbf{B}_p\mathbf{X}\mathbf{A}_{kp}\Big) \tag{6.4}$$

Written in its present form, the loss function is not directly suitable for data analysis because it contains far too many unknowns. It is therefore useful to concentrate on classes of minimization problems that are more manageable. We distinguish among three classes:

- *canonical class*
- *ARMA class*
- *state space class*

In the remainder, we will mainly deal with the canonical class. In the canonical class we require that $K > 1$ and that $\mathbf{F}_{k0} = \mathbf{I}$ for all $k = 1 \ldots K$. As we will see later, the effect of these restrictions is that for the optimal solution the K hyperspaces $\underline{\mathbf{X}}_k$ - ($\underline{\mathbf{Z}}_k$ - $\mathbf{Z}$) will be as similar as possible in terms of squared distances between time points. If, in addition, it is true that $Q_k = 0$ for all k, then $\underline{\mathbf{Z}}_k - \mathbf{Z} = \mathbf{0}$, and the problem will reduce to a form of generalized canonical correlation analysis.

In the ARMA class we require that $K = 1$, $\mathbf{A}_0 = \mathbf{I}$ and $\mathbf{F}_0 = \mathbf{I}$. We then obtain the least squares loss function (2.31) for the ARMA(P,Q) model. More about the ARMA class will be said in section 6.9.

The state space class appears if we set $K = 2$, $P_1 = P_2 = Q_2 = 0$ and $Q_1 = 1$. By using zero row constraints on $\mathbf{A}_1$ and $\mathbf{A}_2$ we partition the variables into a set of inputs and a set of outputs. The state space class is discussed in more detail in section 6.10.

These three classes cover a great deal of the interesting analysis options, but they are certainly not an exhaustive partitioning of the omnibus. Moreover, there exists considerable overlap between them.

6.2 Canonical class: Definition

The loss function of the canonical class is defined by

$$\sigma(\mathbf{Z}; \mathbf{X}; \mathbf{A}_{kp}; \mathbf{F}_{kq}) = \sum_{k=1}^{K} \text{ssq}\left(\sum_{q=0}^{Q_k} \mathbf{B}_q \mathbf{Z} \mathbf{F}_{kq}, \sum_{p=0}^{P_k} \mathbf{B}_p \mathbf{X} \mathbf{A}_{kp} \right). \tag{6.5}$$

in which $K>1$ and $\mathbf{F}_{k0} = \mathbf{I}$ for all $k = 1 \ldots K$.

Since $\mathbf{B}_0 = \mathbf{I}$ we may write (6.5) as

$$\sigma(\mathbf{Z}; \mathbf{X}; \mathbf{A}_{kp}; \mathbf{F}_{kq}) = \sum_{k=1}^{K} \text{ssq}\,(\mathbf{Z}, (\sum_{p=0}^{P_k} \mathbf{B}_p\mathbf{X}\mathbf{A}_{kp} - \sum_{q=1}^{Q_k} \mathbf{B}_q\mathbf{Z}\mathbf{F}_{kq})), \tag{6.6}$$

so that the corresponding approximation structure is

$$\begin{aligned}
\mathbf{Z} &= \sum_{p=0}^{P_1} \mathbf{B}_p\mathbf{X}\mathbf{A}_{1p} - \sum_{q=1}^{Q_1} \mathbf{B}_q\mathbf{Z}\mathbf{F}_{1q} \\
&= \ldots \\
&= \sum_{p=0}^{P_k} \mathbf{B}_p\mathbf{X}\mathbf{A}_{kp} - \sum_{q=1}^{Q_k} \mathbf{B}_q\mathbf{Z}\mathbf{F}_{kq} \\
&= \ldots \\
&= \sum_{p=0}^{P_K} \mathbf{B}_p\mathbf{X}\mathbf{A}_{kp} - \sum_{q=1}^{Q_K} \mathbf{B}_q\mathbf{Z}\mathbf{F}_{kq}.
\end{aligned} \tag{6.7}$$

Except in degenerate cases, (6.7) will not be true. Deviations from the approximation structure introduce loss.

Let us define for each set k the R-dimensional variate subspace of all linear combinations

$$\mathbf{Z}_k = \sum_{p=0}^{P_k} \mathbf{B}_p\mathbf{X}\mathbf{A}_{kp} - \sum_{q=1}^{Q_k} \mathbf{B}_q\mathbf{Z}\mathbf{F}_{kq}. \tag{6.8}$$

Assuming that $\mathbf{Z}$ is of full rank, the columns of the N × R matrix $\mathbf{Z}$ span an R-dimensional subspace in $\Re^N$. For the minimum of (6.6) this subspace will be closest to the K R-dimensional variate subspaces $\mathbf{Z}_k$, also located in $\Re^N$. If all variate subspaces coincide, then (6.7) is true and the loss (6.6) is equal to zero. The goal is of course to find the K variate hyperplanes that are as similar as possible in the sense of (6.6). In terms of Section 6.1 we say that the K hyperspaces $\underline{\mathbf{X}}_k$ - ($\underline{\mathbf{Z}}_k$ - $\mathbf{Z}$) should have maximum correspondence.

If optimal scaling is required, we insert $\mathbf{X} = \mathbf{GY}$ into (6.5). The loss function then becomes

$$\sigma(\mathbf{Z}; \mathbf{Y}; \mathbf{A}_{kp}; \mathbf{F}_{kq}) = \sum_{k=1}^{K} \text{ssq} \left(\sum_{q=0}^{Q_k} \mathbf{B}_q \mathbf{Z} \mathbf{F}_{kq}, \sum_{p=0}^{P_k} \mathbf{B}_p \mathbf{G} \mathbf{Y} \mathbf{A}_{kp} \right). \tag{6.9}$$

There are four types of unknowns: $\mathbf{Z}$, $\mathbf{Y}$, $\mathbf{A}_{kp}$ and $\mathbf{F}_{kq}$. In order to find R distinct and nontrivial solutions for $\mathbf{Z}$ we impose the normalization restrictions $\mathbf{1}'\mathbf{Z} = \mathbf{0}$ and $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$ on (6.9). In many cases we restrict the quantified variables to have zero mean, i.e. $\mathbf{1}'\mathbf{X} = \mathbf{0}$, and equal dispersion, i.e. $\text{dg}(\mathbf{X}'\mathbf{X}) = \mathbf{I}$.

Note that the equality constraints on the quantifications are automatically satisfied. If $\mathbf{X} = \mathbf{GY}$ denotes the quantified data matrix, then $\mathbf{BX} = \mathbf{BGY}$ is the lagged quantified data matrix. Both $\mathbf{X}$ and $\mathbf{BX}$ use the same quantification matrix $\mathbf{Y}$. Consequently, equality constraints for lagged variables are always implicitly present.

We minimize (6.9) by a double majorization algorithm that consists of three main steps:

- minimization over $\mathbf{Y}$ for fixed $\mathbf{Z}$, $\mathbf{A}_{kp}$ and $\mathbf{F}_{kq}$ by majorization
- minimization over $\mathbf{Z}$ for fixed $\mathbf{Y}$, $\mathbf{A}_{kp}$ and $\mathbf{F}_{kq}$ by majorization
- minimization over $\mathbf{A}_{kp}$ and $\mathbf{F}_{kq}$ for fixed $\mathbf{Z}$ and $\mathbf{Y}$ by least squares

Each of the steps will lower the loss value, so that on iterative application of the steps, the procedure will converge to a minimum. Sections 6.4 to 6.8 discuss these steps in more detail. But before dealing with minimization problems, we prefer to show how the previous optimal scaling techniques relate to the canonical class.

6.3 Canonical class: Relations with other techniques

In this section we illustrate how the canonical class loss function reduces to the optimal scaling techniques discussed in Chapter 3, 4 and 5. Special cases can be derived by choosing a specific set of values for K, P_k and Q_k and by systematically applying the zero row constraint.

Let us first take a look at the case $P_k = Q_k = 0$ for all $k = 1 \ldots K$. Then there are no lagged variables of either $\mathbf{X}$ or $\mathbf{Z}$ and the loss function (6.5) becomes

$$\sigma(\mathbf{Z}; \mathbf{X}; \mathbf{A}_1 \ldots \mathbf{A}_K) = \sum_{k=1}^{K} \text{ssq}\,(\mathbf{Z}, \mathbf{X}\mathbf{A}_k). \tag{6.10}$$

Assuming that the variables have been partitioned into K sets by setting the appropriate rows of $\mathbf{A}_k$ to be zero, minimizing (6.10) is equivalent to minimizing (3.9) for single variables. Because the OVERALS loss function is the most general one considered in Gifi (1981), the canonical class covers the majority of nonlinear Gifi techniques such as homogeneity analysis, correspondence analysis and nonlinear versions of principal components analysis, regression analysis, discriminant analysis, MANOVA and others. However (6.10) is not the easiest representation of these nonlinear techniques, and we must use some tricks to derive them from loss function (6.10).

As an example, homogeneity analysis can be formulated, in a rather clumsy way though, in terms of (6.10) by making R copies of each variable so that $\mathbf{X}$ becomes of order $N \times MR$. Using zero row constraints we define K = M sets, each of which contains R copies of the j^{th} variable. In addition we restrict $\mathbf{A}_k$ such that only the r^{th} copy loads on the r^{th} dimension ($j, k = 1 \ldots M$; $r = 1 \ldots R$). If all MR variables are treated as nominal (single nominal, of course) the solution corresponds with homogeneity analysis. The dispersion matrix for the j^{th} variable is given by $\mathbf{A}_k'\mathbf{X}'\mathbf{X}\mathbf{A}_k$ and the multiple category quantifications can be found in the appropriate columns of $\mathbf{Y}$.

In Chapter 4 we defined the lag-1 predictor model as the problem of minimizing loss function (4.20):

$$\sigma_{arc}(\mathbf{z}; \mathbf{x}; a_0; a_1) = ssq(\mathbf{z}, \mathbf{x}a_0) + ssq(\mathbf{z}, \mathbf{B}\mathbf{x}a_1) \qquad (4.20)$$

over $\mathbf{z}$, $\mathbf{x}$, a_0 and a_1 under normalizations $\mathbf{1}'\mathbf{z} = \mathbf{1}'\mathbf{x} = 0$ and $\mathbf{z}'\mathbf{z} = \mathbf{x}'\mathbf{x} = 1$. If we choose $K = 2$, $M = R = P_2 = 1$, $Q_1 = Q_2 = P_1 = 0$ and restrict $a_{2,0} = 0$ then the canonical class loss function becomes

$$\sigma(\mathbf{z}; \mathbf{x}; a_{1,0}; a_{2,1}) = ssq(\mathbf{z}, \mathbf{x}a_{1,0}) + ssq(\mathbf{z}, \mathbf{B}\mathbf{x}a_{2,1}), \qquad (6.11)$$

which is equivalent to (4.20). The lag-p predictor model for seasonal autoregression is the same except for setting $P_2 = P$ and restricting $a_{2,0} \ldots a_{2,P-1} = 0$.

Multiple autoregression was defined as the problem of minimizing

$$\sigma_{mar}(\mathbf{z}; \mathbf{x}; a_0; a_p) = ssq(\mathbf{z}, \mathbf{x}a_0) + ssq(\mathbf{z}, \sum_{p=1}^{P} \mathbf{B}_p \mathbf{x}a_p) \qquad (4.27)$$

over $\mathbf{z}$, $\mathbf{x} = \mathbf{G}\mathbf{y}$, a_0 and a_p ($p = 1 \ldots P$) under normalizations $\mathbf{1}'\mathbf{z} = \mathbf{1}'\mathbf{x} = 0$ and $\mathbf{z}'\mathbf{z} = \mathbf{x}'\mathbf{x} = 1$. The canonical function reduces to (4.27) if we choose $K = 2$, $M = R = 1$, $Q_1 = Q_2 = P_1 = 0$, $P_2 = P$ and $a_{2,0} = 0$.

For intervention analysis we are looking for the minimum of

$$\sigma(\mathbf{z}; \mathbf{x}_i, \mathbf{x}_o; a_o, a, a_1 \ldots a_P) = ssq(\mathbf{z}, \mathbf{x}_o a_o) + ssq(\mathbf{z}, \mathbf{x}_i a + \sum_{p=1}^{P} \mathbf{B}_p \mathbf{x}_o a_p) \qquad (5.4)$$

under normalization restrictions

$$\mathbf{1}'\mathbf{z} = \mathbf{1}'\mathbf{x}_o = \mathbf{1}'\mathbf{x}_i = 0$$
$$\mathbf{z}'\mathbf{z} = \mathbf{1}'\mathbf{x}_o = \mathbf{1}'\mathbf{x}_i = 1.$$

In this case we have K = M = 2, R = 1, $Q_1 = Q_2 = P_1 = 0$ and $P_2 = P$. The canonical class function (6.6) transforms into

$$\sigma(\mathbf{Z}; \mathbf{X}; \mathbf{a}_{kp}) = \text{ssq}(\mathbf{z}, \mathbf{X}\mathbf{a}_{1,0}) + \text{ssq}(\mathbf{z}, \sum_{p=0}^{P} \mathbf{B}_p\mathbf{X}\mathbf{a}_{2,p}), \tag{6.12}$$

which becomes equal to (5.4) by restricting the second element of $\mathbf{a}_{1,0}$, the first element of $\mathbf{a}_{2,0}$ and the second element of $\mathbf{a}_{2,p}$ for $p > 1$ all to zero.

The predictable components technique comes down to minimizing

$$\sigma(\mathbf{Z}; \mathbf{X}; \mathbf{A}_0 \ldots \mathbf{A}_P) = \text{ssq}(\mathbf{Z}, \mathbf{X}\mathbf{A}_0) + \text{ssq}(\mathbf{Z}, \sum_{p=1}^{P} \mathbf{B}_p\mathbf{X}\mathbf{A}_p) \tag{5.11}$$

over $\mathbf{Z}$, $\mathbf{X} = \mathbf{G}\mathbf{Y}$ and $\mathbf{A}_0 \ldots \mathbf{A}_P$ under constraints $\mathbf{1}'\mathbf{Z} = \mathbf{0}$, $\mathbf{1}'\mathbf{X} = \mathbf{0}$, $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$ and $\text{dg}\,\mathbf{X}'\mathbf{X} = \mathbf{I}$. This is similar to the canonical class function with the setting $K = 2$, $M \geq 1$, $1 \leq R \leq M$, $Q_1 = Q_2 = P_1 = 0$, $P_2 = P$ and $\mathbf{A}_{2,0} = \mathbf{0}$, i.e.

$$\sigma(\mathbf{Z}; \mathbf{X}; \mathbf{A}_{kp}) = \text{ssq}(\mathbf{Z}, \mathbf{X}\mathbf{A}_{1,0}) + \text{ssq}(\mathbf{Z}, \sum_{p=1}^{P} \mathbf{B}_p\mathbf{X}\mathbf{A}_{2,p}). \tag{6.13}$$

Next, the loss function for dynamic components analysis is

$$\sigma(\mathbf{Z}; \mathbf{X}; \mathbf{A}, \mathbf{F}_1 \ldots \mathbf{F}_Q) = \text{ssq}(\mathbf{Z}, \mathbf{X}\mathbf{A}) + \text{ssq}(\mathbf{Z}, \sum_{q=1}^{Q} \mathbf{B}_q\mathbf{Z}\mathbf{F}_q) \tag{5.18}$$

which can be minimized over $\mathbf{Z}$, $\mathbf{X} = \mathbf{G}\mathbf{Y}$, $\mathbf{A}$ and $\mathbf{F}_1 \ldots \mathbf{F}_Q$ under normalizations $\mathbf{1}'\mathbf{Z} = \mathbf{0}$, $\mathbf{1}'\mathbf{X} = \mathbf{0}$, $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$ and $\text{dg}(\mathbf{X}'\mathbf{X}) = \mathbf{I}$. Choosing the values $K = 2$, $M \geq 1$, $1 \leq R \leq M$, $Q_1 = P_1 = P_2 = 0$ and $Q_2 = Q$ yields (6.6) as

$$\sigma(\mathbf{Z}; \mathbf{X}; \mathbf{A}_{kp}; \mathbf{F}_{kq}) = \text{ssq}(\mathbf{Z}, \mathbf{X}\mathbf{A}_{1,0}) + \text{ssq}(\mathbf{Z}, \sum_{q=1}^{Q} \mathbf{B}_q\mathbf{Z}\mathbf{F}_{2,q})). \tag{6.14}$$

Finally, multiset dynamic components analysis corresponds to minimizing

$$\sigma(\mathbf{Z}; \mathbf{X}; \mathbf{A}_k; \mathbf{F}_q) = \sum_{k=1}^{K} \text{ssq}\,(\mathbf{Z}, \mathbf{X}\mathbf{A}_k) + \text{ssq}\,(\mathbf{Z}, \sum_{q=1}^{Q} \mathbf{B}_q\mathbf{Z}\mathbf{F}_q), \qquad (5.23)$$

which has a total of K + 1 sets. Here $M \geq 1$, $1 \leq R \leq M$, $Q_k = P_k = P_k = 0$ for $k = 1...K$ and $P_{K+1} = 0$ and $Q_{K+1} = Q$.

6.4 Canonical class: Majorization over Y

We now direct our attention to the minimization problem. As we will see below, no useful partitioning of the loss function (6.9) is possible (i.e. the cross-products do not vanish) so we can not isolate simpler minimization subproblems that can be managed by least squares. It is then appropriate to use a majorization approach. Majorization algorithms have been applied before in the context of maximum likelihood estimation (Dempster et al. 1977), multidimensional scaling (De Leeuw, 1977, 1986; De Leeuw & Heiser, 1980; Heiser, 1981; Meulman, 1986) and dynamic linear models (De Leeuw & Bijleveld, 1988).

We need to majorize over two sets over unknowns: **Y** and **Z**. Therefore our complete algorithm will be a *double majorization algorithm*. In this section we derive the majorization of **Y**. The procedure described below is loosely based on the DYNAMALS majorization algorithm derived by De Leeuw and Bijleveld (1988), but in some respects it is more complicated. First, instead of dealing with two sets we deal with multiple sets. Second, we have many lags of **Z** whereas DYNAMALS has only one, and third, we consider lags of **X**.

We consider the problem of minimizing

$$\sum_{k=1}^{K} \text{ssq}\,(\sum_{q=0}^{Q_k} \mathbf{B}_q\mathbf{Z}\mathbf{F}_{kq}, \sum_{p=0}^{P_k} \mathbf{B}_p\mathbf{G}\mathbf{Y}\mathbf{A}_{kp}) \qquad (6.15)$$

over the block diagonal S × M matrix **Y** (S is the total number of categories) that holds the $k_j \times 1$ vector of category quantifications y_j in its j^{th} column. See Figure 3.1 for an illustration of the shape of **Y**. Vectors y_j should satisfy the appropriate level constraints as well as the normalizations $1'D_jy_j = 0$ and $y_j'D_jy_j = 1$ for all j = 1...M.

Note that we can not split the problem over the variables since

$$\sum_{k=1}^{K} \text{ssq}\left(\sum_{q=0}^{Q_k} B_qZF_{kq}, \sum_{p=0}^{P_k} B_pGYA_{kp}\right) \neq \sum_{j=1}^{M}\sum_{k=1}^{K} \text{ssq}\left(\sum_{q=0}^{Q_k} B_qZF_{kq}, \sum_{p=0}^{P_k} B_pG_jy_ja_{kpj}'\right)$$

if we have more variables in a set. Moreover we can not split the problem over the sets since we may have the same variable appearing in more than one set. We are obliged to consider all variables simultaneously.

Let us define $C_s = B_sG$, i.e. C_s is the s^{th} order lagged indicator matrix, and let

$$Y = Y^o + (Y - Y^o) = Y^o + \Delta \tag{6.16}$$

where Y^o is some old solution satisfying all appropriate constraints. Writing $\sigma(Y)$ for (6.15) as a function of **Y** only, and substituting (6.16) into it, the loss can be written as

$$\begin{aligned}\sigma(Y) &= \Sigma_k \text{ ssq } [\Sigma_q B_qZF_{kq}, \Sigma_p C_p(Y^o+\Delta)A_{kp}] \\ &= \Sigma_k \text{ ssq } [(\Sigma_q B_qZF_{kq} - \Sigma_p C_pY^oA_{kp}), \Sigma_p C_p\Delta A_{kp}]\end{aligned}$$

Furthermore we define

$$P_k = \Sigma_q B_qZF_{kq} - \Sigma_p C_pY^oA_{kp}$$

as the least squares residuals of the k^{th} set and we vectorize Δ as

$$\delta = \text{vec } \Delta.$$

Then

$$\begin{aligned}\sigma(\mathbf{Y}) &= \sigma(\mathbf{Y}^0) - 2\sum_k \mathrm{tr}\,\mathbf{P}_k'(\sum_p \mathbf{C}_p\Delta\mathbf{A}_{kp}) + \\ &\quad + \sum_k \mathrm{tr}\,(\sum_p \mathbf{C}_p\Delta\mathbf{A}_{kp})'(\sum_p \mathbf{C}_p\Delta\mathbf{A}_{kp}) \\ &= \sigma(\mathbf{Y}^0) - 2\,\mathrm{tr}\,\Delta'(\sum_k\sum_p \mathbf{C}_p'\mathbf{P}_k\mathbf{A}_{kp}') + \\ &\quad + \delta'[\sum_k (\sum_p \mathbf{A}_{kp}' \otimes \mathbf{C}_p)'(\sum_p \mathbf{A}_{kp}' \otimes \mathbf{C}_p)]\delta \\ &= \sigma(\mathbf{Y}^0) - 2\,\delta'\mathbf{u} + \delta'\mathbf{W}\delta \qquad (6.17)\end{aligned}$$

where

$$\mathbf{u} = \mathrm{vec}(\sum_k\sum_p \mathbf{C}_p'\mathbf{P}_k\mathbf{A}_{kp}')$$

and

$$\mathbf{W} = \sum_k (\sum_p \mathbf{A}_{kp}' \otimes \mathbf{C}_p)'(\sum_p \mathbf{A}_{kp}' \otimes \mathbf{C}_p).$$

By choosing $\alpha \geq \lambda_{max}(\mathbf{W})$, the maximum eigenvalue of the symmetric matrix $\mathbf{W}$, the loss function $\sigma(\mathbf{Y})$ is majorized by

$$\sigma(\mathbf{Y}) \leq \sigma(\mathbf{Y}^0) - 2\,\delta'\mathbf{u} + \alpha\delta'\delta, \qquad (6.18)$$

since

$$\delta'\mathbf{W}\delta \leq \alpha\delta'\delta$$

by the Rayleigh quotient inequality (cf. Magnus & Neudecker, 1988, 203). If we substitute for

$$\delta = \mathrm{vec}(\mathbf{Y} - \mathbf{Y}^0) = \mathrm{vec}(\mathbf{Y}) - \mathrm{vec}(\mathbf{Y}^0) = \mathbf{y} - \mathbf{y}^0$$

and compute the update vector $\mathbf{y}^u = \alpha^{-1}\mathbf{u}$, the problem of minimizing the right hand side over δ becomes equivalent to minimizing

$$(\mathbf{y} - (\mathbf{y}^o + \mathbf{y}^u))'(\mathbf{y} - (\mathbf{y}^o + \mathbf{y}^u)) \tag{6.19}$$

over $\mathbf{y}$. This problem has the simple solution $\mathbf{y} = \mathbf{y}^o + \mathbf{y}^u$ for unrestricted $\mathbf{y}$, but if $\mathbf{y}$ is subject to constraints, which is generally the case, the solution is also easy to find.

Note that we have not used the backshift property of $\mathbf{B}_s$ anywhere, so the majorization result holds for any real N × N matrix $\mathbf{B}_s$ and not just for backshift matrices.

Since $\mathbf{Y}$ is block diagonal with S(M-1) elements equal to zero, the optimal $\mathbf{y}$ will also contain S(M-1) zeroes. It follows that computing the complete update vector $\mathbf{y}^u$ is rather inefficient, because only S out of the SM values are actually being used for finding $\mathbf{y}$. The remaining S(M-1) elements of $\mathbf{y}^u$ are redundant, and need not be computed. Thus, instead of (6.19) we can minimize

$$(\mathbf{y}_j - (\mathbf{y}_j^o + \mathbf{y}_j^u))'(\mathbf{y}_j - (\mathbf{y}_j^o + \mathbf{y}_j^u)) \tag{6.20}$$

over $\mathbf{y}_j$ for each variable j = 1...M separately. The update vector $\mathbf{y}_j^u$ is given by

$$\mathbf{y}_j^u = \alpha^{-1}\, \mathbf{G}_j'(\Sigma_k \Sigma_p\, \mathbf{B}_p'\mathbf{P}_k\mathbf{a}_{kpj}'), \tag{6.21}$$

where $\mathbf{a}_{kpj}'$ is the j^{th} row of $\mathbf{A}_{kp}$ written as a column. The redundant elements are not computed by (6.21).

Solving each of the M subproblems of (6.20) under the appropriate constraints will decrease the total loss in (6.15). We may repeat the majorization steps until we have obtained a minimum for (6.15) but we could also proceed by minimizing the loss over one of the other types of unknowns, for example $\mathbf{Z}$.

6.5 Canonical class: Majorization over Z

We consider the problem of minimizing

$$\sum_{k=1}^{K} \text{ssq}\left(\sum_{q=0}^{Q_k} \mathbf{B}_q \mathbf{Z} \mathbf{F}_{kq}, \sum_{p=0}^{P_k} \mathbf{B}_p \mathbf{X} \mathbf{A}_{kp}\right) \tag{6.22}$$

over the N × R matrix **Z** of object scores. The solution should satisfy normalization constraints $\mathbf{1}'\mathbf{Z} = \mathbf{0}$ and $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$.

Remember that the columns of **Z** span the R-dimensional hyperplane that is a subspace of the hyperspace spanned by the columns of $\mathbf{Z}_k$ as given in (6.8). In the case that $Q_k = 0$ for all k = 1...K, we find the unconstrained minimum of (6.22) by simply averaging, i.e.

$$\mathbf{Z} = K^{-1} \sum_k \mathbf{Z}_k. \tag{6.23}$$

Solving the two-sided Procrustes problem of finding the best fitting orthonormal **Z** results in the desired constrained solution. But if $Q_k > 0$ for some k this will not work because $\mathbf{Z}_k$ is in its turn dependent on **Z**, and so we need a more complicated method then.

Majorizing (6.22) over **Z** can be done in more or less the same way as for **Y**. Let

$$\mathbf{Z} = \mathbf{Z}^o + (\mathbf{Z} - \mathbf{Z}^o) = \mathbf{Z}^o + \Delta,$$

then (6.22) can be written as a function of **Z** only as

$$\begin{aligned} \sigma(\mathbf{Z}) &= \sum_k \text{ssq}\left[\sum_q \mathbf{B}_q(\mathbf{Z}^o+\Delta)\mathbf{F}_{kq}, \sum_p \mathbf{B}_p \mathbf{X} \mathbf{A}_{kp}\right] \\ &= \sum_k \text{ssq}\left[\left(\sum_p \mathbf{B}_p \mathbf{X} \mathbf{A}_{kp} - \sum_q \mathbf{B}_q \mathbf{Z}^o \mathbf{F}_{kq}\right), \sum_q \mathbf{B}_q \Delta \mathbf{F}_{kq}\right] \end{aligned} \tag{6.24}$$

Let

$$\mathbf{P}_k = \sum_p \mathbf{B}_p \mathbf{X} \mathbf{A}_{kp} - \sum_q \mathbf{B}_q \mathbf{Z}^o \mathbf{F}_{kq}$$

then

$$\begin{aligned}\sigma(\mathbf{Z}) &= \sigma(\mathbf{Z}^o) - 2 \sum_k \mathrm{tr}\, \mathbf{P}_k' (\sum_q \mathbf{B}_q \Delta \mathbf{F}_{kq}) + \\ &\quad + \sum_k \mathrm{tr}\, (\sum_q \mathbf{B}_q \Delta \mathbf{F}_{kq})' (\sum_q \mathbf{B}_q \Delta \mathbf{F}_{kq}) \\ &= \sigma(\mathbf{Z}^o) - 2\, \mathrm{tr}\, \Delta' (\sum_k \sum_q \mathbf{B}_q' \mathbf{P}_k \mathbf{F}_{kq}') + \\ &\quad + \delta' [\sum_k (\sum_q \mathbf{F}_{kq}' \otimes \mathbf{B}_q)' (\sum_q \mathbf{F}_{kq}' \otimes \mathbf{B}_q)] \delta \\ &= \sigma(\mathbf{Z}^o) - 2\, \delta' \mathbf{u} + \delta' \mathbf{W} \delta \end{aligned} \tag{6.25}$$

where

$$\mathbf{u} = \mathrm{vec}(\sum_k \sum_q \mathbf{B}_q' \mathbf{P}_k \mathbf{F}_{kq}')$$

and

$$\mathbf{W} = \sum_k (\sum_q \mathbf{F}_{kq}' \otimes \mathbf{B}_q)' (\sum_q \mathbf{F}_{kq}' \otimes \mathbf{B}_q).$$

Choose $\gamma = \lambda_{max}(\mathbf{W})$ and let

$$\mathbf{Z}^u = \gamma^{-1} \sum_k \sum_q \mathbf{B}_q' \mathbf{P}_k \mathbf{F}_{kq}'.$$

Following the same reasoning as before, we must then minimize the majorized loss function

$$(\mathbf{z} - (\mathbf{z}^o + \mathbf{z}^u))' (\mathbf{z} - (\mathbf{z}^o + \mathbf{z}^u)) = \mathrm{ssq}\, (\mathbf{Z}, (\mathbf{Z}^o + \mathbf{Z}^u)) \tag{6.26}$$

over $\mathbf{Z}$ under constraints $\mathbf{1}'\mathbf{Z} = \mathbf{0}$ and $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$. This can be done by computing the singular value decomposition of the column centered version $\mathbf{Z}^*$ of $\mathbf{Z}^o + \mathbf{Z}^u$, i.e. $\mathbf{Z}^* = \mathbf{K}\Phi\mathbf{L}'$, and setting $\mathbf{Z} = \mathbf{K}\mathbf{L}'$.

Again, we have not used any properties of $\mathbf{B}_s$, so the result holds for general N × N matrices $\mathbf{B}_s$. The problem (6.22) is solved by repeated application of the majorization steps until the solution becomes stable.

6.6 Canonical class: Estimation of A and F

As a last step, we consider the problem of minimizing

$$\sum_{k=1}^{K} \text{ssq}\left(\sum_{q=0}^{Q_k} \mathbf{B}_q \mathbf{Z} \mathbf{F}_{kq}, \sum_{p=0}^{P_k} \mathbf{B}_p \mathbf{X} \mathbf{A}_{kp}\right) \tag{6.27}$$

over the matrices $\mathbf{A}_{kp}$ of component loadings of order M × R and over the matrices $\mathbf{F}_{kq}$ of weights of order R × R. For the canonical class it is known that $\mathbf{F}_{k0} = \mathbf{I}$ for k = 1...K. Furthermore, we allow for linear restrictions on the parameters, for example, we deselect a variable for the k^{th} set by setting the corresponding rows in $\mathbf{A}_{k0}...\mathbf{A}_{kPk}$ to zero.

Let us define $\mathbf{X}_k^*$ as the data matrix for the k^{th} set in which all columns corresponding to the variables not belonging to set k have been deleted. Likewise, we define $\mathbf{A}_{kp}^*$ as the loading matrices with its zero rows deleted. We then construct K matrices $\mathbf{P}_k$ as

$$\mathbf{P}_k = [\mathbf{X}_k^*, \mathbf{B}_1 \mathbf{X}_k^*, \ldots, \mathbf{B}_{Pk} \mathbf{X}_k^*, -\mathbf{B}_1 \mathbf{Z}, \ldots, -\mathbf{B}_{Qk} \mathbf{Z}], \tag{6.28}$$

and K matrices $\mathbf{U}_k$ of weights as

$$\mathbf{U}_k = [\mathbf{A}_{k0}^{*\prime}, \ldots, \mathbf{A}_{kPk}^{*\prime}, \mathbf{F}_{k1}^{\prime}, \ldots, \mathbf{F}_{kQk}^{\prime}]'. \tag{6.29}$$

Since $\mathbf{F}_{k0} = \mathbf{I}$ for k = 1...K, we may write (6.27) more conveniently as a function of **A**'s and **F**'s only, as K multivariate multiple regression problems of minimizing

$$\sigma(\mathbf{U}_1 \ldots \mathbf{U}_K) = \Sigma_k \text{ ssq } (\mathbf{Z}, \mathbf{P}_k \mathbf{U}_k). \tag{6.30}$$

Splitting (6.30) over the columns, estimating the elements of **A** and **F** then involves solving the KR independent least squares problems

$$\text{ssq}\,(\mathbf{z}_r, \mathbf{P}_k\mathbf{u}_{kr}) \tag{6.31}$$

each of which has the general solution $\mathbf{u}_{kr} = \mathbf{P}_k{}^{+}\mathbf{z}_r$. Linear restrictions on the regression weights may be satisfied by applying the results of Section 3.12 to $\mathbf{U}_{kr}$. We must be careful however since the use of additional linear restrictions may have undesirable side effects on the normalization of the solution. If $\mathbf{P}_k$ has R columns, we may also apply orthogonality restrictions on $\mathbf{U}_k$ by using the method of Section 3.13.

6.7 Canonical class: First algorithm

The steps outlined in Sections 6.4, 6.5 and 6.6 minimize different sets of parameters. Since each of the procedures decreases the loss (6.6) over a specific set of parameters, alternating these steps will lead to a minimum loss value. In this way convergence is guaranteed, although the obtained minimum is only locally optimal. Since two of the steps involve majorization, the algorithm is of a double majorization type.

Combining the three procedures leads to an algorithm with a total of six step. They are:

A	*Initialization*	
B	*Estimation of* **F** *and* **A**	⎫
C	*Estimation of* **Y**	⎬ Iteration loop
D	*Estimation of* **Z**	⎭
E	*Convergence test*	
F	*Final rotation*	

Each substep is treated in detail below.

Before the procedure can do anything, it needs the following input:

- the data matrix $\mathbf{H}$ ($N \times M$)
- specification of the measurement level of each variable
- specification of the dimensionality R
- specification of the set partitioning
- specification of the lag orders P_k and Q_k for $k = 1..K$
- convergence test value: $\varepsilon > 0$
- maximum number of iterations: MAXIT

Initialization

A1 Construct the super-indicator matrix $\mathbf{G}$. Scale continuous variables according to $\mathbf{1}'\mathbf{G}_j = 0$ and $\mathbf{G}_j'\mathbf{G}_j = 1$. Let $\mathbf{D}_j = \mathbf{G}_j'\mathbf{G}_j$ for $j = 1...M$.

A2 Define $\mathbf{C}_p \leftarrow \mathbf{B}_p\mathbf{G}$ for $p \geq 0$.

A3 Initialize $\mathbf{Z}$ with random numbers such that $\mathbf{1}'\mathbf{Z} = \mathbf{0}$ and $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$.

A4 For categorical variables compute initial category quantifications $\mathbf{y}_j$ for $j = 1...M$ as the first quantifications of the unlagged variable from a R-dimensional homogeneity analysis of all categorical variables and active lags of them. Scale them according to $\mathbf{1}'\mathbf{D}_j\mathbf{y}_j = 0$ and $\mathbf{y}_j'\mathbf{D}_j\mathbf{y}_j = 1$. For continuous variables set $\mathbf{y}_j = 1$. Collect $\mathbf{y}_j$ ($j = 1...M$) into the block diagonal matrix $\mathbf{Y}$ of order $S \times M$. Use $\mathbf{Z}$ as object scores matrix.

A5 $\mathbf{X} \leftarrow \mathbf{GY}$. The initial quantified data matrix.

A6 Initialize $\mathbf{F}_{k0} = \mathbf{I}$ for $k = 1...K$.

A7 Initialize $\mathbf{F}$ and $\mathbf{A}$ with steps C.1 - C.4

A8 Set $t \leftarrow 0$

A9 Compute the initial loss value σ_t according to (6.5)

The function of the initialization is to set up reasonable starting values. The procedure employs a homogeneity analysis to find initial quantification vectors. As a by-product we also get an optimized version of $\mathbf{Z}$.

Estimation of $\mathbf{Y}$

B1 Repeat for each set $k = 1...K$

B2 $\mathbf{P}_k \leftarrow \Sigma_{q=0,Q_k} \mathbf{B}_q \mathbf{Z} \mathbf{F}_{kq} - \Sigma_{p=0,P_k} \mathbf{C}_p \mathbf{Y} \mathbf{A}_{kp}$

B3 $\mathbf{R}_k \leftarrow \Sigma_p \mathbf{A}_{kp}' \otimes \mathbf{C}_p$

B4 $\mathbf{Q}_k \leftarrow \Sigma_p \mathbf{C}_p' \mathbf{P}_k \mathbf{A}_{kp}'$

B5 $\mathbf{W} \leftarrow \Sigma_k \mathbf{R}_k' \mathbf{R}_k$

B6 $\alpha \leftarrow \lambda_{max}(\mathbf{W})$

B7 $\mathbf{U} \leftarrow \alpha^{-1} \Sigma_k \mathbf{Q}_k$

B8 Repeat for each variable $j = 1...M$

B9 if j is nominal: $\mathbf{y}_j \leftarrow \mathbf{y}_j + \mathbf{u}_j$

if j is ordinal: $\mathbf{y}_j \leftarrow$ MONREG($\mathbf{y}_j + \mathbf{u}_j$)

B10 standardize $\mathbf{y}_j$ such that $1'\mathbf{D}_j\mathbf{y}_j = 0$ and $\mathbf{y}_j'\mathbf{D}_j\mathbf{y}_j = 1$

B11 $\mathbf{X} \leftarrow \mathbf{G}\mathbf{Y}$

The Y-step is a direct translation of the majorization results derived in Section 6.4. MONREG() is used to insure that the quantifications are in proper order. It performs a monotone regression of its argument on the category order. The result of the procedure is an updated $\mathbf{Y}$ and an updated $\mathbf{X}$.

Estimation of $\mathbf{F}$ *and* $\mathbf{A}$

C1 Repeat for each set $k = 1...K$

C2 $\mathbf{P}_k \leftarrow [\mathbf{X}_k^*, \mathbf{B}_1\mathbf{X}_k^*, \ldots, \mathbf{B}_{P_k}\mathbf{X}_k^*, -\mathbf{B}_1\mathbf{Z}, \ldots, -\mathbf{B}_{Q_k}\mathbf{Z}]$

C3 $\mathbf{U}_k \leftarrow (\mathbf{P}_k)^+\mathbf{Z}$

C4 Store the rows of $\mathbf{U}_k$ into the appropriate rows of $\mathbf{A}_k$ and $\mathbf{F}_k$.

The definition of $\mathbf{P}_k$ according to (6.28) in step C2 takes care of the zero row constraints in $\mathbf{A}_k$ and $\mathbf{F}_k$. Step C3 is the least squares projection. The procedure results in updated versions of $\mathbf{F}_{kq}$ and $\mathbf{A}_{kp}$.

Estimation of $\mathbf{Z}$

D1 Repeat for each set k = 1...K

D2 $\quad \mathbf{P}_k \leftarrow \Sigma_{p=0,Pk}\, \mathbf{B}_p\mathbf{X}\mathbf{A}_{kp} - \Sigma_{q=0,Qk}\, \mathbf{B}_q\mathbf{Z}\mathbf{F}_{kq}$

D3 $\quad \mathbf{R}_k \leftarrow \Sigma_q\, \mathbf{F}_{kq}' \otimes \mathbf{B}_q$

D4 $\quad \mathbf{Q}_k \leftarrow \Sigma_q\, \mathbf{B}_q'\mathbf{P}_k\mathbf{F}_{kq}'$

D5 $\mathbf{W} \leftarrow \Sigma_k\, \mathbf{R}_k'\mathbf{R}_k$

D6 $\gamma \leftarrow \lambda_{max}\,(\mathbf{W})$

D7 $\mathbf{U} \leftarrow \gamma^1\, \Sigma_k\, \mathbf{Q}_k$

D8 $\mathbf{Z}^* \leftarrow$ DEVMN $(\mathbf{Z} + \mathbf{U})$

D9 Compute the singular value decomposition $\mathbf{Z}^* = \mathbf{K}\Phi\mathbf{L}'$

D10 $\mathbf{Z} \leftarrow \mathbf{K}\mathbf{L}'$

This Z-step is a straightforward implementation of the majorization results of Section 6.5. The DEVMN() function transforms its argument into a column centered matrix with zero mean.

Convergence test

E1 $t \leftarrow t+1$

E2 Compute σ_t using (6.5)

E3 If $\varepsilon < \sigma_{t-1} - \sigma_t$ and $t <$ MAXIT then goto B1

Final rotation

F1 Repeat for each set k = 1...K

F2 $\quad \mathbf{P}_k \leftarrow \Sigma_{p=0,Pk}\, \mathbf{B}_p\mathbf{X}\mathbf{A}_{kp} - \Sigma_{q=1,Qk}\, \mathbf{B}_q\mathbf{Z}\mathbf{F}_{kq}$

F3 $\mathbf{W} \leftarrow \Sigma_k\, \mathbf{P}_k'\mathbf{P}_k$

F4 Compute the eigenvector decomposition $\mathbf{W} = \mathbf{Q}\Phi\mathbf{Q}'$

F5 $\mathbf{Z} \leftarrow \mathbf{Z}\mathbf{Q}$

F6 Repeat steps B and C

The final rotation step rotates the entire solution such that its first dimension has the lowest contribution to the loss value. The rotation does not change the total amount of loss.

Since the majorization results hold for general $\mathbf{B}_S$ the entire minimization procedure can also be applied to other forms of dependencies among

objects, as for example in spatially ordered data. Bijleveld (1989) discusses other useful forms of $\mathbf{B}_s$. In this work however, we limit ourselves to backshift matrices and we have not actually tested the procedure for other kinds of dependency structures.

The next section describes how we can obtain a considerable increase in efficiency by explicitely using some properties of the backshift matrix.

6.8 Canonical class: Second algorithm

The previous algorithm spends most of its time in computing the $\mathbf{W}$ matrix (steps B3, B5, D3, D5) and in determining the largest eigenvalue of $\mathbf{W}$ (steps B6, D6). These steps only serve to compute values for the "gain factors" α^{-1} and γ^{-1} and in this sense they play only a limited role in the algorithm. We now show how we are able to obtain a significant reduction in computing time by using the backshift properties of $\mathbf{B}_s$ and by applying a theorem by Wolkowicz and Styan (1980).

The factors α and γ have been introduced as elements of the Rayleigh inequality in the majorizations with respect to $\mathbf{Y}$ and $\mathbf{Z}$ in Sections 6.4 and 6.5. The majorization results hold for $\alpha \geq \lambda_{max}(\mathbf{W})$ and $\gamma \geq \lambda_{max}(\mathbf{W})$. Choosing α and γ equal to the largest eigenvalue yields optimal iteration step lengths. For larger α and γ the step length will be suboptimal since the gain factors α^{-1} and γ^{-1} will become smaller. However, because the computation of $\lambda_{max}(\mathbf{W})$ can be quite time consuming, even if we use simple vector iteration, it is more efficient to set α and γ equal the upper bound of $\lambda_{max}(\mathbf{W})$ as derived by Wolkowicz and Styan (1980). This bound depends on only tr $(\mathbf{W})$ and tr $(\mathbf{W}^2)$. This has the advantage that less work per iteration is needed. The disadvantage is that more iterations will be used, but the advantage more than counterbalances this.

Theorem (Wolkowicz & Styan, 1980)

If the B × B matrix **W** has real eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_B$ then

$$C + A\,(B - 1)^{-1/2} \leq \lambda_1 \leq C + A\,(B - 1)^{1/2} \tag{6.32}$$

$$C - A\,(B - 1)^{1/2} \leq \lambda_B \leq C - A\,(B - 1)^{-1/2} \tag{6.33}$$

where $C = B^{-1}\ \mathrm{tr}\,(\mathbf{W})$ and $A^2 = B^{-1}\ \mathrm{tr}\,(\mathbf{W}^2) - C^2$. Equality holds on the left (right) if and only if the B - 1 largest (smallest) eigenvalues are equal.

If we set $\alpha = C + A\,(B - 1)^{1/2}$ we have a computational cheap upper bound of λ_{max}. Usually it will be true that $\lambda_{max} \leq \alpha \leq 2\,\lambda_{max}$, so the change in the iteration step length is not dramatic. We may thus replace the right hand sides of B6 and D6 by $C + A\,(B - 1)^{1/2}$, where C and A are defined as above, thereby increasing the efficiency of the algorithm.

The computation of the **W** matrix in steps B3, B5, D3 and D5 relies on Kronecker products. While Kronecker products are wonderful tools for algebraic manipulations, they are awkward for computational purposes since they tend to introduce very large arrays. However, by using certain properties of the backshift matrix we are able to eliminate all Kronecker products from the algorithm. We first deal with the Z-step.

Steps D3 and D5 compute the NR × NR matrix

$$\mathbf{W} = \sum_{k=1}^{K} \Big(\sum_{q=0}^{Q_k} \mathbf{F}_{kq}' \otimes \mathbf{B}_q\Big)'\Big(\sum_{q=0}^{Q_k} \mathbf{F}_{kq}' \otimes \mathbf{B}_q\Big), \tag{6.34}$$

which is the sum of

$$\mathbf{W}_k = \Big(\sum_{q=0}^{Q_k} \mathbf{F}_{kq}' \otimes \mathbf{B}_q\Big)'\Big(\sum_{q=0}^{Q_k} \mathbf{F}_{kq}' \otimes \mathbf{B}_q\Big). \tag{6.35}$$

Since

$$\lambda_{max}(\mathbf{W}) \leq \sum_{k=1}^{K} \lambda_{max}(\mathbf{W_k}) \tag{6.36}$$

for any symmetric $\mathbf{W_k}$, we find an upper bound on $\lambda_{max}(\mathbf{W})$ by the right hand side of (6.36). Let γ and γ_k denote the Wolkowicz-Styan upper bounds on $\lambda_{max}(\mathbf{W})$ and $\lambda_{max}(\mathbf{W_k})$ respectively. Then it will be true that

$$\lambda_{max}(\mathbf{W}) \leq \sum_{k=1}^{K} \lambda_{max}(\mathbf{W_k}) \leq \sum_{k=1}^{K} \gamma_k. \tag{6.37}$$

The advantage of using γ_k instead of γ is that by using backshift properties of $\mathbf{B_q}$ the bounds γ_k can be expressed as simple functions of $\mathbf{F_{kq}}$ only. So by combining inequalities (6.32) and (6.37) we do not need to compute either $\lambda_{max}(\mathbf{W})$ or the Kronecker products contained in (6.35).

To see how γ_k can be written in terms of $\mathbf{F_{kq}}$ we first note that it can be demonstrated that

$$(\Sigma_q \mathbf{F_{kq}}' \otimes \mathbf{B_q})'(\Sigma_q \mathbf{F_{kq}}' \otimes \mathbf{B_q})$$

and

$$(\Sigma_q \mathbf{B_q} \otimes \mathbf{F_{kq}}')'(\Sigma_q \mathbf{B_q} \otimes \mathbf{F_{kq}}')$$

have the same eigenvalues for any $\mathbf{F_{kq}}'$ and $\mathbf{B_q}$.

Let

$$\mathbf{R_k} = \Sigma_q \mathbf{B_q} \otimes \mathbf{F_{kq}}' \tag{6.38}$$

and let $\mathbf{W_k} = \mathbf{R_k}'\mathbf{R_k}$. In order to compute γ_k we are interested in finding convenient expressions for tr $(\mathbf{W_k})$ and tr $(\mathbf{W_k}\mathbf{W_k})$. Using the fact that $\mathbf{B_q}$ is a backshift matrix, it can be seen that $\mathbf{R_k}$ and $\mathbf{W_k}$ are patterned as in

$$\mathbf{R}_k = \begin{bmatrix} \mathbf{F}_0' & & & & & & \\ \mathbf{F}_1' & \mathbf{F}_0' & & & & & \\ \mathbf{F}_2' & \mathbf{F}_1' & \mathbf{F}_0' & & & \mathbf{0} & \\ & \mathbf{F}_2' & \mathbf{F}_1' & \cdots\cdots & \mathbf{F}_0' & & \\ & & \mathbf{F}_2' & & \mathbf{F}_1' & \mathbf{F}_0' & \\ & & & & \mathbf{F}_2' & \mathbf{F}_1' & \mathbf{F}_0' \\ & \mathbf{0} & & & & \mathbf{F}_2' & \mathbf{F}_1' & \mathbf{F}_0' \end{bmatrix}$$

Note: subscript k has been left out

$$\mathbf{W}_k = \begin{bmatrix} \mathbf{F}_0\mathbf{F}_0' + \mathbf{F}_1\mathbf{F}_1' + \mathbf{F}_2\mathbf{F}_2' & \mathbf{F}_1\mathbf{F}_0' + \mathbf{F}_2\mathbf{F}_1' & & & \\ \mathbf{F}_0\mathbf{F}_1' + \mathbf{F}_1\mathbf{F}_2' & \mathbf{F}_0\mathbf{F}_0' + \mathbf{F}_1\mathbf{F}_1' + \mathbf{F}_2\mathbf{F}_2' & & \mathbf{0} & \\ \mathbf{F}_0\mathbf{F}_2' & \mathbf{F}_0\mathbf{F}_1' + \mathbf{F}_1\mathbf{F}_2' & \cdots\cdots & & \\ & \mathbf{F}_0\mathbf{F}_2' & & & \\ & & & \mathbf{F}_2\mathbf{F}_0' & \\ & \mathbf{0} & & \mathbf{F}_1\mathbf{F}_0' + \mathbf{F}_2\mathbf{F}_1' & \mathbf{F}_2\mathbf{F}_0' \\ & & & \mathbf{F}_0\mathbf{F}_0' + \mathbf{F}_1\mathbf{F}_1' & \mathbf{F}_1\mathbf{F}_0' \\ & & & \mathbf{F}_0\mathbf{F}_1' & \mathbf{F}_0\mathbf{F}_0' \end{bmatrix}$$

Figure 6.1 Matrix patterns under the backshift matrix

Figure 6.1 for $Q_k = 2$. Counting the trace blocks of $\mathbf{W}_k$, we obtain

$$\text{tr}\,(\mathbf{W}_k) = \sum_{q=0}^{Q_k} (N\text{-}q)\ \text{tr}\,(\mathbf{F}_{kq}'\mathbf{F}_{kq}). \tag{6.39}$$

$\tilde{\mathbf{F}}_k$ = Subscript k has been left out $\begin{bmatrix} \mathbf{F}_0 & \mathbf{F}_1 & \mathbf{F}_2 \\ \mathbf{F}_0 & \mathbf{F}_1 & \\ \mathbf{F}_0 & & \end{bmatrix}$

$\tilde{\tilde{\mathbf{F}}}_k$ = Subscript k has been left out $\begin{bmatrix} \mathbf{F}_0' & \mathbf{F}_1' & \mathbf{F}_2' \\ \mathbf{F}_1' & \mathbf{F}_2' & \\ \mathbf{F}_2' & & \end{bmatrix}$

$\mathbf{T}_2 =$

	R			R			R		
	N-Q	N-Q	N-Q						
	N-Q	N-Q	N-Q						
	N-Q	N-Q	N-Q						
	1	1	1	2(N-Q)	2(N-Q)	2(N-Q)			
	1	1	1	2(N-Q)	2(N-Q)	2(N-Q)			
	1	1	1	2(N-Q)	2(N-Q)	2(N-Q)			
	1	1	1	2	2	2	2(N-Q)	2(N-Q)	2(N-Q)
	1	1	1	2	2	2	2(N-Q)	2(N-Q)	2(N-Q)
	1	1	1	2	2	2	2(N-Q)	2(N-Q)	2(N-Q)

Figure 6.2 Patterned matrices for computing tr(WW) (Q=2, R = 3)

An expression for tr $(\mathbf{W}_k\mathbf{W}_k)$ can be found in an analogous way, but this involves a lot of administrative tasks which we shall not discuss here. The result can be expressed, though not very elegantly, as the sum of elements of a $R(Q_k + 1) \times R(Q_k + 1)$ matrix

$$\mathbf{T}_q \odot \mathbf{S}_k \odot \mathbf{S}_k \tag{6.40}$$

where $\mathbf{T}_q$ is a design matrix of lag order Q_k, where

$$\mathbf{S}_k = \tilde{\mathbf{F}}_k \tilde{\tilde{\mathbf{F}}}_k. \tag{6.41}$$

Matrices $\tilde{\mathbf{F}}_k$ and $\tilde{\tilde{\mathbf{F}}}_k$ have $\mathbf{F}_{k0} \ldots \mathbf{F}_{kQk}$ arranged in a specific pattern as in Figure 6.2.

The $\odot$ operator denotes the Hadamard matrix product. The Hadamard product of two m $\times$ n matrices **A** and **B** is also an m $\times$ n matrix and its i,jth element is $a_{ij}b_{ij}$.

The steps D3, D5 and D6 may thus be replaced by the computation of the sum of the K eigenvalue upper bounds, which is much more economical than computing the largest eigenvalue of the NR $\times$ NR matrix **W**.

For the Y-step, we can derive expressions for tr ($\mathbf{W}_k$) and tr ($\mathbf{W}_k\mathbf{W}_k$) in the same way. However, this is a rather tedious enterprise since these traces not only depend on $\mathbf{A}_{kp}$ but also on **G**, and so, the matrices become less sparse. A better way is to rewrite

$$\mathbf{W} = \Sigma_k (\Sigma_p \mathbf{C}_p \otimes \mathbf{A}_{kp}')'(\Sigma_p \mathbf{C}_p \otimes \mathbf{A}_{kp}') \tag{6.42}$$

as

$$\mathbf{W} = (\mathbf{G} \otimes \mathbf{I}_M)' [\Sigma_k (\Sigma_p \mathbf{B}_p \otimes \mathbf{A}_{kp}')'(\Sigma_p \mathbf{B}_p \otimes \mathbf{A}_{kp}')] (\mathbf{G} \otimes \mathbf{I}_M) \tag{6.43}$$

by inserting $\mathbf{C}_p = \mathbf{B}_p\mathbf{G}$ and by applying some elementary properties of the Kronecker product. Equation (6.43) is of the form $\mathbf{W} = \mathbf{R}'\mathbf{P}\mathbf{R}$, which makes it possible to use the inequality

$$\lambda_{max} (\mathbf{R}'\mathbf{P}\mathbf{R}) \leq \lambda_{max}(\mathbf{P}) \lambda_{max}(\mathbf{R}'\mathbf{R}) \tag{6.44}$$

which holds for any square and positive semidefinite matrix **P** (Magnus & Neudecker, 1988, 237). An upper bound for $\lambda_{max}(\mathbf{P})$ can be obtained in a way comparable to the Z-step by applying equations (6.37), (6.39) and (6.40) to **P**.

As a last result we note that the eigenvalue $\lambda_{max}(\mathbf{R}'\mathbf{R})$ in (6.44) is constant. It is equal to

$$\lambda_{max}\,(\mathbf{G} \otimes \mathbf{I}_M)'(\mathbf{G} \otimes \mathbf{I}_M) = \lambda_{max}\,(\mathbf{G}'\mathbf{G} \otimes \mathbf{I}_M) = \lambda_{max}\,(\mathbf{G}'\mathbf{G}). \qquad (6.45)$$

Thus, the upper bound α can be derived in the same manner as γ. So by using the backshift property of $\mathbf{B}_s$ all Kronecker products and most eigenvalue computations vanish. As a result, the algorithm becomes much faster.

The improved Y-step becomes:

Estimation of $\mathbf{Y}$ *(improved)*

B1 Repeat for each set $k = 1 \ldots K$

B2 $\quad \mathbf{P}_k \leftarrow \Sigma_{q=0,Q_k}\, \mathbf{B}_q \mathbf{Z} \mathbf{F}_{kq} - \Sigma_{p=0,P_k}\, \mathbf{C}_p \mathbf{Y} \mathbf{A}_{kp}$

B3 $\quad \mathbf{Q}_k \leftarrow \Sigma_p\, \mathbf{C}_p' \mathbf{P}_k \mathbf{A}_{kp}'$

B4 $\quad \alpha_k \leftarrow$ BOUND $(\mathbf{A}_{kp})$

B5 $\alpha \leftarrow \lambda_{max}(\mathbf{G}'\mathbf{G})\, \Sigma_k\, \alpha_k$

B6 $\mathbf{U} \leftarrow \alpha^{-1}\, \Sigma_k\, \mathbf{Q}_k$

B7 Repeat for each variable $j = 1 \ldots M$

B8 $\quad$ if j is nominal: $\mathbf{y}_j \leftarrow \mathbf{y}_j + \mathbf{u}_j$

$\quad$ if j is ordinal: $\mathbf{y}_j \leftarrow$ MONREG$(\mathbf{y}_j + \mathbf{u}_j)$

B9 $\quad$ standardize $\mathbf{y}_j$ such that $\mathbf{1}'\mathbf{D}_j\mathbf{y}_j = 0$ and $\mathbf{y}_j'\mathbf{D}_j\mathbf{y}_j = 1$

B10 $\mathbf{X} \leftarrow \mathbf{G}\mathbf{Y}$

The BOUND() function computes the upper bound on $\lambda_{max}(\mathbf{W})$ using the method outlined above.

Similarly, the Z-step turns into:

Estimation of $\mathbf{Z}$ *(improved)*

D1 Repeat for each set k = 1..K

D2 $\quad \mathbf{P}_k \leftarrow \Sigma_{p=0,P_k} \mathbf{B}_p\mathbf{X}\mathbf{A}_{kp} - \Sigma_{q=0,Q_k} \mathbf{B}_q\mathbf{Z}\mathbf{F}_{kq}$

D3 $\quad \mathbf{Q}_k \leftarrow \Sigma_q \mathbf{B}_q'\mathbf{P}_k\mathbf{F}_{kq}'$

D4 $\quad \gamma_k \leftarrow$ BOUND($\mathbf{F}_{kq}$)

D5 $\gamma \leftarrow \Sigma_k \gamma_k$

D6 $\mathbf{U} \leftarrow \gamma^{-1} \Sigma_k \mathbf{Q}_k$

D7 $\mathbf{Z}^* \leftarrow$ DEVMN ($\mathbf{Z}$ + $\mathbf{U}$)

D8 Compute the singular value decomposition $\mathbf{Z}^* = \mathbf{K}\Phi\mathbf{L}'$

D9 $\mathbf{Z} \leftarrow \mathbf{K}\mathbf{L}'$

The improved algorithm was implemented as an APL program named CAB. CAB is an acronym of "Canonical Analysis with Backshifts". A listing of the program can be found in the Appendix.

6.9 ARMA class

Until now, we have largely dealt with the canonical class. In the present and the next section we glance at two other classes: the ARMA class and the state space class.

The ARMA class is defined by the loss function

$$\sigma(\mathbf{Z}; \mathbf{X}; \mathbf{A}_p; \mathbf{F}_q) = \text{ssq} \left(\sum_{q=0}^{Q} \mathbf{B}_q\mathbf{Z}\mathbf{F}_q, \sum_{p=0}^{P} \mathbf{B}_p\mathbf{X}\mathbf{A}_p \right) \qquad (6.46)$$

where $\mathbf{A}_0 = \mathbf{I}$ and $\mathbf{F}_0 = \mathbf{I}$. Equation (6.46) is a special case of (6.4) with K=1. It is equivalent to loss function (2.31) for the multivariate ARMA(P,Q) model.

There is one particular annoying thing about loss function (6.46): it will become zero if we choose $\mathbf{Z} = \mathbf{X}$, a solution that has little to do with data

analysis. This problem was of course also realized by Box and Jenkins and they assumed that **Z** was generated by a white noise process. This means that if we have an infinite amount of observations then

$$\mathbf{Z}'\mathbf{B}_q\mathbf{Z} = \mathbf{0} \qquad q > 0 \tag{6.47}$$

$$\mathbf{Z}'\mathbf{B}_p\mathbf{X} = \mathbf{0} \qquad p = 1 \ldots P \tag{6.48}$$

will be true. The first condition states that **Z** is not autocorrelated, the second one states that **Z** is uncorrelated with the first P lags of **X**. Since we have only a sample of N observations, one thing we can do is require that something like

$$\mathbf{Z}'\mathbf{B}_q\mathbf{Z} = \mathbf{0} \qquad q = 1 \ldots Q \tag{6.49}$$

$$\mathbf{Z}'\mathbf{B}_p\mathbf{X} = \mathbf{0} \qquad p = 1 \ldots P \tag{6.50}$$

holds in the sample. It is however not entirely clear how it is possible to find a **Z** that not only minimizes (6.46) but also satisfies (6.49) and (6.50). It seems that we are going into the wrong direction.

Alternatively, following the approach of Box and Jenkins (1976, Ch. 7) it is possible to define a sums of squares loss function. Given a set of parameters $\mathbf{A}_p$ and $\mathbf{F}_q$ it is possible to compute the residual

$$\mathbf{Z} = \sum_{p=0}^{P} \mathbf{B}_p\mathbf{X}\mathbf{A}_p - \sum_{q=1}^{Q} \mathbf{B}_q\mathbf{Z}\mathbf{F}_q, \tag{6.52}$$

in a recursive manner. The goal is then to find that set of parameters that minimize the sum of squared residuals, tr $\mathbf{Z}'\mathbf{Z}$. Using this approach, there is no assurance that **Z** will approximate white noise. The amount of autocorrelation of **Z** can only be assessed after a model has been fitted.

Generalizing the ARMA model to an optimal scaling technique is not easy. Using a priori white noise constraints on **Z** is not a viable alternative. What remains is to minimize the recursive residual (6.52), but since this residual is also part of the approximation structure, it is difficult to

separate the structural component from the error component. Consequently, many problems remain.

6.10 State space class

We obtain the state space class from (6.4) if we set K = 2, $P_1 = P_2 = Q_2 = 0$ and $Q_1 = 1$. The loss function then becomes

$$\sigma(\mathbf{Y}; \mathbf{Z}; \mathbf{F}_1; \mathbf{F}_2; \mathbf{A}_1) = \text{ssq}\,(\mathbf{Z}, (\mathbf{X}\mathbf{A}_1 - \mathbf{B}_1\mathbf{Z}\mathbf{F}_1)) + \text{ssq}\,(\mathbf{X}_2, \mathbf{Z}\mathbf{F}_2). \tag{6.53}$$

The columns of $\mathbf{X}_2 = \mathbf{X}\mathbf{A}_2$ form a subset of the columns of $\mathbf{X}$ obtained by multiplication with a known matrix $\mathbf{A}_2$ with zero row constraints. Equation (6.53) measures the loss between the data and the expected value structure of the state space model. In matrix notation, this structure is defined by two equations:

$$\mathbf{Z} = -\mathbf{B}_1\mathbf{Z}\mathbf{F}_1 + \mathbf{X}\mathbf{A}_1 \tag{6.54}$$

$$\mathbf{X}_2 = \mathbf{Z}\mathbf{F}_2 \tag{6.55}$$

Minimizing (6.53) over $\mathbf{X}$, $\mathbf{Z}$, $\mathbf{F}_1$, $\mathbf{F}_2$ and $\mathbf{A}_1$ under constraint $\mathbf{Z}'\mathbf{Z} = \mathbf{I}$ is the DYNAMALS problem studied by De Leeuw and Bijleveld (1988). In their notation $\mathbf{F}' = -\mathbf{F}_1$, $\mathbf{G}' = \mathbf{A}_1$, $\mathbf{H}' = \mathbf{F}_2$ and $\mathbf{Y} = \mathbf{X}_2$. Actually, the DYNAMALS problem is slightly more general since the first part of (6.53) is weighted by a constant $\omega^2 \geq 0$ which indicates the relative importance of the system equation (6.54) in the minimization problem.

The minimization problem has been solved by De Leeuw and Bijleveld (1988) and we will not repeat their solution here. Bijleveld (1989) gives many interesting examples of how linear dynamic systems analysis can be applied to behavioral data.

6.11 Summary

It has been a long journey for those who read the entire chapter. Since not everyone will appreciate the forest of technical details, we give a short verbal summary.

The chapter commences by introducing a general loss function. This function can be molded into three classes of minimization problems: the canonical class, the ARMA class and the state space class. The canonical class is highly important in this work. Section 6.3 shows that the techniques discussed in Chapters 4 and 5 are special cases of this class. Next, the minimization problem of the canonical class is treated at length. The minimization strategy includes majorization over two sets of parameters. We discuss two algorithms: one for general design matrices, and one for backshift matrices only. The latter algorithm is much more efficient. The chapter concludes with short notes on the ARMA class and on the state space class. The minimization problem of the state space class has been solved before, for the ARMA class however, no solutions are available yet.

CHAPTER 7

Conclusion

All time series are finite, and so are all books. This last chapter reconsiders the problem of the study and summarizes the main results. There are also things in life that are never finished. Scientific quest is one of them. We therefore conclude with some suggestions for further research.

7.1 A retrospect on the problem

The problem we set out in Chapter 1 is:

In what way can we integrate time series analysis and optimal scaling ?

We have concentrated on constructing optimal scaling equivalents for a number of time series methods. These include intervention analysis, autoregressive models, dynamic components analysis and predictable components analysis. We formulated these methods as least squares optimal scaling problems with lagged variables.

The loss function for the canonical class and its associated minimization procedures given in Chapter 6 provide the most direct and outright answer to the central question. We have shown how many time series techniques can be derived as special cases of the canonical class. Other methods for modelling dependency among observations can be handled in the same manner. Since Chapter 6 furnishes the double majorization procedure, the minimization problem is solved as soon as we recognize the problem as a case of the canonical class (remember that **B** may be any fixed matrix in the first algorithm). By applying the Y-step majorization results, the optimal scaling problem is also automatically solved then.

The work has by no means ended. For example, we have been very concise on model identification, the stability of the solution, convergence properties of the algorithm and on the comparison with loglinear analysis. However this work provides a convenient starting point for further research on these topics. With this in mind we feel that the text renders a satisfactory answer to the problem of the study.

The sections below summarize the main results and give some indications for future research.

7.2 Main results

The correlation box proposed in Chapter 2 is a practical organization of different types of autocorrelations. It has little to do with optimal scaling as such, but it provides a proper means to unify cross-sectional multivariate methods and time series analysis. The relationship with least squares problems outlined in Section 2.5 is a first step into this direction. The idea here is to generalize multivariate techniques for the two dimensional correlation matrix to the three dimensional correlation box. The box has also an educational value, since it organizes autocorrelations in such a way that useful slices can be made without much difficulty.

Matrix algebra provides an alternative way to formulate time series models. The majority of the time series literature relies on extensive use of t subscripts and lag operators. The advantage of matrix notation is not only that the t subscripts disappear, leading to simpler and more compact formulas, but also that it will become easier to apply results from linear algebra and projective geometry.

Chapter 3 presents the Gifi system of optimal scaling based on restricted cases of homogeneity analysis. Some sections contain novel results. These include the section on the equality restriction on the quantifications, the section on linear restrictions on the object scores, and both sections on the component loadings. Writing $\mathbf{X} = \mathbf{GY}$ for the quantified data matrix is convenient for generalizing linear models. We use it a lot in later chapters.

In Chapter 4 we found that optimal scaling changes the autocorrelation of the series. For example, the lag-1 predictor model either maximizes or minimizes the first order autocorrelation of the quantified series. For multiple autoregression, the optimal quantification of the dominant, most influential lag was used for all P lags. We do not know if this is a systematic feature. If it is, then the optimal scaling problem can probably be simplified to finding the transformation of the dominant lag only, instead of solving the problem under equality constraints. This would

also alleviate the model dependency problem of the optimal transformation.

Another interesting result concerns the output of the exponential smoothing filter. This filter outperforms the direct, recursive filter in two respects. First, the output series better represents the input series, i.e. it has higher correlation with the input series. Second, it is smoother since its autocorrelation is higher. Again, more research is needed to establish these properties for the general case, but the results obtained thus far are promising. Besides time series analysis, one alternative application of the filter is to use it for the optimal transformation itself, i.e. to smooth the transformation function. This use is analogous to that of spline functions in data analysis (cf. Winsberg & Ramsay, 1980, 1983).

The initial goal of the research was to generalize Immink's dynamic factor analysis (Immink, 1986) to categorical data. While the methods differ substantially, it is gratifying to see that the factors from Immink's analysis and our analysis are virtually identical (the correlation between the factors is equal to 0.97). Both methods seem to extract the same type of information.

The use of sets enables us to analyze replicated time series. An example is the multiset dynamic components technique. Another use is to partial out a specific combination of series (cf. Gifi, 1981, 238). We have not explored this possibility in a time series context, but if an interesting application exists, it can be done.

Finally, the major technical result is the double majorization procedure in Chapter 6. This procedure is capable of handling most optimal scaling problems of the Gifi system as well as many time series problems. We hope that the procedure and its associated loss function may have a fertilizing effect on both fields.

7.3 Suggestions for further research

Because time series analysis covers a huge domain of methods and techniques our work may be extended in a multitude of ways. Below we list some possibilities.

There is no adequate and proven model selection strategy yet. Probably the most significant aspects of the work by Box and Jenkins (1976) are the model identification tools. It may be worthwhile to draw upon their classic research in order to develop something similar for our methods. This requires rather stringent distributional assumptions on behalf of the series. The development of the statistical theory of optimal scaling methods is still in its infancy, so this research would not be particularly easy.

A related point concerns the stability of the least squares estimates. It is well known that in the presence of autocorrelation least squares estimates are unbiased, but that they do not have minimum variance. Moreover multicollinearity in higher order autoregressive models may add to the variability of the estimates. In principle, it is possible to devise equivalent GLS or even Maximum Likelihood procedures, but this probably requires a formidable effort while the direct gain may be small. A more promising way appears to be given by resampling methods like the Jackknife and the Bootstrap (Efron, 1979; Efron & Gong, 1983). These methods have already been applied to forms of optimal scaling (e.g. Meulman, 1982; Van der Burg, 1988) so at least some work into this direction is available.

Econometrists may be interested in constraining the contribution of successive lags. Popular candidates are the Almon distributed lag model and generalizations thereof like the spline lag and the Shiller lag models. See Judge et al. (1985) for an overview. For most of these lag models least squares procedures are known, so incorporating them into the general loss function seems feasible. Also, linear constraints on the loadings (cf. Section 3.12) can be integrated without much difficulty.

The double majorization procedure remains exactly the same under an arbitrary change of the **B** matrices. This opens up possibilities for analyzing spatial data. Suppose that we have data arranged on a two-dimensional spatial grid and that we are interested in studying mutual influences into the directions North, South, West and East. Each direction defines a **B** matrix by a proper allocation of zeroes and ones. In principle, the influence of a direction may be assessed by studying the corresponding loadings. Another potential application area could be in analyzing the flow in social network data, in which the **B** matrix contains the links between the individual group members.

For the moment, we need more practical experience with the techniques. Some typical questions are: How do they compare with other methods of categorical time series analysis like Markov chain models, DYNAMALS and qualitative regression analysis ? When to apply which method ? Are there any conditions under which the techniques do not work ?

The results obtained thus far are encouraging, but undoubtedly I will have fallen into traps here and there. This need not be alarming: after all, the techniques are still in their infancy. I hope others will continue to aid in signalling growing pains.

Appendix

This appendix contains the APL program that was used for all computations. APL is not a particularly fast computer language, but is a nifty tool for prototyping because it is interactive and especially because it contains many built-in matrix operations.

The program uses the standard APL syntax with one exception. We use the diamond operator ⋄ to allow for more statements on one program line. For example, program line [14] in CAB is

```
[14] IT←IT+1 ⋄ OLS←LS
```

This statement is equivalent to

```
[14A] IT←IT+1
[14B] OLS←LS
```

The remainder of the syntax adheres to Iverson's original definition.

The main program is called CAB, which stands for Canonical Analysis with Backshifts. Its structure is comparable to the second algorithm derived in Chapter 6.

Some important scalars in the program are

NO←(⍴DATA)[1]	The number of time points: N
NS←(⍴MODELX)[1]	The number of sets: K
NV←(⍴MODELX)[3]	The number of variables: M
ND←(⍴MODELZ)[3]	The number of dimensions: R
NLX←(⍴MODELX)[2]	One plus the maximum number of lags for X: P
NLZ←(⍴MODELZ)[2]	One plus the maximum number of lags for Z: Q

The program is called with one argument, the data matrix. This is not the only input the porgram requires: a set of global variables is used to define the precise type of the analysis. These globals and their function are listed below.

The input globals are:

MODELZ	
MODELX	
LEVELS	
CONTIN	
FILTER	
DATA	
A	(optional)
F	(optional)

- MODELX
  This is a binary three dimensional array with ⍴MODELX = NS NLX NV.
  This array tells the program
  - which lags of X to include in the analysis
  - the values for NS, NLX and NV
  For example:

```
    MODELX
1 1 1
0 0 0

0 0 0
1 1 1
```

  specifies that NS = 2 The number of sets
  NLX = 2 One plus the maximum number of lags for X
  NV = 3 The number of variables
  and that set 1 contains three unlagged variables
  set 2 contains three variables at lag 1

- MODELZ
  This is also a binary three dimensional array with ⍴MODELZ = NS NLZ ND
  This array tells the program
  - which lags of Z to include in the analysis
  - the values of NLZ and ND
  For example:

```
    MODELZ
1

1
```

  specifies that NLZ = 1 One plus the maximum number of lags for Z
  ND = 1 The number of dimensions
  and that both sets contain one unlagged Z

- LEVELS
  This is an array of length NV
  It tells the program
  - the measurement level of each series
  For example:

```
    LEVELS
2 1 3
```

  specifies that series 1 has an ordinal measurement level (2)
  series 2 has a nominal measurement level (1)
  series 3 has an interval measurement level (3)

- CONTIN
  This is a binary array of length NV
  It tells the program
  - whether a series must be treated as continuous (1) or as categorical (0)
  For example:

```
     CONTIN
0 0 1
```

  specifies that series 1 and 2 are categorical
  series 3 is continuous

  Note: the continuous flag can only be set for series with an interval level
  the program does not check this, so be careful

- FILTER
  This is a boolean.
  It tells the program
  - if arrays A and F must be treated as fixed filtering weights (1 = yes, 0 = no)
  For example:

```
     FILTER
0
```

  specifies not to use input from A and F.
  Note: If filtering is turned on then arrays A and F should contain valid filter coefficients that the program will use.
  Additionally, the variance of Y will not be standardized

- DATA
  This is the data matrix with ρDATA = NO NV
  It tells the program
  - what the data values are
  - how many observations to analyze
  For example:

```
      DATA
1 1 2
1 1 3
3 3 2
3 3 3
3 2 1
2 1 1
1 1 1
1 2 2
1 1 1
2 1 1
2 1 2
2 2 1
2 3 3
```

```
 3 3 3
 3 3 2
 3 3 3
 3 3 2
 3 3 1
 1 2 1
 2 3 1
```

specifies that NO = 20 The number of time points

Note: DATA is the argument of the CAB function. It is local to the function. Therefore the data array may have any unreserved name.

Optional input can be specified in arrays A and F (if filtering is on). These arrays are discussed in the output section. This completes the input for the program.

Below, we invoke the program and discuss its output.

We start the program with

```
     CAB DATA
INITIALIZING...
     .728566
     .449807
     .421980
     .418612
     .418125
PRELIMINARY SSQ: 0.1951

                X          A F          Z

IT   1     .20195137    .20140742    .16554010
IT   2     .16525215    .16143900    .14421554
IT   3     .14402437    .14169072    .13236226
IT   4     .13222202    .13068793    .12504436
IT   5     .12493328    .12386867    .12014566
IT   6     .12005289    .11928265    .11666917
IT   7     .11658889    .11601442    .11410029
IT   8     .11402911    .11359153    .11210533
IT   9     .11208545    .11174748    .11064153
IT  10     .11058274    .11031937    .10946122
IT  11     .10940684    .10920046    .10852978
IT  12     .10847903    .10831677    .10779029
IT  13     .10774253    .10761472    .10720033
IT  14     .10715508    .10705431    .10672766
   .10660505
```

and the program will produce various types of output.

First the message

```
INITIALIZING...
```

appears. This is followed by a table of loss values of the initial homogeneity analysis. Then

```
PRELIMINARY SSQ: 0.1951
```

is the value of loss function (6.5) for the initial configuration. This value can be lower than the ones given later since no measurement level restrictions are placed on the variables. This is also the case in the above example.

Subsequently, the program prints three columns of loss values. The first column corresponds to the loss values computed after the Y-step (the column is labelled X), the second is computed after the A/F step, and the third one after the Z-step. The loss values will always become lower after more steps have been carried out.
The final loss after rotation is equal to

```
   .10660505
```

and the program stops.

The CAB program defines various global output arrays that contain the analysis results.

The output globals are:

A	Component loadings
D	Marginal frequencies
F	System parameters
G	Super indicator matrix
P	Residuals
X	Quantified data matrix
Y	Quantification matrix
Z	Object scores

- A (NS, NLX × NV, ND) **Component loadings**

  For the above example:

```
     ⍴A
2 6 1
     A
   1.093
  ¯0.1563
   0.05033
   0
   0
   0
```

```
   0
   0
   0
   0.2669
   0.1943
  ¯0.7407
```

The entries for which MODELX is zero are all zero.

```
•    D     (total of number of categories)        Marginal frequencies
     ρD
7
     D
6 6 8 7 4 9 ¯6.939E¯17
```

The entry for the last series is zero since this series is continuous

```
•    F     (NS, ND × NLZ, ND)                     System parameters
     ρF
2 1 1
     F
1

1
```

In the canonical class F(0) is always unity.

```
•    G     (NO, NU)                               Super indicator matrix
      ρG
20 7
      G
   1        0        0        1        0        0        0.055
   1        0        0        1        0        0        0.33
   0        0        1        0        0        1        0.055
   0        0        1        0        0        1        0.33
   0        0        1        0        1        0       ¯0.22
   0        1        0        1        0        0       ¯0.22
   1        0        0        1        0        0       ¯0.22
   1        0        0        0        1        0        0.055
   1        0        0        1        0        0       ¯0.22
   0        1        0        1        0        0       ¯0.22
   0        1        0        1        0        0        0.055
   0        1        0        0        1        0       ¯0.22
   0        1        0        0        0        1        0.33
   0        0        1        0        0        1        0.33
   0        0        1        0        0        1        0.055
   0        0        1        0        0        1        0.33
   0        0        1        0        0        1        0.055
   0        0        1        0        0        1       ¯0.22
   1        0        0        0        1        0       ¯0.22
   0        1        0        0        0        1       ¯0.22
```

The last column of G is corresponds to the continuous series.

```
•   P    (NS, NO, ND)                    Residuals
     ⍴P
2 20 1
    +/+/+/P*2                            The total loss value
0.1066
     +/+/P[1;;]*2                        Loss for set no. 1
0.05263
     +/+/P[2;;]*2                        Loss for set no. 2
0.05397
    +/+/P[;10;]*2                        Loss for time point no. 10
0.006068
```

```
•   X    (NO, NU)                        Quantified data matrix

     ⍴X
20 3
     X
   0.1826      0.2689      0.05505
   0.1826      0.2689      0.3303
  ¯0.2739     ¯0.2318      0.05505
  ¯0.2739     ¯0.2318      0.3303
  ¯0.2739      0.05082    ¯0.2202
   0.1826      0.2689     ¯0.2202
   0.1826      0.2689     ¯0.2202
   0.1826      0.05082     0.05505
   0.1826      0.2689     ¯0.2202
   0.1826      0.2689     ¯0.2202
   0.1826      0.2689      0.05505
   0.1826      0.05082    ¯0.2202
   0.1826     ¯0.2318      0.3303
  ¯0.2739     ¯0.2318      0.3303
  ¯0.2739     ¯0.2318      0.05505
  ¯0.2739     ¯0.2318      0.3303
  ¯0.2739     ¯0.2318      0.05505
  ¯0.2739     ¯0.2318     ¯0.2202
   0.1826      0.05082    ¯0.2202
   0.1826     ¯0.2318     ¯0.2202
```

```
•    Y    (total number of categories, NU)   Quantification matrix
      ⍴Y
7 3
      Y
   0.1826      0           0
   0.1826      0           0
  ¯0.2739      0           0
   0           0.2689      0
   0           0.05082     0
   0          ¯0.2318      0
   0           0           1
```

```
•    Z    (NO, ND)                          Object scores, latent vectors

     ⍴Z
20 1
     ,Z
0.09026 0.1311 ¯0.2064 ¯0.2115 ¯0.3517 0.1384 0.224 0.2465 0.0919
0.224 0.2308 0.1332 0.2479 ¯0.2579 ¯0.3247 ¯0.2115 ¯0.3247 ¯0.225
0.1213 0.2343
```

This concludes the output section. The entire example analysis takes about 50 seconds on an Apple Macintosh II running APL.68000 V5.0.

The program is listed below. Some routines are accompanied by a short comment. They are not examples of neat and efficient APL programming. Much of them could be rewritten into a more APL-like style. A floppy disk can be obtained from the author. The functions that are not credited may be freely used for non-commercial purposes.

```
      ∇CAB[⎕]∇
      ∇CAB DATA;IT;OLS;LS;LS1;LS2;LS3;LS4;NO;NU;ND;NS;NZ;CAT;CCAT;C;
      NC;L1;L2;PZ;NC;UECALPHA;UECGAMMA;EUA;EUC;SUDP;SUDPHI;SUDQ;W;
      RANK;NLX;NLZ;CRIT;MAXIT;DJ;AT;FF;LABGG;MK;PK;SPECIAL;SPECA;SK;
      SINGUEC;UK;UECLAB;FA
[1]   ⍝
[2]   ⍝ CAB: CANONICAL ANALYSIS WITH BACKSHIFT MATRICES
[3]   ⍝ ESTIMATES X, Y, Z, F AND A
[4]   ⍝ SEE: STEF VAN BUUREN: OPTIMAL SCALING OF TIME SERIES (1990)
[5]   ⍝ INPUT: DATA, MODELX, MODELZ, LEVELS, CONTIN, FILTER,
[6]   ⍝        F(OPT), A(OPT)
[7]   ⍝ OUTPUT: G, X, Y, Z, F, A, D, P
[8]   ⍝
[9]    CABINI2 DATA ⋄ ' PRELIMINARY SSQ: ',⍕LS←CABLOSS ⋄ IT←0
[10]  ' ' ⋄ '              X          A F          Z' ⋄ ' '
[11]  ⍝
[12]  ⍝                    MAIN LOOP
[13]  ⍝
[14]  ITERATE:IT←IT+1 ⋄ OLS←LS
[15]   ⎕←'IT ',(3 0⍕IT)
[16]   X←CABMAX3 X ⋄ ⎕←12 8⍕LS1←CABLOSS
```

```
[17]  →(FILTER)/NOFA ◊ FA←CABMAF FA ◊ ⎕←12 8⍕LS2←CABLOSS
[18] NOFA:Z←CABMAZ2 Z ◊ ⎕←12 8⍕LS3←CABLOSS
[19]  ⎕←□R ◊ LS←LS3
[20]  →((MAXIT≥IT)^(CRIT≤|LS-OLS))/ITERATE
[21] ⍝
[22] ⍝                  END OF MAIN LOOP
[23] ⍝
[24]  FINROT ◊ ⎕←12 8⍕LS←CABLOSS
      ∇
```

CAB contains the main iteration loop. After initializing with CABINI2 the program iterates over CABMAX3 (Y-step), CABMAF (F/A step) and CABMAZ2 (Z-step). At the end FINROT takes care of the final rotation.

```
      ∇CABINI2[⎕]
      ∇CABINI2 DATA;KJ;J;R;IDX;T;GJ;S;K;UNO;MASK;YJ;C1;LAG;GL;HOMG;
          HOMY;HOMCAT;HOMCCAT;HOMCON;SUPERG;INITCRIT;L;OLDL;DJ;NYJ
[1]    ' INITIALIZING...'
[2]   ⍝ ---- CONVERGENCE CRITERIA -----
[3]    CRIT←0.0005 ◊ MAXIT←200 ◊ INITCRIT←0.001
[4]   ⍝ ---- BOUNDERIES ------
[5]    NO←(⍴DATA)[1] ◊ NS←(⍴MODELX)[1]
[6]    NV←(⍴MODELX)[3] ◊ ND←(⍴MODELZ)[3]
[7]    NLX←(⍴MODELX)[2] ◊ NLZ←(⍴MODELZ)[2]
[8]   ⍝ ---- SHAPE DATA IN PROPER FORM ------
[9]    G←INDICATOR DATA ◊ J←1
[10] VARLOOP1:→(CONTIN[J]=0)/NEXTVAR1
[11] CONT1:G[;CCAT[J]]←(÷NO*0.5)×STAND(NO,1)⍴G[;CCAT[J]]
[12] NEXTVAR1:→(NV≥J←J+1)/VARLOOP1
[13]  NC←(⍴G)[2] ◊ D←+⌿G
[14] ⍝ ---- COMPUTE C (NEEDED FOR SPEEDING UP CABMAX3 AND CABMAF)
[15]  C←(NLX,NO,NC)⍴0 ◊ S←1
[16] LAGLOOP2:C[S;;]←(1⍴S-1)LAGOF G ◊ →(NLX≥S←S+1)/LAGLOOP2
[17] ⍝ ---- INITIALIZATION OF CONTINUOUS VARIABLES -----
[18]  Y←(NC,NV)⍴0 ◊ J←1
[19] V2:→(CONTIN[J]=0)/NXTV ◊ Y[CCAT[J];J]←1
[20] NXTV:→(NV≥J←J+1)/V2
[21] ⍝ ---- INITIALIZE Z
[22]  Z←GRAM DEVMN(NO,ND)⍴NO?99999
[23] ⍝
[24] ⍝ ---- HOMALS TO OBTAIN INITIAL QUANTIFICATION MATRIX Y
[25] ⍝
[26] ⍝ + +  INLCUDE ALL LAGS IN THE ANALYSIS
[27]  K←1 ◊ HOMCAT←CAT ◊ HOMCON←CONTIN
[28]  SUPERG←(⍳NLX-1)LAGOF G ◊ HOMG←G
[29] SG:MASK←(((NLX-1)×⍴CAT)⍴CAT)/NV↓,MODELX[K;;]
[30]  HOMG←HOMG,MASK/SUPERG
[31]  HOMCAT←HOMCAT,(((NLX-1)×⍴CAT)⍴CAT)×NV↓,MODELX[K;;]
[32]  HOMCON←HOMCON,(((NLX-1)×⍴CONTIN)⍴CONTIN)×NV↓,MODELX[K;;]
[33]  →(NS≥K←K+1)/SG
[34]  HOMCON←(HOMCAT≥1)/HOMCON
```

```
[35]  HOMCCAT←+\HOMCAT←(HOMCAT≥1)/HOMCAT
[36]  HOMY←((⍴HOMG)[2],ND)⍴0
[37] ⍝ + +  NEXT ITERATE: BEGIN OF INITIALIZATION LOOP (INIHOM)
[38]  L←1000
[39] INIHOM:OLDL←L ⋄ J←1
[40] ⍝ + +  RECIPROCAL AVERAGING
[41] CNT:IDX←(HOMCCAT[J]-KJ)+⍳KJ←HOMCAT[J]
[42]  →(HOMCON[J]=0)/CNT1 ⋄ HOMY[IDX;]←Z⌸(NO,1)⍴HOMG[;IDX] ⋄ →CLOOP
[43] CNT1:GJ←(NO,KJ)⍴HOMG[;IDX]
[44]  HOMY[IDX;]←((⍉GJ)+.×Z)÷⍉(ND,(⍴GJ)[2])⍴+⌿GJ
[45] CLOOP:→((⍴HOMCAT)≥J←J+1)/CNT
[46]  Z←GRAM DEVMN HOMG+.×HOMY
[47] ⍝ + +  COMPUTE LOSS L AND RELOOP IF NEEDED
[48]  J←1 ⋄ L←0
[49] VARLOOP:IDX←(HOMCCAT[J]-KJ)+⍳KJ←HOMCAT[J]
[50]  GJ←(NO,KJ)⍴HOMG[;IDX] ⋄ YJ←(KJ,ND)⍴HOMY[IDX;]
[51]  L←L++/+/(Z-GJ+.×YJ)*2
[52]  →((⍴HOMCAT)≥J←J+1)/VARLOOP
[53]  ⎕←'      ',6⍕L←L÷⍴HOMCAT
[54]  →(INITCRIT≤|OLDL-L)/INIHOM
[55] ⍝ -----  END OF VNIDIMENSIONAL HOMALS LOOP
[56] ⍝
[57] ⍝ -----  WE HAVE A GOOD INITIAL Y NOW, FILL IT IN THE RIGHT SPOT
[58]  J←1
[59] VARLOOP2:→(HOMCON[J]=1)/NEXTVAR2
[60]  DJ←D[IDX←(CCAT[J]-KJ)+⍳KJ←CAT[J]]
[61]  NYJ←NYJ-(+/(NYJ←HOMY[IDX;1])×DJ)÷NO
[62]  →(FILTER)/L3 ⋄ NYJ←NYJ÷(+/DJ×NYJ×NYJ)*0.5
[63] L3:Y[IDX;J]←(KJ,1)⍴NYJ
[64] NEXTVAR2:→(NV≥J←J+1)/VARLOOP2
[65] ⍝ -----  CONSTRUCT INITIAL QUANTIFIED DATA MATRIX X
[66]  X←G+.×Y
[67] ⍝ -----  INITIALIZE F AND A
[68]  →(~FILTER)/INITFA ⋄ FA←F,[2]A ⋄ →DONE
[69] INITFA:FA←CABMAF(NS,((NLZ×ND)+(NLX×NV)),ND)⍴0 ⋄ K←1
[70] SETLOOP3:FA[K;(⍳ND);]←F[K;(⍳ND);]←UNIT ND ⋄ →(NS≥K←K+1)/SETLOOP3
[71] DONE:
[72] ⍝ ---- INITIALIZE AUXILARY MAJORIZATION GLOBALS
[73]  VECALPHA←STAND((NC×NV),1)⍴(NC×NV)↑,Z
[74]  VECGAMMA←STAND((NO×ND),1)⍴(NO×ND)↑,X
[75]  SPECIAL←(C1,LAG LAG0 2×C1←(NLZ,1)⍴(NO-LAG),(LAG←NLZ-1)⍴1)
KRONEK(ND,ND)⍴1
[76]  SPECA←(C1,LAG LAG0 2×C1←(NLX,1)⍴(NO-LAG),(LAG←NLX-1)⍴1)
KRONEK(NV,NV)⍴1
[77]  VECLAB←STAND(NC,1)⍴NC↑,Z
[78]  LABGG←VECLAB MAXSIVAL(⍉G)+.×G
```

This function sets up all arrays for the analysis. Steps `[23]` to `[55]` are the initial homogeneity iterations. In `[75]` array `SPECIAL` is the design matrix for the majorization over `Z` (named Tq in the text), in `[76]` array `SPECA` is the corresponding array for the `Y`-step.

```
      ▽CABLOSS[□]▽
      ▽LS←CABLOSS;K
[1]   ⍝ COMPUTATION OF THE LOSS VALUE OF
[2]   ⍝      SUM[      SSQ[ SB Z F)   - SUM (C Y A)]]
[3]   ⍝ NEEDED: Z, Y, A, F, NS
[4]    LS←+/+/+/(P←CABRES)*2
      ▽
```

Computes the value of (6.5).

```
      ▽CABMAF[□]▽
      ▽FA←CABMAF FA;S;R;K;MASK
[1]   ⍝ ESTIMATION OF LOADINGS A AND WEIGHTS F
[2]   ⍝ NEEDED:C, X, Z, FA
[3]    S←1 ⋄ R←Z,(⍳NLZ-1)LAGOF-Z
[4]   LAGLOOP:R←R,C[S;;]+.×Y ⋄ →(NLX≥S←S+1)/LAGLOOP
[5]    K←1
[6]   SETLOOP:MASK←(,MODELZ[K;;]),,MODELX[K;;] ⋄ MASK[⍳ND]←0
[7]    FA[K;((+/MASK)↑⍒MASK);]←Z⌹MASK/R
[8]    →(NS≥K←K+1)/SETLOOP
[9]    F←(NS,(ND×NLZ),ND)↑FA
[10]   A←(0,(ND×NLZ),0)↓FA
      ▽


      ▽CABMAX3[□]▽
      ▽X←CABMAX3 X;K;S;NOND;NCNV;R;W;AS;U;YJ;UJ;IDX;KJ;GJ;NYJ;J;Q;
      LEVEL;FF1;FF2;LAG;UPPER;TRAK;TRAAK;AK;NONV
[1]   ⍝ ESTIMATION OF THE CATEGORY QUANTIFICATIONS Y IN X=GY
[2]   ⍝ NEEDED: A, C, Z, Y, P
[3]    →(×/CONTIN)/0
[4]    NOND←NO×ND ⋄ NCNV←NC×NV ⋄ NONV←NO×NV ⋄ Q←U←(NC,NV)⍴0
[5]    FF1←FF2←((NV×NLX),(NV×NLX))⍴0
[6]    K←1 ⋄ UPPER←0
[7]   ⍝
[8]   ⍝ ---- DETERMINE THE UPDATE MATRIX U OVER ALL SETS
[9]   ⍝
[10] SETLOOP:
[11]   PK←P[K;;] ⋄ AK←A[K;;] ⋄ AT←(0,NV)⍴0
[12]   TRAK←+/(+/AK*2)×((NLX⍴NV)/NO+1-⍳NLX)
[13]   S←1
[14] ⍝ - CONSTRUCT SPECIAL FF MATRICES
[15] LAGLOOP:LAG←S-1
[16]   AS←((LAG×NV),0)↓((S×NV),NV)↑AK ⋄ AT←AT,[1]⍉AS
[17]   FF1[(⍳NV×NLX-LAG);(NV×LAG)+⍳NV]←((NV×NLX-LAG),NV)⍴AS
[18]   FF2[((NV×LAG)+⍳NV);⍳NV×NLX-LAG]←(NV,(-NV×NLX-LAG))↑⍉AK
[19] ⍝ - ACCUMULATE THE UPDATE MATRICES IN Q
[20]   Q←Q+(⍉C[S;;])+.×PK+.×⍉(NV,ND)↑AS
[21]   →(NLX≥S←S+1)/LAGLOOP
[22] ⍝ - THE HADAMARD PRODUCT
[23]   TRAAK←+/+/SPECA×FF×FF←FF1+.×FF2
```

```
[24]  MK←TRAK÷NONV ◊ SK←((TRAAK÷NONV)-MK*2)*0.5
[25]  UK←MK+SK×(NONV-1)*0.5
[26]  UPPER←UPPER+UK
[27]  →(NS≥K←K+1)/SETLOOP
[28] ⍝ - AND FINALLY... THE UPDATE MATRIX U
[29]  U←Q÷(UPPER×LABGG)
[30] ⍝
[31] ⍝ --- RESTRICT AND SCALE THE NEW CATEGORY VECTORS YJ+UJ
[32] ⍝
[33]  J←1
[34] VARLOOP:→(1=CONTIN[J])/NEXTVAR
[35]  LEVEL←LEVELS[J]
[36]  IDX←(CCAT[J]-KJ)+⍳KJ←CAT[J] ◊ DJ←D[IDX]
[37]  YJ←Y[IDX;J] ◊ UJ←U[IDX;J]
[38]  →(LEVEL=1)/NOM ◊ →(LEVEL=2)/ORD ◊ →(LEVEL=3)/NUM
[39] NOM:NYJ←YJ+UJ ◊ →SCALE
[40] ORD:NYJ←⌽MANOR⌽(YJ+UJ) ◊ →(~×/NYJ[1]=NYJ)/SCALE
[41]  NYJ←⌽MANOR⌽(-YJ+UJ) ◊ →SCALE
[42] NUM:NYJ←LIDICA YJ+UJ
[43] SCALE:NYJ←NYJ-(+/NYJ×DJ)÷NO
[44]  →(FILTER)/L1 ◊ NYJ←NYJ÷(+/DJ×NYJ×NYJ)*0.5
[45] L1:Y[IDX;J]←(KJ,1)⍴NYJ
[46] NEXTVAR:→(NV≥J←J+1)/VARLOOP
[47]  X←G+.×Y
      ∇


      ∇CABMAZ2[⎕]∇
      ∇Z←CABMAZ2 Z;NOND;K;S;UPPER;UK;LAG;SK;MK;FS;U;CY;X;TRAK;FK;PK;
      FT;FF1;FF2;LAG;TRAAK;AK
[1]  ⍝ ESTIMATION OF THE LATENT SCORES Z BY MAJORIZATION
[2]  ⍝ NEEDED: F     , Z, P, SPECIAL
[3]   →(NLZ=1)/NOLAGS
[4]   NOND←NO×ND ◊ U←(NO,ND)⍴0
[5]   FF1←FF2←((ND×NLZ),(ND×NLZ))⍴0
[6]   K←1 ◊ UPPER←0
[7]  ⍝
[8]  ⍝ ---- DETERMINE THE UPDATE MATRIX U OVER ALL SETS
[9]  ⍝
[10] SETLOOP:
[11]  PK←-P[K;;] ◊ FK←F[K;;] ◊ FT←(0,ND)⍴0
[12]  TRAK←+/(+/FK*2)×((NLZ⍴ND)/NO+1-⍳NLZ)
[13] ⍝ CONSTRUCT SPECIAL FF MATRICES
[14]  S←1
[15] LAGLOOP:LAG←S-1
[16]  FS←((LAG×ND),0)↓((S×ND),ND)↑FK ◊ FT←FT,[1]⍪FS
[17]  FF1[(⍳ND×NLZ-LAG);(ND×LAG)+⍳ND]←((ND×NLZ-LAG),ND)⍴FS
[18]  FF2[((ND×LAG)+⍳ND);⍳ND×NLZ-LAG]←(ND,(-ND×NLZ-LAG))↑⍪FK
[19]  →(NLZ≥S←S+1)/LAGLOOP
[20] ⍝ THE HADAMARD PRODUCT
[21]  TRAAK←+/+/SPECIAL×FF×FF←FF1+.×FF2
[22]  MK←TRAK÷NOND ◊ SK←((TRAAK÷NOND)-MK*2)*0.5
```

```
[23]  UK←MK+SK×(NOND-1)*0.5
[24] ⍝ ACCUMULATE UPPER BOUNDS
[25]  UPPER←UPPER+UK
[26] ⍝ ACCUMULATE UPDATE MATRICES IN U
[27]  U←U+(PK,(-NLZ-1)↑LAG0 PK)+.×FT
[28]  →(NS≥K←K+1)/SETLOOP
[29] ⍝ THE FINAL UPDATE MATRIX U
[30]  U←U÷UPPER
[31] ⍝
[32] ⍝ ---- PROCRUSTES ROTATION FOR OPTIMAL LS FIT AND QUIT
[33] ⍝
[34]  SVD DEVMN Z+U ◇ Z←SVDP+.×⍉SVDQ ◇ →0
[35] ⍝
[36] ⍝ ---- NO LAGS ? DO IT THE EASY WAY..
[37] ⍝
[38] NOLAGS:U←(⍴Z)⍴0
[39]  CY←(0,⍳NLX-1)LAGOF X←G+.×Y ◇ K←1
[40] AVLOOP:U←U+CY+.×A[K;;] ◇ →(NS≥K←K+1)/AVLOOP
[41]  SVD DEVMN Z+U÷NS ◇ Z←SVDP+.×⍉SVDQ
      ∇
```

```
      ∇COR[□]∇
      ∇R←X COR Y;SX;SY;N
[1]  ⍝ RETURNS THE CORRELATION(S) BETWEEN TWO MATRICES X AND Y
[2]   N←(⍴X)[1]
[3]   SX←STAND X
[4]   SY←STAND Y
[5]   R←((⍉SX)+.×SY)÷N
      ∇
```

```
      ∇CORBOX[□]∇
      ∇Z←P CORBOX X;XL;XD;S;N;M
[1]  ⍝ CONSTRUCTS A CORRELATION BOX OF P LAGS OF DATA MATRIX X
[2]   →(P<0)/0 ◇ N←(⍴X)[1] ◇ M←(⍴X)[2]
[3]   Z←((P+1),M,M)⍴0
[4]   XD←STAND X
[5]   S←0
[6]  LAGLOOP:
[7]   XL←(1⍴S)LAGOF XD
[8]   Z[S+1;;]←(⍉XD)+.×XL
[9]   →(P≥S←S+1)/LAGLOOP
[10]  Z←Z÷N
      ∇
```

CORBOX computes the correlation box up to P lags.

Example:

```
     2 CORBOX DATA
   1          0.7314      0.2519
   0.7314     1           0.3739
   0.2519     0.3739      1
```

```
   0.4848      0.5276      0.6313
   0.3176      0.5563      0.4515
   0.03112     0.1537      0.2545

   0.03478     0.2492      0.492
   0.168       0.1696      0.3213
  ¯0.2415     ¯0.2534      0.1758


      ∇DEVMN[□]∇
      ∇Z←DEVMN A
[1]   ⍝ RETURNS MATRIX IN DEVIATIONS FROM ITS COLUMN MEANS
[2]    Z←A-(⍴A)⍴(+⌿A)÷(⍴A)[1]
      ∇


      ∇DIA[□]∇
      ∇Z←DIA V;N
[1]   ⍝ CONSTRUCTS THE DIAGONAL MATRIX FROM VECTOR V
[2]    Z←(N,N)⍴V,(N,N←⍴V)↑0
      ∇


       ∇DG[□]∇
      ∇Z←DG X;N
[1]   ⍝ ZEROES THE OFF-DIAGONAL ELEMENTS OF THE SQUARE MATRIX X
[2]    Z←(N,N)⍴(1 1⍉X),(N,N←(⍴X)[1])↑0
      ∇


      ∇EIGEN[□]∇
      ∇EIGEN S;R
[1]   ⍝ EIGENVECTOR/EIGENVALUE DECOMPOSITION OF A REAL SYMMETRIC
MATRIX
[2]   ⍝ THE RESULT ARE IN THE GLOBALS: EVC - EIGENVECTORS
[3]   ⍝                                EVA - EIGENVALUES (SORTED)
[4]   ⍝ CALLS JACOBI
[5]    →((⍴S)[1]=(⍴S)[2])/OK
[6]    'EIGEN: MATRIX IS NOT SQUARE -- NO EIGENVALUE DECOMPOSITION'
[7]    EVA←((⍴S)[1])⍴0 ⋄ EVC←(⍴S)⍴0 ⋄ →0
[8]   OK:→((⍴S)[1]>1)/JACO ⋄ EVA←1⍴S ⋄ EVC←1 1⍴1 ⋄ →0
[9]   JACO:EVC←1 0⍉R←1E¯10 JACOBI S ⋄ EVA←R[1;]
      ∇


        ∇EXFIL[□]∇
      ∇Z←F EXFIL X;N;T
[1]   ⍝ SIMPLE RECURSIVE EXPONENTIAL FILTER Z = F B Z + X
[2]    N←(⍴X)[1] ⋄ Z←(N,1)⍴0
[3]    Z[1;]←X[1;]
[4]    T←2
```

```
[5]  LOOP:
[6]   Z[T;]←(F×Z[T-1;])+X[T;]
[7]   →(N≥T←T+1)/LOOP
      ∇
```

EXFIL implements a direct, recursive exponential smoothing filter. It was used in Chapter 4. Parameter F is the smoothing coefficient.

```
      ∇FINROT[□]∇
      ∇FINROT;PP;PR;PQ;EVC;EVA;K
[1]  ⍝ FINAL ROTATION FOR CAB
[2]   PP←-P-(⍴P)⍴Z
[3]   K←1 ◇ PQ←(NS,NO,NO)⍴0
[4]   PR←(ND,ND)⍴0
[5]  SETLOOP:PR←PR+(⍉PP[K;;])+.×PP[K;;] ◇ →(NS≥K←K+1)/SETLOOP
[6]   EIGEN PR
[7]   Z←Z+.×EVC
[8]   X←CABMAX3 X
[9]   →(FILTER)/0 ◇ FA←CABMAF FA
      ∇


      ∇GRAM[□]∇
      ∇XX←GRAM YY;I;J;NP
[1]  ⍝ GRAM-SCHMIDT ORTHOGONALISATION OF MATRIX YY
[2]   NP←(⍴YY)[2]
[3]   I←1
[4]   XX←YY
[5]  STEP1:XX[;I]←XX[;I]÷(+/XX[;I]*2)*0.5
[6]   →0×⍳I=NP
[7]   J←I+1
[8]  STEP2:XX[;J]←XX[;J]-(+/XX[;I]×XX[;J])×XX[;I]
[9]   →STEP2×⍳NP≥J←J+1
[10]  I←I+1
[11]  →STEP1
      ∇
```

GRAM : author Jan de Leeuw.

```
      ∇INDICATOR[□]∇
      ∇G←INDICATOR DATA;J;R
[1]  ⍝ CONSTRUCTION OF THE INDICATOR MATRIX
[2]  ⍝ NEEDED: DATA, CONTIN, LEVELS -- DEFINES: G, CAT, CCAT
[3]   CCAT←+\CAT←1+(~CONTIN)×(⌈/DATA)-1
[4]  ⍝ -------------------- DISCRETE CASE ----------------------
[5]   J←1 ◇ R←0⍴0
[6]  LP1:R←R,⍳CAT[J] ◇ →((⍴DATA)[2]≥J←J+1)/LP1
[7]   G←(CAT∘.=DATA)=((⍴DATA)[1],(⍴R))⍴R
[8]  ⍝ -------------------- CONTINUOUS INTERVAL CASE ------------
[9]   →(0=⍴R←(+/R)↑⍳R←CONTIN^.=LEVELS=3)/JP1 ◇ J←1
[10] LP2:G[;CCAT[R[J]]]←DEVMN DATA[;R[J]] ◇ →((⍴R)≥J←J+1)/LP2
```

```
[11] ⍝ -------------------- ERROR CASE -------------------------
[12] JP1:→(~(0<+/CONTIN^LEVELS=1)∨(0<+/CONTIN^LEVELS=2))/0
[13]  'INDICATOR: INCOMPATIBLE VALUES IN CONTIN AND LEVELS'
     ∇


     ∇IP[□]∇
     ∇Z←IP X
[1]  ⍝ INPRODUCT SHORTCUT
[2]  Z←(⍉X)+.×X
     ∇


     ∇JACOBI[□]∇
     ∇R←E JACOBI M;NN;N;N2;J;K;PQ;CSS;TH;T;C;S;IT;MAXIT
[1]  ⍝ JACOBI METHOD -- REAL SYMMETRIC MATRIX
[2]  MAXIT←50
[3]  R←M,[J←1]NN⍴1,(1↑N2←(N←1↑NN←⍴M),K←2)⍴IT←0
[4]  M←,(⍳N)∘.=⍳N
[5]  L1:→(E>|R[J;K])/B1
[6]  TH←¯0.5×(-/1 1⍉CSS)÷(CSS←R[PQ;PQ←J,K])[1;2]
[7]  T←÷TH+(SGN TH)×(1+TH×TH)*0.5
[8]  S←T×C←÷(1+T×T)*0.5
[9]  R[PQ;]←⍉N2↑R[;PQ]←R[;PQ]+.×T←2 2⍴C,S,-S
[10] R[PQ;PQ]←(⍉T)+.×CSS+.×T
[11] B1:→(N≥K←K+1)/L1
[12] →(N≥K←1+J←J+1)/L1
[13] →(MAXIT>IT←IT+J←3-K←2)/B2
[14] 'JACOBI: CYCLE LIMIT EXCEEDED -- HOW MANY MORE ?'
[15] →(MAXIT≤IT←IT-□)/B3
[16] B2:→(E<⌈/|M/,NN↑R)/L1
[17] B3:R←(1 1⍉R),[1](N,0)↓R
[18] R←R[;⍒R[1;]]
     ∇
```

JACOBI comes from the STATLIB package of IBM.

```
     ∇KRONEK[□]∇
     ∇R←A KRONEK B
[1]  ⍝ RIGHT SIDED KRONECKER PRODUCT
[2]  R←((⍴A)×⍴B)⍴1 3 2 4⍉A∘.×B
     ∇
```

KRONEK demostrates the real power of the APL outer product. This function is from J.B. Ramsay's book on APL programming.

```
     ∇LAGOF[□]∇
     ∇Z←P LAGOF Y;N;M;I;S;PI;K
[1]  ⍝ CONSTRUCT REQUESTED LAGS MATRIX OF Y
[2]  ⍝ RESULT: P IS AN ARRAY CONTAINING LAG NUMBERS
```

```
[3]   Z←((⍴Y)[1],0)⍴0 ⋄ →((K←⍴P)=0)/0 ⋄ I←1
[4]  LOOP:PI←P[I]
[5]   S←(-PI)⌽[1]Y
[6]   →(PI=0)/CONCAT
[7]   →(PI<0)/LOW
[8]   S[⍳PI;]←0 ⋄ →CONCAT
[9]  LOW:S[((⍴Y)[1]+1-⍳|PI);]←0
[10] CONCAT:Z←Z,S
[11]  →(K≥I←I+1)/LOOP
      ∇
```

Example:

```
      (1 3) LAGOF 5 1⍴⍳5
 0 0
 1 0
 2 0
 3 1
 4 2
```

```
      ∇LAG0[□]∇
      ∇R←P LAG0 Y;N;M;I;S
[1]   ⍝ CONSTRUCTS A P'TH ORDER LAGGED SUPERMATRIX OF Y
[2]   ⍝ RESULT: R=B(1)Y,B(2)Y,...,B(P)Y   (PADDED BY ZEROES)
[3]   N←(⍴Y)[1] ⋄ R←(N,0)⍴0 ⋄ →(P=0)/0
[4]   M←(⍴Y)[2] ⋄ I←0 ⋄ →(P<0)/LOW
[5]  LOOP:I←I+1 ⋄ S←(-I)⌽[1]Y ⋄ S[⍳I;]←(I,M)⍴0 ⋄ R←R,S
[6]   →(I<P)/LOOP ⋄ →0
[7]  LOW:I←I+1 ⋄ S←(I)⌽[1]Y ⋄ S[N+1-⍳|I;]←0 ⋄ R←R,S
[8]   →(I<|P)/LOW
      ∇
```

Example:

```
      2 LAG0 5 1⍴⍳5
 0 0
 1 0
 2 1
 3 2
 4 3
```

```
      ∇LIDICA[□]∇
      ∇YU←LIDICA YU;CONST;DELTA;GAMMA;EPLUS;DPLUS;H1;H2;ALPHA;BETA;KJ
[1]   ⍝ PLACES A LINEAR RESTRICTION ON VECTOR YU
[2]   ⍝ NEEDED: DJ, YU
[3]    KJ←⍴DJ
[4]    DELTA←(+/DJ×⍳KJ)÷CONST←+/DJ
[5]    GAMMA←(+/DJ×YU)÷CONST
[6]    EPLUS←YU-GAMMA
[7]    DPLUS←(⍳KJ)-DELTA
[8]    H1←+/EPLUS×DPLUS×DJ
[9]    H2←+/DPLUS×DPLUS×DJ
[10]   ALPHA←|H1÷H2
```

```
[11]  BETA←GAMMA-ALPHA×DELTA
[12]  YV←BETA+ALPHA×⍳KJ
      ∇
```

LIDICA is used to restrict the category quantifications of interval variables to equal distance.

```
      ∇MAXSIVAL[□]∇
      ∇ALPHA←Q MAXSIVAL X;P;QOLD;R;S
[1]   ⍝ MAXIMAL VALUE SINGULAR VALUE OF X BY VECTOR ITERATION
[2]   ⍝ Q IS THE INITIAL ESTIMATE
[3]   ⍝ NEEDED: SINGVEC (STORES LAST SINGULAR VECTOR)
[4]   LOOP:Q←S÷((⍉S)+.×S←(⍉X)+.×P←R÷((⍉R)+.×R←X+.×QOLD←Q)*0.5)*0.5
[5]    →(1E¯5<+/(QOLD-Q)*2)/LOOP
[6]    ALPHA←(⍉P)+.×X+.×SINGVEC←Q
       ∇


      ∇MANOR[□]∇
      ∇XHAT←MANOR XX;ACBL;BLS;BLV;KBL;NBL
[1]   ⍝ UNWEIGHTED MONOTONE REGRESSION OF VECTOR XX
[2]    BLS←(NBL←⍴BLV←XX)⍴ACBL←1
[3]   SAT:→SHIFT×⍳BLV[ACBL]≤BLV[ACBL+1]
[4]   MERGE:BLV[ACBL]←(BLV[ACBL]×BLS[ACBL])+BLV[ACBL+1]×BLS[ACBL+1]
[5]    BLV[ACBL]←BLV[ACBL]÷BLS[ACBL]←BLS[ACBL]+BLS[ACBL+1]
[6]    BLS←(ACBL↑BLS),(-NBL-ACBL+1)↑BLS
[7]    BLV←(ACBL↑BLV),(-NBL-ACBL+1)↑BLV
[8]    NBL←NBL-1
[9]    →SHAFT×⍳1=ACBL
[10]   ACBL←ACBL-1
[11]   →SAT
[12]  SHIFT:ACBL←ACBL+1
[13]  SHAFT:→SAT×⍳NBL>ACBL
[14]   XHAT←⍳¯1+KBL←1
[15]  LYM:XHAT←XHAT,BLS[KBL]⍴BLV[KBL]
[16]   →LYM×⍳NBL≥KBL←KBL+1
       ∇
```

MANOR: author Jan de Leeuw.

```
       ∇MEAN[□]∇
      ∇Z←MEAN A
[1]   ⍝ RETURNS THE COLUMNS MEANS OF A
[2]    Z←(+/A)÷(⍴A)[1]
       ∇


      ∇NOFUZZ[□]∇
      ∇Z←A NOFUZZ B
[1]   ⍝ CONVERTS VALUES IN B THAT ARE SMALLER THAN ⎕CT TO REAL ZEROES
```

```
[2]  ⍝ USUALLY ⎕CT IS EQUAL TO 1E¯13
[3]  ⍝ THE LEFT ARGUMENT CAN BE USED TO SPECIFY AN OTHER COMPARISON
[4]  ⍝ VALUE, E.G.: 1E¯8 NOFUZZ 10*-⍳10
[5]   →(0≠⎕NC'A')/L1
[6]   A←⎕CT
[7]  L1:Z←B×A<|B
      ∇
```

NOFUZZ: author Wim de Brinker

```
      ∇QSTAT[⎕]∇
      ∇Q←QSTAT X;N
[1]  ⍝ COMPUTES THE Q STATISTIC OF THE RESIDUALS X USING 25 LAGS
[2]  ⍝ THE FIRST VALUE IS THE BOX-PIERCE STATISTIC, THE SECOND ONE
[3]  ⍝ IS THE MODIFIED BOX-PIERCE STATISTIC
[4]   N←(⍴X)[1]
[5]   ⎕←Q←Nx+/(R←,1 0 0↓25 CORBOX X)*2
[6]   Q←Nx(N+2)x+/+/+/(R*2)÷N-⍳25
      ∇


      ∇STAND[⎕]∇
      ∇Z←STAND A
[1]  ⍝ RETURNS MATRIX A IN STANDARD SCORES (ZERO MEAN, UNIT VARIANCE)
[2]   Z←(DEVMN A)÷(⍴A)⍴STD A
      ∇


      ∇SVD[⎕]∇
      ∇SVD M
[1]  ⍝ SINGULAR VALUE DECOMPOSITION OF MATRIX M
[2]  ⍝ RESULTS ARE: SVDP, SVDQ, SVDPHI
[3]   EIGEN(⍉M)+.xM
[4]   RANK←(⍴M)[2]-+/0=NOFUZZ EVA
[5]   EVA←RANK↑EVA ◇ EVC←((⍴M)[2],RANK)↑EVC
[6]   SVDQ←EVC ◇ SVDPHI←EVA*0.5 ◇ SVDP←M+.xSVDQ+.xDIA EVA*¯0.5
      ∇


      ∇SGN[⎕]∇
      ∇Z←SGN A
[1]  ⍝ SIGN OF A
[2]   Z←(A≥0)-A<0
      ∇


      ∇STD[⎕]∇
      ∇Z←STD A
[1]  ⍝ STANDARD DEVIATION
[2]   Z←(VAR A)*0.5
      ∇
```

```
      ∇UNIT[□]∇
      ∇Z←UNIT N
[1]   ⍝ RETURNS AN N x N IDENTITY MATRIX
[2]    Z←(⍳N)∘.=⍳N
      ∇
```

```
      ∇VAR[□]∇
      ∇Z←VAR A
[1]   ⍝ VARIANCE
[2]    Z←+⌿((DEVMN A)*2)÷(⍴A)[1]
      ∇
```

Akaike, H. (1974). Markovian representations of stochastic processes and its application to the analysis of autoregressive moving average processes. *Annals of the Institute of Statistical Mathematics*, *26*, 363-387.

Akaike, H. (1976). Canonical correlation analysis of time series and the use of an information criterion. In: R.K. Mehra & D.G. Lainiotis (Eds.), *System identification: advances and case studies*. Academic Press, New York.

Allport, G. (1961). *Pattern and growth in personality*. Holt, Rinehart, Winston, New York.

Anderson, T.W. (1958). *An introduction to multivariate statistical analysis*. Wiley, New York.

Anderson, T.W. (1963). The use of factor analysis in the statistical analysis of multiple time series. *Psychometrika*, *28*, 1-25.

Anderson, T.W. (1971). *The statistical analysis of time series*. Wiley, New York.

Aoki, M. (1987). *State space modelling of time series*. Springer-Verlag, Berlin.

Bartlett, M.S. (1946). On the theoretical specification of sampling properties of autocorrelated time series. *Journal of the Royal Statistical Society B*, *8*, 27-41.

Bartlett, M.S. (1947). The general canonical correlation distribution. *Annals of Mathematical Statistics*, *18*, 1-17.

Bekker, P. & Leeuw, J. de (1988). Relations between variants of non-linear principal component analysis. In: J.L.A. van Rijckevorsel & J. de Leeuw (Eds.), *Component and correspondence analysis*, 1-31. Wiley, Chichester.

Bennett, R.J. (1979). *Spatial time series: forecasting and control*. Pion, London.

Benzécri, J.-P. (1977). Histoire et préhistoire de l'analyse des données V: l'analyse des correspondences. *Cahiers de l'Analyse des Données*, *2*, 9-40.

Berge, J.M.F. ten (1977). Orthogonal Procrustes rotation for two or more matrices. *Psychometrika*, *42*, 267-276.

Berge, J.M.F. ten & Knol, D.L. (1984). Orthogonal rotations to maximal agreement for two or more matrices of different column orders. *Psychometrika*, *49*, 49-55.

Bijleveld, C.C.J.H. (1989). *Exploratory linear dynamic systems analysis*. Dissertation, University of Leiden. DSWO Press, Leiden.

Bishop, Y.M.M., Fienberg, S.E. & Holland, P.W. (1975). *Discrete multivariate analysis*. M.I.T. Press, Cambridge.

Boom, D.C. van den (1988). *Neonatal irritability and the development of attachment: observation and intervention*. Dissertation, University of Leiden.

Box, G.E.P. & Jenkins, G.M. (1976). *Time series analysis, forecasting and control (revised edition)*. Holden-Day, San Francisco.

Box, G.E.P. & Tiao, G.C. (1965). A change in level of a nonstationary time series. *Biometrika*, *52*, 181-192.

Box, G.E.P. & Tiao, G.C. (1975). Intervention analysis with applications to economic and environmental problems. *Journal of the American Statistical Association*, *70*, 70-79.

Box, G.E.P. & Tiao, G.C. (1977). A canonical analysis of multiple time series. *Biometrika*, *64*, 355-365.

Braak, C.J.F. ter (1986). Canonical correspondence analysis: A new eigenvector technique for multivariate direct gradient analysis. *Ecology*, *67*, 1167-1179.

Brillinger, D.R. (1969). The canonical analysis of stationary time series. In: P.R. Krishnaiah (Ed.), *Multivariate analysis II*, 331-350. Academic Press, New York.

Brillinger, D.R. (1975). *Time series: data analysis and theory*. Holt, Rinehart & Winston, New York.

Burg, E. van der (1984). *Homogeneity analysis of a time series*. RR-84-02. Dept. of Data Theory, University of Leiden.

Burg, E. van der (1988). *Nonlinear canonical correlation and some related techniques*. Dissertation, University of Leiden. DSWO Press, Leiden.

Burg, E. van der & Leeuw, J. de (1983). Nonlinear canonical correlation. *British Journal of Mathematical and Statistical Psychology*, *36*, 54-80.

Buuren, S. van (1986). *GROUPALS: a method to cluster objects for variables with mixed measurement levels*. RR-86-10, Dept. of Data Theory, University of Leiden.

Buuren, S. van & Dijksterhuis, G.B. (1988). Procrustes analysis of discrete data. In: M.G.H. Jansen & W.H. van Schuur (Eds.), *The many faces of multivariate analysis: proceedings of the SMABS-88 conference*, 53-66. Rion, University of Groningen.

Buuren, S. van & Heiser, W.J. (1989). Clustering n objects into k groups under optimal scaling of variables. *Psychometrika*, in press.

Cabannes, J.P. (1978). Analyse de quelques séries relatives au chômage. *Cahiers de l'Analyse des Données*, *3*, 366-399. Reprinted in: J.-P. Benzécri et al. (1986), *Pratique de l'analyse des données en économie*. Dunod, Paris.

Cabannes, J.P. (1981). Analyse des séries temporelles de chômage et essais de prévision. *Cahiers de l'Analyse des Données*, *6*, 87-98. Reprinted in: J.-P. Benzécri et al. (1986), *Pratique de l'analyse des données en économie*. Dunod, Paris.

Caines, P.E. (1987). *Linear stochastic systems*. Wiley, New York.

Campbell, D.T. & Stanley, J.C. (1966). *Experimental and quasi-experimental designs for research*. Rand McNally & Co., Chicago.

Carlier, A., Lavit, C., Pages, M., Pernin, M.O. & Turlot, J.C. (1988). *Analysis of data tables indexed by time: a comparative review*. Paper presented at Multiway 88, Rome.

Cattell, R.B. (1952). *Factor analysis; an introduction and manual for the psychologist and social scientist*. Harper, New York.

Cattell, R.B. (1957). *Personality and motivation: structure and measurement*. World Book, Yonkers-on-Hudson.

Cattell, R.B. (1963). The structuring of change by P-technique and incremental R-technique. In C.W. Harris (Ed.), *Problems in measuring change*. University of Wisconsin Press, Madison.

Cook, T.D. & Campbell, D.T. (1979). *Design and analysis of quasi-experiments for field settings*. Rand-McNally, Chicago.

Coolen, H. & Leeuw, J. de (1987). *Least squares path analysis with optimal scaling*. RR-87-03, Dept. of Data Theory, University of Leiden.

Cryer, J.D. (1986). *Time series analysis*. Duxbury Press, Boston.

Deeg, D.J.H., Hofman, A. & Zonneveld, R.J. van (1989). *The association between change in cognitive function and longevity in dutch elderly*. Submitted.

Deeg, D.J.H., Zonneveld, R.J. van, Maas, P.J. van der & Habbema, J.D.F. (1985). *Levensverwachting en lichamelijke, psychische en sociale kenmerken bij bejaarden*. Instituut Maatschappelijke Gezondheidszorg, Erasmus Universiteit, Rotterdam.

Dempster, A.P., Laird, N.M. & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM-algorithm. *Journal of the Royal Statistical Society B*, *39*, 1-38.

Deville, J.-C. & Saporta, G. (1980). Analyse harmonique qualitative. In E. Diday et al. (Eds.), *Data analysis and informatics*. North-Holland, Amsterdam.

Deville, J.-C. & Saporta, G. (1983). Correspondence analysis with an extension towards nominal time series. *Journal of Econometrics*, *22*, 169-189.

Efron, B. (1979). Bootstrap methods: another look at the Jackknife. *Annals of Statistics*, *7*, 1-26.

Efron, B. & Gong, G.A. (1983). A leisurely look at the Bootstrap, the Jackknife and cross-validation. *American Statistician*, *37*, 36-48.

El Moussaoui, A. (1987). Disparités départementales en France d'après la consommation mensuelle de quatre produits pétroliers. *Cahiers de l'Analyse des Données*, *12*, 169-194.

Engle, R.F. & Watson, M. (1981). A one-factor multivariate time series model for metropolitan wage rates. *Journal of the American Statistical Association*, *76*, 774-781.

Eykhoff, P. (1974). *System identification*. Wiley, London.

Fienberg, S.E. (1980). *The analysis of cross-classified categorical data (second edition)*. M.I.T. Press, Cambridge.

Fisher, R.A. (1940). The precision of discriminant functions. *Annals of Eugenics*, *10*, 422-429.

Flinn, J.F. & Heckman, J.J. (1982). New methods for analyzing individual event histories. In: S. Leinhardt (Ed.), *Sociological Methodology 1982*, 99-140.

Fortier, J.J. (1966). Simultaneous linear prediction. *Psychometrika*, *31*, 369-381.

Geer, J.P. van de (1986). Relations among k sets of variables with geometrical representation and application to categorical variables. In: J. de Leeuw et al. (Eds.), *Multidimensional data analysis*, 67-79. DSWO Press, Leiden.

Gelb, A. (Ed.) (1974). *Applied optimal estimation*. M.I.T. Press, Cambridge, MA.

Geweke, J.F. (1977). The dynamic factor analysis of economic time series models. In: D.V. Aigner & A.S. Goldberger (Eds.), *Latent variables in socio-economic models*, 365-383. North-Holland, Amsterdam.

Geweke, J.F. & Singleton, K.J. (1981). Maximum likelihood "confirmatory" factor analysis of economic time series. *International Economic Review*, *22*, 37-54.

Gifi, A. (1980). *Niet-lineaire multivariate analyse*. Dept. of Data Theory, University of Leiden.

Gifi, A. (1981). *Nonlinear multivariate analysis*. Dept. of Data Theory, University of Leiden.

Gittins, R. (1985). *Canonical analysis: a review with applications in ecology*. Springer-Verlag, Berlin.

Glass, G.V., Willson, V.L. & Gottman, J.M. (1975). *Design and analysis of time-series experiments*. Colorado Associated University Press, Boulder, Colorado.

Goldstein, H. (1987). The choice of constraints in correspondence analysis. *Psychometrika*, *52*, 207-215.

Goodman, L.A. (1978). *Analyzing qualitative/categorical data*. Addison Wesley, Reading.

Goodwin, G.C. & Payne, R.L. (1977). *Dynamic system identification: experiment design and data analysis*. Academic Press, New York.

Gottman, J.M. (1981). *Time series analysis*. Cambridge University Press, Cambridge.

Gower, J.C. (1975). Generalized Procrustes analysis. *Psychometrika*, *40*, 33-51.

Granger, C.W.J. & Newbold, P. (1977). *Forecasting economic time series*. Academic Press, New York.

Greenacre, M.J. (1984). *Theory and applications of correspondence analysis*. Academic Press, London.

Gregson, R.A.M. (1983). *Time series in psychology*. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Guttman, L. (1941). The quantification of a class of attributes: A theory and method of scale construction. In: P. Horst et al. (Eds.), *The prediction of personal adjustment*, 317-348. Social Science Research Council, New York.

Guttman, L. (1966). *The nonmetric breakthrough for the behavioral sciences*. Proceedings on the second national conference of data processing. Information processing association of Israel, 495-510.

Haan, E. de, Hoogduin, K. & Jong, P. de (1989). Geleidelijke exposure bij fobische klachten. In: K. van der Velden (Ed.), *Directieve therapie 3*, 216-225. Van Loghum Slaterus, Deventer.

Hannan, E.J. (1967). Canonical correlation and multiple equation systems in economics. *Econometrica*, *35*, 79-90.

Hannan, E.J. (1970). *Multiple time series*. Wiley, New York.

Hannan, E.J. & Poskitt, D.S. (1988). Unit canonical correlations between future and past. *Annals of Statistics*, *16*, 784-790.

Hartigan, J.A. (1975). *Clustering algorithms*. Wiley, New York.

Harvey, A.S., Szalai, A., Elliot, D.H., Stone, P.J. & Clark, S.M. (1984). *Time budget research*. Campus Verlag, New York.

Hathout, A. (1987). Etude préalable à la constitution d'une cible pour l'identification des valeurs mobilières dans la période comprise entre 18/10/85 et le 31/3/86. *Cahiers de l'Analyse des Données*, *12*, 91-110.

Hayashi, C. (1952). On the predictions of phenomena from qualitative data and quantifications from the mathematico-statistical point of view. *Annals of the Institute of Statistical Mathematics*, *3*, 69-92.

Healy, M.J.R. & Goldstein, H. (1976). An approach to the scaling of categorized attributes. *Biometrika*, *63*, 219-229.

Heijden, P.G.M. van der (1987). *Correspondence analysis of longitudinal categorical data*. Dissertation, University of Leiden. DSWO Press, Leiden.

Heijden, P.G.M. van der & Leeuw, J. de (1985). Correspondence analysis used complementary to loglinear analysis. *Psychometrika*, *50*, 429-447.

Heiser, W.J. (1981). *Unfolding analysis of proximity data*. Dissertation, University of Leiden.

Heiser, W.J. & Meulman, J.J. (1987). Afstandsmodellen voor multivariate analyse. In: H.F.M. Crombag, L.J.Th. van der Kamp & C.A.J. Vlek (Eds.), *De psychologie voorbij: ontwikkelingen rond model en methode in de gedragswetenschappen*, 209-235. Swets & Zeitlinger, Lisse.

Hersen, M. & Barlow, D.H. (1976). *Single case experimental design: strategies for studying behavior change*. Pergamon Press, Oxford.

Hibbs, D.A., jr. (1977). On analyzing the effects of policy interventions: Box-Jenkins and Box-Tiao versus structural equation models. In: D.R. Heise (Ed.), *Sociological Methodology 1977*. Jossey-Bass, San Francisco.

Hoogduin, K. (1989). Directieve therapie bij een man met paranoide schizofrenie. In: K. van der Velden (Ed.), *Directieve therapie 3*, 117-124. Van Loghum Slaterus, Deventer.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, *28*, 321-377.

Huitema, B.E. (1986). Statistical analysis and single subject designs: some misunderstandings. In: A. Poling & R.W. Fuqua (Eds.), *Research methods in applied behavior analysis: issues and advances*. Plenum Press, New York.

Hutt, S.J., Lenard, H.G. & Prechtl, H.F.R. (1969). Psychophysiological studies in newborn infants. In: L.P. Lipsett & H.W. Reese (Eds.), *Advances in child development and behavior*. Academic Press, New York.

Iacobucci, D. & Wasserman, S. (1988). A general framework for the statistical analysis of sequential dyadic interaction data. *Psychological Bulletin*, *103*, 379-390.

Immink, W. (1986). *Parameter estimation in Markov models and dynamic factor analysis*. Dissertation, University of Utrecht.

Israëls, A.Z. (1987). *Eigenvalue techniques for qualitative data*. Dissertation, University of Leiden. DSWO Press, Leiden.

Izenman, A.J. (1975). Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, *5*, 248-264.

Jenkins, G.M. & Alavi, A.S. (1981). Some aspects of modelling and forecasting time series. *Journal of Time Series Analysis*, *2*, 1-47.

Jenkins, G.M. & Watts, D.G. (1968). *Spectral analysis and its applications*. Holden-Day, San Francisco.

Jewell, N.P. & Bloomfield, P. (1983). Canonical correlations of past and future for time series: definitions and theory. *Annals of Statistics*, *11*, 837-847.

Jewell, N.P., Bloomfield, P. & Bartmann, F.C. (1983). Canonical correlations of past and future for time series: bounds and computation. *Annals of Statistics*, *11*, 848-855.

Jones, R.R., Vaught, R.S. & Weinrott, M. (1977). Time series analysis in operant research. *Journal of Applied Behavior Analysis*, *10*, 151-166.

Judge, G.G., Griffiths, W.E., Hill, R.C., Lütkepohl, H. & Lee, T.-C. (1985). *The theory and practice of econometrics (second edition)*. Wiley, New York.

Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, *82*, 35-46. Reprinted in: T. Kailath (Ed.), *Linear least-squares estimation*. Dowder, Hutchinson & Ross, Stroudsburg, Pa.

Kalman, R.E. & Bucy, R.S. (1961). New results in linear filtering and prediction theory. *Journal of Basic Engineering*, *83*, 95-108.

Kazdin, A.E. (1980). *Research design in clinical psychology*. Harper & Row, New York.

Kazdin, A.E. (1982). *Single case research designs: methods for clinical and applied settings*. Oxford University Press, New York.

Keller, W.J. & Wansbeek, T. (1983). Multivariate methods for quantitative and qualitative data. *Journal of Econometrics*, *22*, 91-111.

Kendall, M.G. & Stuart, A. (1968). *The advanced theory of statistics III: design and analysis, and time series*. Griffin & Co., London.

Kratochwill, T.R. (Ed.) (1978). *Single subject research: strategies for evaluating change*. Academic Press, New York.

Kruskal, J.B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, *29*, 1-27.

Kruskal, J.B. (1965). Analysis of factorial experiments by estimating monotone transformations of the data. *Journal of the Royal Statistical Society B*, *27*, 251-263.

Land, K.C. (1980). Modeling macro social change. In: K.F. Schuessler (Ed.), *Sociological Methodology 1980*. Jossey-Bass, San Francisco.

Landis, J.R. & Koch, G.G. (1979). The analysis of categorical data in longitudinal studies of behavioral development. In: J.R. Nesselroade & P.B. Baltes (Eds.), *Longitudinal research in the study of behavior and development*, 233-261. Academic Press, New York.

Lans, I.A. van der & Heiser, W.J. (1988). *Nonlinear multiple regression analysis with common scale transformations across predictor variables*. RR-88-10, Dept. of Data Theory, University of Leiden.

Leeuw, J. de (1973). *Canonical analysis of categorical data*. Dissertation, University of Leiden. Published in 1984 by DSWO Press, Leiden.

Leeuw, J. de (1977). Applications of convex analysis to multidimensional scaling. In: J.R. Barra et al. (Eds.), *Progress in Statistics*. North-Holland, Amsterdam.

Leeuw, J. de (1983a). On the prehistory of correspondence analysis. *Statistica Neerlandica*, *37*, 161-164.

Leeuw, J. de (1983b). Models and methods for the analysis of correlation coefficients. *Journal of Econometrics*, *22*, 113-137.

Leeuw, J. de (1984a). The Gifi-system of nonlinear multivariate analysis. In: E. Diday et al. (Eds.), *Data analysis and informatics III*, 415-424. North-Holland, Amsterdam.

Leeuw, J. de (1984b). *Statistical properties of multiple correspondence analysis*. RR-84-06, Dept. of Data Theory, University of Leiden.

Leeuw, J. de (1986). *Multivariate analysis with optimal scaling*. RR-86-01, Dept. of Data Theory, University of Leiden.

Leeuw, J. de & Bijleveld, C.C.J.H. (1988). *Fitting longitudinal reduced rank regression models by alternating least squares*. Dept. of Data Theory, RR 88-03, University of Leiden.

Leeuw, J. de, Heijden, P.G.M. van der & Kreft, I. (1985). Homogeneity analysis of event history data. *Methods of Operations Research*, *50*, 299-316.

Leeuw, J. de & Heiser, W.J. (1980). Multidimensional scaling with restrictions on the configuration. In: P.R. Krishnaiah (Ed.), *Multivariate analysis V*, 501-522. North-Holland, Amsterdam.

Leeuw, J. de & Meulman, J.J. (1986). Principal component analysis and restricted multidimensional scaling. In: W. Gaul & M. Schrader (Eds.), *Classification as a tool of research*, 83-96. North-Holland, Amsterdam.

Leeuw, J. de & Mooijaart, A. (1987). Multivariate analyse van lineaire structurele modellen. In: In: H.F.M. Crombag, L.J.Th. van der Kamp & C.A.J. Vlek (Eds.), *De psychologie voorbij: ontwikkelingen rond model en methode in de gedragswetenschappen*, 167-182. Swets & Zeitlinger, Lisse.

Leeuw, J. de & Rijckevorsel, J.L.A. van (1980). HOMALS and PRINCALS: Some generalizations of principal components analysis. In: E. Diday et al. (Eds.), *Data analysis and informatics*, 231-242. North-Holland, Amsterdam.

Leeuw, J. de & Rijckevorsel, J.L.A. van (1988). Beyond homogeneity analysis. In: J.L.A. van Rijckevorsel & J. de Leeuw (Eds.), *Component and correspondence analysis*, 55-80. Wiley, Chichester.

Ljung, L. (1985). Estimation of parameters in dynamical systems. In: E.J. Hannan et al. (Eds.), *Handbook of statistics V*, 189-211. North-Holland, Amsterdam.

Ljung, L. & Söderström, T. (1983). *Theory and practice of recursive identification methods*. M.I.T. Press, Cambridge, Massachusetts.

MacCallum, R. & Ashby, F.G. (1986). Relationships between linear systems theory and covariance structure modelling. *Journal of Mathematical Psychology*, *30*, 1-27.

MacCleary, R. & Hay, R.A. (1980). *Applied time series analysis for the social sciences*. Sage, Beverly Hills.

Magnus, J.R. & Neudecker, H. (1988). *Matrix differential calculus with applications in statistics and econometrics*. Wiley, Chichester.

Makridakis, S. (1976). A survey of time series. *International Statistical Review*, *44*, 29-70.

Makridakis, S. (1978). Time series analysis and forecasting: An update and evaluation. *International Statistical Review*, *46*, 255-278.

Mason, W.M. & Fienberg, S.E. (Eds.) (1985). *Cohort analysis in social research*. Springer-Verlag, New York.

McKeon, J.J. (1966). *Canonical analysis: some relations between canonical correlation, factor analysis, discriminant function analysis and scaling theory*. Psychometric Monograph 13. University of Chicago Press, Chicago.

Mefferd, R.B., Moran, L.J. & Kimble, J.P. (1958). Use of a factor analytic technique in the analysis of long term repetitive measurements made upon a single schizophrenic patient. (Unpublished, cf. Anderson, 1963).

Meulman, J.J. (1982) *Homogeneity analysis of incomplete data*. DSWO Press, Leiden.
Meulman, J.J. (1986) *A distance approach to nonlinear multivariate analysis*. Dissertation, University of Leiden. DSWO Press, Leiden.
Molenaar, P.C.M. (1981). *Dynamische factormodellen*. Dissertation, University of Utrecht.
Molenaar, P.C.M. (1985). A dynamic factor model for the analysis of multivariate time series. *Psychometrika*, *50*, 181-202.
Molenaar, P.C.M. (1987). Dynamic factor analysis in the frequency domain: Causal modelling of multivariate psychophysiological time series. *Multivariate Behavioral Research*, 22, 329-353.
Newbold, P. (1981). Some recent developments in time series analysis. *International Statistical Review*, *49*, 53-66.
Newbold, P. (1984). Some recent developments in time series analysis II. *International Statistical Review*, *52*, 183-192.
Newbold, P. (1988). Some recent developments in time series analysis III. *International Statistical Review*, *56*, 17-29.
Nijkamp, P., Leitner, H. & Wrigley, N. (Eds.) (1985). *Measuring the unmeasurable*. Martinus Nijhoff Publishers, Dordrecht.
Nishisato, S. (1980). *Analysis of categorical data: dual scaling and its applications*. University of Toronto Press, Toronto.
O'Connell, P.E. (1984). Kalman filtering. In: Ledermann, W. (Ed.), *Handbook of applicable mathematics VI: Statistics B*, 897-938. Wiley, Chichester.
Otter, P.W. (1986). Dynamic structural systems under indirect observation: Identifiability and estimation aspects from a system theoretic perspective. *Psychometrika*, *51*, 415-428.
Oud, J.H., Bercken, J.H. van der & Essers, R.J. (1986). Longitudinal factor score estimation using the Kalman filter. *Kwantitatieve Methoden*, *20*, 109-129.
Parzen, E. & Newton, H.J. (1980). Multiple time series modelling II. In: P.R. Krishnaiah (Ed.), *Multivariate analysis V*, 181-197. North-Holland, Amsterdam.
Peay, E.R. (1988). Multidimensional rotation and scaling of configurations to optimal agreement. *Psychometrika*, *53*, 199-208.
Picci, G. & Pinzoni S. (1986). A new class of dynamic models for stationary time series. In: S. Bittanti (Ed.), *Time series and linear systems*, 67-114. Springer-Verlag, Berlin.
Priestley, M.B., Subba Rao, T. & Tong, H. (1973). Identification of the structure of multivariable stochastic systems. In: P.R. Krishnaiah (Ed.), *Multivariate Analysis III*, 351-368. Academic Press, New York.
Priestley, M.B., Subba Rao, T. & Tong, H. (1974). Applications of principal component analysis and factor analysis in the identification of multi-variable systems. *IEEE Transactions on Automatic Control*, *AC-19*, 730-734.
Quenouille, M.H. (1957). *The analysis of multiple time series*. Griffin & Co., London.
Ramsay, J.O. (1982). When the data are functions. *Psychometrika*, *47*, 379-396.
Rijckevorsel, J.L.A. van (1987). *The application of fuzzy coding and horseshoes in multiple correspondence analysis*. Dissertation, University of Leiden. DSWO Press, Leiden.
Rijken van Olst, B.T. (1981). *Canonical correlation and canonical variables in econometrics*. Dissertation, University of Groningen.
Robinson, P.M. (1973). Generalized canonical analysis for time series. *Journal of Multivariate Analysis*, *3*, 141-160.
Saporta, G. (1981). *Méthodes exploratoires d'analyse de données temporelles*. Dissertation. l'Université P. et M. Curie, Paris.

Sargent, T.J. & Sims, C.A. (1977). Business cycle modelling without pretending to have too much a priori economic theory. In: C.A. Sims (Ed.), *New methods of business cycle research*. Federal Reserve Bank of Minneapolis, Minneapolis.

Shumway, R.H. (1988). *Applied statistical time series analysis*. Prentice-Hall, Englewood Cliffs.

Singer, B. & Spielerman, S. (1976). The representation of social processes by Markov models. *American Journal of Sociology*, *82*, 1-54.

Spliid, H. (1983). A fast estimation method for the vector autoregressive moving average model with exogenous variables. *Journal of the American Statistical Association*, *78*, 843-849.

Stobberingh, R.A. (1972). *Dynamische componenten analyse: een integratie van componenten- en tijdreeksanalyse*. Dissertation, University of Tilburg.

Stoer, J. & Bulirsch, R. (1980). *Introduction to numerical analysis*. Springer-Verlag, New York.

Tenenhaus, M. (1988). *Generalized canonical analysis and principal component analysis of three-way data*. Paper presented at Multiway 88, Rome.

Tenenhaus, M. & Young, F.W. (1985). An analysis and synthesis of multiple correspondence analysis, optimal scaling, dual scaling, homogeneity analysis and other methods for quantifying categorical multivariate data. *Psychometrika*, *50*, 91-119.

Tiao, G.C. & Box, G.E.P. (1981). Modelling multiple time series with applications. *Journal of the American Statistical Association*, *78*, 843-849.

Tiao, G.C. & Tsay, R.S. (1983). Multiple time series modelling and extended sample cross-correlations. *Journal of Business and Economic Statistics*, *1*, 43-56.

Tsay, R.S. & Tiao, G.C. (1985). Use of canonical analysis in time series model identification. *Biometrika*, *72*, 299-315.

Tuma, N.B., Hannan, M.T. & Groeneveld, L.P. (1979). Dynamic analysis of event histories. *American Journal of Sociology*, *84*, 820-854.

Velu, R.P., Reinsel, G.C. & Wichern, D.W. (1986). Reduced rank regression models for multiple time series. *Biometrika*, *73*, 105-118.

Vidgerhous, G. (1978). Forecasting sociological phenomena: Application of Box-Jenkins methodology to suicide rates. In: K.F. Schuessler (Ed.), *Sociological Methodology 1978*. Jossey-Bass, San Francisco.

Visser, R.A. (1982). *On quantitative longitudinal data in psychological research*. Dissertation, University of Leiden. Published in 1985 as: Analysis of longitudinal data in behavioural and social research. DSWO Press, Leiden.

Whittle, P. (1953). The analysis of multiple time series. *Journal of the Royal Statistical Society B*, *15*, 125-139.

Willems, J.C. (1978). Recursive filtering. *Statistica Neerlandica*, *32*, 1-39.

Winsberg, S. & Ramsay, J.O. (1980). Monotonic transformations to additivity using splines. *Biometrika*, *67*, 669-674.

Winsberg, S. & Ramsay, J.O. (1983). Monotone spline transformations for dimension reduction. *Psychometrika*, *46*, 171-186.

Winship, C. (1978). The allocation of time among individuals. In: K.F. Schuessler (Ed.), *Sociological Methodology 1978*. Jossey-Bass, San Francisco.

Wold, H.O. (1938). *A study in the analysis of stationary time series*. Almquist and Wicksell, Uppsala.

Wolkowicz, H. & Styan, G.P.H. (1980). Bounds for eigenvalues using traces. *Linear Algebra and Its Applications*, *29*, 471-506.

Wollenberg, A.L. van den (1977). Redundancy analysis: An alternative for canonical correlation analysis. *Psychometrika*, *42*, 207-219.

Young, F.W. (1981). Quantitative analysis of qualitative data. *Psychometrika*, *46*, 347-388.

Summary

This thesis discusses a method to analyze categorical time series. A categorical time series is a series in which each observation falls into one of a finite number of distinct categories.

Most time series analysis techniques are based on the assumption that the series are measured on an interval scale and this assumption prevents the analysis of categorical series by these method. The approach adopted in this thesis is to quantify each separate category, that is, we replace category identifications by numbers. The advantage of doing so is that traditional techniques can then be applied to the resulting quantified series.

There are of course many ways to assign numbers to categories. We confine ourselves to optimal scaling, a specific form of quantification. The idea of optimal scaling is that we look for those quantifications that are optimal for some mathematical criterion. We are free to choose the criterion and it depends on the goal of our analysis. Examples of suitable criteria in time series analysis are: fitting the smoothest curve to the series, reducing many series to one or a few representative series without discarding too much information, fitting the series after a stochastic process or finding adequate predictions of future values.

Chapter 1 introduces the main problem of the research and it describes a number of potential fields of application.

Chapter 2 defines some central concepts in time series analysis like autocorrelations and lagged variables. Autocorrelations can be conveniently organized into a correlation box. The chapter continues with a discussion of a number of popular time series models and techniques: the ARMA model, the state space model, dynamic factor analysis, canonical correlation analysis and graphic techniques.

Chapter 3 describes how optimal scaling can be embedded into cross-sectional multivariate analysis. The starting point forms the Gifi system of nonlinear multivariate analysis, which was developed in the Department of Data Theory at the University of Leiden. After introducing the relevant concepts and formulas, the chapter shows how a large number of multivariate techniques can be generalized to their optimal scaling equivalents by restricting an homogeneity analysis model.

Chapter 4 studies in what way optimal scaling can be combined with univariate time series analysis. It discusses the properties of three elementary filters. Subsequently the chapter presents generalizations of three popular forms of univariate autoregressive analysis.

Chapter 5 deals with multivariate rather than univariate time series. The chapter proposes generalizations of four multivariate techniques: intervention analysis, the Box-Tiao transformation, dynamic components analysis and multiset dynamic components analysis.

Chapter 6 integrates the optimal scaling techniques into one mathematical optimization problem and discusses a solution for an important subset of the problem, called the canonical class. The optimal scaling time series techniques of Chapter 4 and 5 are all special cases of this class.

Finally, Chapter 7 summarizes the main results and provides some suggestions for future research.

Postscript

I thank the following people and institutions for their help: Jos ten Berge, Dorly Deeg, Garmt Dijksterhuis, DSWO Press, Teije Euverman, Willem Heiser, Celine van Hoek, Leo van der Kamp, Eveline Kroezen, Jan de Leeuw, OP & P Tekstwerk, Johan Oud, Pieter Punter, Jan van Rijckevorsel, Ton Snijders, Rob Stobberingh and Tom Wansbeek.

OPTIMAL SCALING OF TIME SERIES

A time series consists of a succession of observations made at equidistant points in time or covering equal intervals of time. Most methods for studying the variability of time series are based upon the inclusion of lagged variables in the analysis, i.e. upon the same succession of observations shifted one or more time points backwards or forwards. When the observations are categorical it can be useful to apply optimal scaling of the observed categories. One of the elementary situations considered is optimal scaling of the row and column categories of a transition table, which gives the relative frequency of going from some particular state at one time point to some other or the same state at the next time point. A basic problem here is how to ensure that the optimal scaling of the row categories is equal to the optimal scaling of the column categories. Equating the two scalings is desirable, since we are dealing with two versions of the same empirical variable.
The problem of equating optimal quantifications is solved in this book for a much more general class of situations than the one described above. Various time series models are formulated as multivariate analysis problems with lagged variables. Combining this approach with optimal scaling techniques derived from the Gifi system of nonlinear multivariate analysis yields a broad class of methods for analyzing categorical time series. The text discusses seven useful options in detail. These are: lag-one predictor analysis, seasonal autoregression, multiple autoregression, intervention analysis, predictable components analysis, dynamic components analysis and multiset dynamic components analysis. Examples taken from psychology, sociology and economics illustrate the theoretical material. The appendix provides a complete listing of a computer program to carry out the analyses.

DSWO PRESS

ISBN 90 6695 040 4