Biplots and Triplots for exploring three mode data with an application to the investigation of the immune response to Bacille Calmette Guérin vaccine in HIV positive infants

Author: Darryn Williams
Supervisor: Associate Professor Sugnet Lubbe

Dissertation presented for the degree of
Master of Science

IN THE DEPARTMENT OF STATISTICAL SCIENCES

UNIVERSITY OF CAPE TOWN

May 11, 2013

Publication

I hereby grant the University free licence to publish this dissertation in whole or part in any format that the University deems fit.

Plagiarism Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is my own.
2. I have used the APA referencing guide for citation and referencing. Each contribution to, and quotation in this dissertation from the work(s) of other people has been contributed, and has been cited and referenced.
3. I know the meaning of plagiarism and declare that all of the work in the dissertation, save for that which is properly acknowledged, is my own.

Signature:

Date:

Acknowledgements

I would like to express my heartfelt thanks to the Harry Crossley Foundation as well as the Institute of Applied Statistics; without their financial support I would not have been able to pursue this research. To Associate Professor Sugnet Lubbe, your meticulous and unwaivering supervision made this research process enjoyable and enriching; thank you for all your effort. Your critique and suggestions proved invaluable throughout this research. I would also like to thank Mansoor *et al.* (2009) for allowing me to use their data throughout this research.

I also wish to thank my parents, Russel and Carole-Ann, for their support in this process. To Ashlynne Williams, thank you for lending an ear when I needed one. To Sheree Lang and Shannon Bernhardt, thank you for keeping my spirits high.

Abstract

Statistics is well acquainted with the use of exploratory techniques to reveal detail about the structure of the data. Graphical displays of data are key in this process, chief amongst them the scatterplot. However, a scatterplot is restricted to use with data comprising at most three variables. For higher dimensional data comprising $p > 3$ variables, the analogue to this display is the biplot. It is a plot that displays axes for all the variables that have been measured together with points to represent all the subjects in a low dimensional approximation of the p dimensional space. It is useful in that it affords the means to instantly assess correlations between variables as well as any groupings of the subjects that might be inherent in the data set.

This display is well developed for data comprising n samples and p variables. Such data sets are referred to as two mode data where the objects and variables represent modes. This dissertation is focused on using biplots for data of a different nature. More specifically, three mode data is considered. Generally this type of data comprises a number of objects on which a number of measurements have been made under different conditions. The modes are thus subjects, variables and conditions.

This dissertation is primarily concerned with exploring different techniques to use for the purpose of performing an exploratory data analysis on a three mode data set. More specifically, the focus is on the use of biplots for this purpose. The techniques considered include Principal Component Analysis biplots, Canonical Variate Analysis biplots, Common Principal Component Analysis as well as Generalised Orthogonal Procrustes Analysis. Tensor decomposition techniques are also explored and whilst the Tucker3 decomposition is used to construct biplots, the Parallel Factor Analysis model is shown to have undesirable properties for the construction of an exploratory plot. Tensor Singular Value Decomposition is thus discussed and used as a framework for the construction of a triplot, a plot that simultaneously displays all three modes comprising the data.

All these methods are applied to a longitudinal data set taken from Mansoor *et al.* (2009) in order to ascertain whether they are effective in untangling the relationships in the data but also whether they convey similar information about the data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Exploratory data analysis is undoubtedly one of the most useful tools in the realm of Statistics. Before embarking on a rigorous statistical analysis of any data set the statistician is taught to consult a myriad of graphical displays in order to develop a rudimentary understanding of various aspects of the data. This supports any rigorous analysis and serves to strengthen the researcher's intuition about the data, so that any results from formal modelling can be interrogated with knowledge of the data. Exploratory data analysis relies to a large extent on two graphical displays in the statistician's arsenal: the *scatterplot* and the *box and whisker* plot. These tools have proved invaluable in the task of untangling relationships in any particular data set. In the face of data comprising a large number of variables, the exploratory aspect of the analysis becomes cumbersome because of the large number of plots that must be interpreted. Interactions between subjects, variables as well as subjects and variables can only be understood by considering a number of different plots. A scatterplot provides a sense of the relationship between the two variables comprising the axes. In order to understand other relationships, more scatterplots must be considered. While the need for multiple scatterplots might not have traditionally been a consideration as data tended to comprise few variables, the new paradigm in Statistics concerns itself with large data sets (Efron, 2007). Multivariate data have become ubiquitous. What has now also become commonplace is data that are collected at different time points or under different conditions, and as such are deemed three mode data. Such data are often seen in the fields of Chemometrics and Psychometrics (Bro, 1997).

The biplot, so named because it allows observations and variables to be simultaneously displayed, was introduced by Gabriel (1971) and is the multivariate analogue to the scatterplot. It has proven itself to be a powerful tool for the purpose of understanding multivariate data from an exploratory perspective. It has also proved effective in making the exploratory task efficient, in that the biplot conveys a great deal of information about the data that would ordinarily be gleaned from a combination of scatterplots and box plots. Although this technique has enjoyed extensive use in the context of multivariate data, it has not been as popularly used for exploring three mode data. For this reason, this dissertation is concerned with exploring various techniques that can be used to produce biplots for three mode data, as well as considering the construction of a triplot. Each technique is applied to a longitudinal data set to assess its efficacy as an exploratory tool, as well as to determine whether the various techniques yield similar conclusions about the structures inherent in the data.

1.1 Background information

The common thread throughout this dissertation is Principal Component Analysis (PCA). It has proven invaluable for exploratory data analysis in a two mode multivariate context. The fact that exploratory tools for three mode data are not well developed motivated this research. Graphical tools are vital in Statistics and they are used primarily as exploratory or diagnostic tools. Kroonenberg (2008) dedicates a chapter to graphical displays for three mode methods but the emphasis is on aiding in the interpretation of parameter estimates, as well as considering graphical means to assess the validity of a chosen model. Only brief mention is made of exploratory tools for three mode data. The most comprehensive work on this subject is a paper by Kiers (2000a) which has a stronger exploratory emphasis than what is seen in Kroonenberg (2008). The literature in the area of exploratory analysis for three mode data is therefore somewhat lacking. This dissertation thus seeks to provide some methods for the express purpose of exploratory analysis and although techniques may seem disjointed, it is PCA that reveals itself constantly. The choice of multivariate longitudinal data for the application is born of the fact that this is a common form of three mode data. Longitudinal studies are common thus simple, comprehensive exploratory techniques will prove valuable. The methods discussed are not limited to longitudinal data however and could also be used on data comprising subject scores, for example on various intelligence tests under different conditions affecting concentration.

1.2 Objectives

The objectives of the research undertaken in this dissertation relate to building a sound understanding of the biplot construction and applying this to three mode data in order see how it fairs as an exploratory tool. More specifically the objectives can be summarised as follows:

1. Provide a comprehensive explanation of the theoretical foundations underpinning biplot construction. Focus will be on Principal Component Analysis as well as Canonical Variate Analysis Biplots;

2. Discuss the theoretical framework for two main classes of tensor decomposition techniques and exploring whether these techniques can be used to produce biplots. There is a brief discussion on triplot construction and interpreting some aspects of this plot;

3. Exploring the use of other multivariate techniques for the construction of biplots in a three mode data context;

4. Applying all these methods to a data set taken from Mansoor *et al.* (2009) in order to determine whether the different techniques convey similar information about the structure of the data. Careful consideration is given to the interpretation as well as any similarities and differences that arise in the plots.

It is argued that although different methods are to be used in the construction of the biplots, they will convey similar information about the structure of the data. These plots are used in an exploratory context and thus should lead to the researcher drawing similar conclusions about the data.

1.3 Scope and Limitations

Biplot methodology comprises a vast array of methods but this dissertation is focused on the analysis of asymmetric, linear biplots. Gower *et al.* (2011) provide an excellent introduction to the various aspects embodied in biplot methodology. The techniques that are considered in this dissertation are applied to a longitudinal data set only although they are not limited to use with such data. There is a vast body of literature pertaining to three mode data modelling including issues of dimensionality selection, interpretation of parameter estimates and diagnostic tools, however the fundamental topic of this dissertation is exploratory in spirit and so these issues are not considered here. Kroonenberg (2008) is an excellent reference for understanding these issues but they are not key in constructing and interpreting the biplots that lie at the heart of this dissertation. The methods discussed in this dissertation are largely limited to three mode profile data comprising subjects, variables and conditions or occasions. Specifically, the variable mode should comprise continuous measurements. Longitudinal data tends to fall into this category. A similar perspective to that of Kroonenberg (2008) is taken here where the time aspect of the data is used as an interpretational device rather than being explicitly considered as a modelling device. This is done primarily because the data are not being modelled but explored in the way that precedes rigorous statistical analysis.

1.4 Chapter layout

The dissertation is largely divided into two parts with the former considering simple multivariate techniques that can be considered two mode in nature and the latter looking at three mode techniques, with the final chapter considering tensor decomposition. The remainder of this dissertation is divided as follows: Chapter 2 provides a brief overview of three mode data, defining pertinent concepts that are needed for this dissertation. Chapter 3 discusses the Mansoor *et al.* (2009) data, giving some technical background and performing a traditional exploratory data analysis. Chapter 4 introduces biplots and considers the construction and interpretation of these plots. The basic tools required for biplot construction are discussed and then detail is given on how the marriage of these various aspects results in the PCA biplot. This chapter also constructs PCA biplots for the Mansoor data and discusses the interpretation and conclusions. Chapter 5 goes on to discuss how the grouped nature of the data can be included in the biplot construction process by considering Canonical Variate Analysis (CVA) biplots. This technique is applied to the matricised forms of the Mansoor data and the results discussed. Chapter 7 considers the use of Common Principal Component Analysis (CPC) and how this technique can be used to construct biplots. This is done from the perspective of the researcher having satisfied themselves that the CPC hypothesis is valid and now seeks to construct and interpret a biplot from the parameter estimates. This chapter is vital for an important novel development later on. Chapter 6 details how Generalised Orthogonal Procrustes Analysis (GOPA) can be used to combine into a single plot, the separate PCA biplots produced for each of the data sets comprising the three mode data. Chapter 8 discusses a novel development focused on representing separate CVA biplots in a single comprehensive plot. The method is applied to simulated data and then to the Mansoor data. Chapter

9 is distinctly different in that it explores tensor decomposition techniques and considers the two main classes of models that generalise PCA to three mode data. The Tucker model with orthogonality constraints is used to construct biplots, whilst the Parallel Factor Analysis (PARAFAC) model is discussed and reasons provided for why this is not a suitable model for the construction of an exploratory plot. Finally, a third decomposition technique is discussed and this is used to construct triplots (Araújo, 2009). These techniques are applied to the Mansoor data and the conclusions discussed. The dissertation draws to a close with Chapter 10, which comprises a summary of the conclusions reached throughout and makes a recommendation on the use of the various techniques.

1.5 Software

The implementation of the various methods discussed in this dissertation was performed in the commonly used programming environment *R* version 2.14 (R Development core team, 2011). The functions used to produce the actual biplots are taken from the package developed for use with Gower *et al.* (2011). Where necessary, modifications have been made to these functions. The implementation of some tensor decomposition techniques was performed in $Matlab$ using the N-way toolbox developed by Andersson and Bro (2000). All the code can be found in the appendix.

1.6 Notation

In order to provide a frame of reference for the reader's convenience, commonly used notation throughout the dissertation is defined here. The convention is that bold uppercase letters represent matrices and bold lowercase symbols represent column vectors. Also note that where convenient the Mansoor *et al.* (2009) data set will simply be referred to as the Mansoor data. Legends have been placed on some plots and apply in a similar fashion to those that do not have legends.

- n/N : refers to the number of subjects comprising the data.
- p/P : refers to the number of variables comprising the data.
- k/K : refers to the number of occasions comprising the data.
- g: number of groups into which the subjects are divided.

- $\boldsymbol{X}$: Data matrix with dimensions subjects × variables i.e. $n \times p$.
- $\mathcal{X}$: Order three tensor with dimensions subjects × variables × occasions i.e. $n \times p \times k$.
- $\| \boldsymbol{X} \|$: Frobenius norm.
- $\mathbf{1}$: column vector of ones with size determined by context.
- $tr(\boldsymbol{X})$: trace of the matrix $\boldsymbol{X}$.
- $\boldsymbol{X}_{(m)}$: Unfolding of $\mathcal{X}$ in the m^{th} mode.
- $\boldsymbol{\Psi}$: a $p \times p$ covariance matrix.

Chapter 2

Three way data

2.1 Introduction

Statistics is best acquainted with the analysis of data comprising "scores of subjects on a number of variables" (Kroonenberg, 2008, p. 146). This data can be arranged in a two mode regular rectangular array with rows representing subjects and columns representing variables. When data is collected at different time points or under different conditions, a third way is introduced. In a very crude sense, such data cannot be contained on a single index card but comprises a collection of index cards that must be contained in a box. A myriad studies produce data of this nature. An example can be drawn from child studies where the scores on a collection of variables such as intelligence, mass, age and physical fitness are measured for a number of children over a period of time. A second example comes in the form of measuring plant characteristics for different species of plants grown in different locations. It is important to remember that although three way data comes in different forms such as categorical, and rating scale data, the focus is on data comprising subjects with a number of continuous measurements taken under different conditions or at different occasions. For the remainder of this section, three way data will be thought of as comprising subjects, variables and time points for ease of explanation. This chapter is brief and introduces basic three mode terminology and the concept of matricisation.

2.2 Modes and ways of an array

Figure 2.1 is an illustration of a three-way data array. The entries along the vertical axis, denoted by index i, represent one entity, those along the horizontal axis, denoted by index j, represent another entity and those along

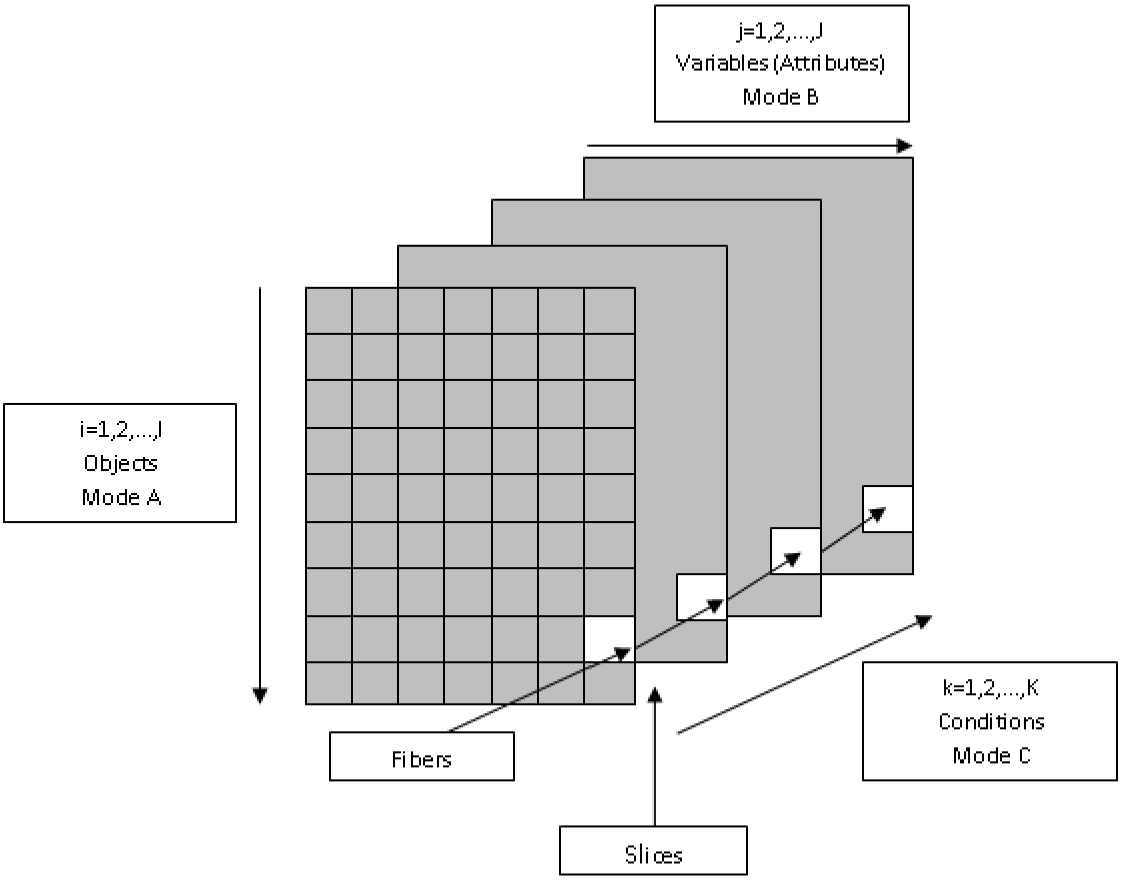


Figure 2.1: Illustration of a three way data set.

the depth axis, denoted by index k, represent yet another entity. Each separate entity is referred to as mode. More specifically, the word mode refers to the content of each of the ways so that each of the objects, variables and conditions comprising the data array can be thought of as modes.

Three-way data can be classified by mode in the following way: When three different entities occur in each of the ways then the data is referred to as three mode three way data. The examples provided at the beginning of this section could be classified as such. When the same entity occurs in two of the ways then the data is referred to as two mode three way data. A collection of correlation matrices for the same variables from several different samples is an example of data of this type where the samples and the variables define the two modes. One mode three way data thus refers to data where the same entity occurs in all three ways.

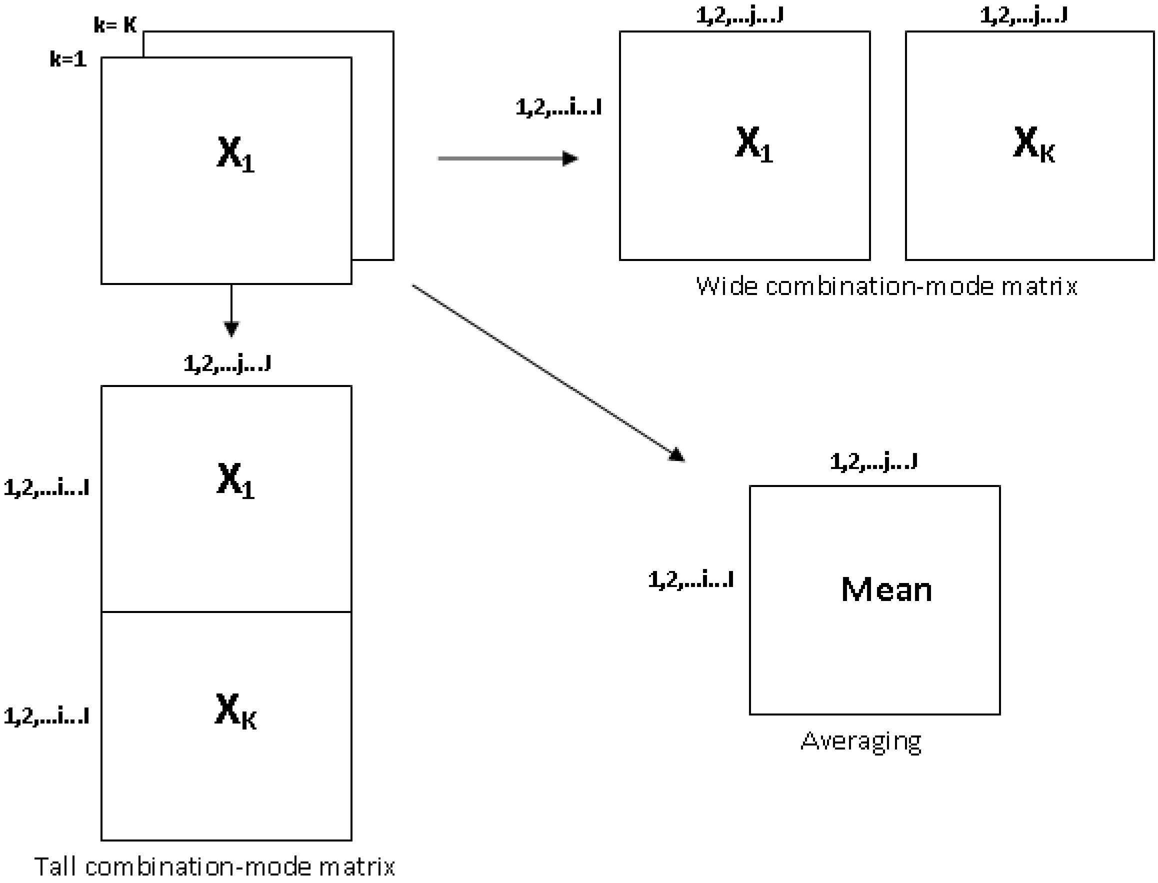


Figure 2.2: Illustration of matricising a three way data set.

2.3 Matricisation of three way data array

Before the advent of statistical methods capable of analysing three mode data, two mode methods were used for the purpose of analysis. The data had to be restructured in order to make it amenable to the application of two mode methods. One of three methods is most commonly used to restructure the data and all of these methods are illustrated in Figure 2.2. The first of these methods, referred to as flattening, requires that the three way data array be collapsed into a single matrix by removing one of the ways. This is usually done by averaging the subject scores across the depth axis or what is usually the time/condition mode. This results in any trends over time being lost. The process of matricisation, which describes the remaining restructuring methods, requires that two modes be combined by either placing each of the data matrices comprising the three way data array next to each other to from what is called a wide combination-mode matrix or by forming what is called a tall combination-mode matrix. Both of these are illustrated in Figure 2.2. In the former case, the variable and time modes are combined so that any relationships between the variables at different time points are neglected. In the latter case, the subject and variable modes are combined so that any similarities between an individual's score at different points in time are lost. These methods provide a means of simplifying the data so as to apply two-mode statistical methods however they are not without a price. Valuable information inherent in the data is lost by applying these methods and it must thus be done with caution when undertaking rigorous statistical analysis.

2.4 Conclusion

This chapter served to introduce the notion of three way data as well as pertinent concepts. Specifically, the concepts of modes and ways was introduced followed by the process of matricisation which is significant in later discussion.

Chapter 3

Mansoor data

3.1 Introduction

Mansoor *et al.* (2009) undertook an investigation to determine the Immune Response induced by Bacille Calmette Guérin (BCG) vaccine in HIV positive (HIV^+) patients. It is commonly used as a vaccine in sub-Saharan Africa and has been found to be particularly efficacious in protecting against Pulmonary Tuberculosis. BCG has been found to cause complications in patients, particularly an illness known as "BCGosis" which has a high level of morbidity in infants. The primary aim was to ascertain whether BCG induces what is thought to be the required immune response in HIV^+ infants to protect against TB. A secondary aim was to determine whether the immune strength induced in HIV-uninfected infants born to HIV-infected mothers was similar to that induced in HIV-unexposed and healthy infants.

3.2 Data Structure

Participants comprised infants born to HIV infected and uninfected mothers recruited from the Worcester region in the Western Cape from 2003 to 2006. All infants had been given the BCG vaccine at the date of birth and received an HIV test at age six weeks. Furthermore, the subjects were assigned to one of three groups. These groups were defined as follows: Group 1 comprised HIV infected infants, Group 2 comprised HIV exposed but uninfected infants and Group 3 comprised HIV unexposed and uninfected infants. The infants in group 2 were born to HIV infected mothers. Antiretroviral therapy was not made available at any point during the study with the implication that any immune strength was attributable to the BCG.

The study has a longitudinal aspect to it in that each of the infants were seen at 3 months, 6 months, 9 months and 12 months respectively. Blood samples and T-cell marker measurements were obtained at each occasion. To better understand the data, a brief digression into the realm of T-cells and cytokines is necessary.

According to Ibelgaufts (2009), T-Helper cells or T_h cells are a type of white blood cell that is instrumental in maximizing the ability of the immune system to launch an immune response. These cells express the protein CD4 on its surface and consequently are also referred to as CD4 cells. T_h cells are incapable of killing infected cells or any pathogens present in the body, but their importance can be attributed to the fact that they are responsible for secreting the cytokines that activate and direct the other immune cells (Ibelgaufts, 2009). These cells are activated when exposed to peptide antigens which are presented by Antigen Presenting Cells. Following activation, these cells divide and produce cytokines. Several distinct T_h cells have been identified, each secreting different cytokines to facilitate different immune responses. Most of these cells secrete only a single cytokine per cell (Karulin et al., 2000). The concern of the Mansoor investigation was the T_h 1 cell which is known to cause strong cellular immunity.

Cytokines produced by the immune system are a group of immune-modulatory proteins or immune-transmitters that are responsible for the modulation of, the reproduction and bioactivity of the immune cells (Ibelgaufts, 2009). In effect, these cytokines facilitate communication between the immune system and other cell types. Several different cytokines have been identified and the T_h 1 cells produce a number of these cytokines, commonly referred to as Type 1 cytokines. These include Interferon-γ (IF-γ), Tumour Necrosis Factor-α (TNF-α) and Interleukin-2 (IL-2). Each of these cytokines has a specific function. For example, it is understood that IF-γ and TNF-α are vital in the prevention of viral replication amongst other things where as IL-2 is pivotal in cell replication.

This brief digression affords the means to state clearly how the data is used to answer the questions put forward in Mansoor *et al.* (2009) regarding the immune response and immune strength amongst the groups in the study. Recall that the primary aim of the study was to determine whether BCG induces what is thought to be the required immune response for TB protection in HIV infected infants. According to Mansoor *et al.* (2009), what is widely thought to be essential in the protection against TB is the T_h 1 cell cytokine response comprising TNF-α, IL-2 and IF-γ. T_h 1 cells that coexpress all 3 these cytokines, polyfunctional T-cells, are thought to be good indicators of quality of immune response and evidence gathered from TB vaccine studies done on animals has shown that these polyfunctional cells are associated with protection against TB. It is for this reason that these T cell markers were the primary measurement in this study.

	Response	Presence of Cytokines			Number of Cytokines Expressed
		IF-γ	IL-2	TNF-α	
V2	cd4_ifngpil2mtnfm	Yes	No	No	1
V6	cd4_ifngmil2ptnfm	No	Yes	No	1
V5	cd4_ifngmil2mtnfp	No	No	Yes	1
V3	cd4_ifngpil2ptnfm	Yes	Yes	No	2
V1	cd4_ifngpil2mtnfp	Yes	No	Yes	2
V7	cd4_ifngmil2ptnfp	No	Yes	Yes	2
V4	c d4_ifngpil2ptnfp	Yes	Yes	Yes	3

Table 3.1: Combination of cytokines with the measure of polyfunctionality.

Table 3.1 indicates the combination of cytokines together with the measure of "polyfunctionality", the number of cytokines expressed by the T_h 1 cells, observed in the T cell marker data set collected. The strength of the cell's polyfunctionality is measured by the number of cytokines coexpressed. In the context of this study, at the very most three cytokines can be coexpressed. For the remainder of the dissertation the labelling in column 1 of Table 3.1 will be used to refer to the the T cell markers.

Table 3.2 indicates how the the number of participants in the study changed over time. It is evident that the number of infants in all three groups of interest declined possibly due to death or withdrawal from the study for other reasons. For the purpose of this dissertation all missing observations were removed so that the final data set used contained 29 observations at each occasion. Of the 29 observations, 3 comprised the first group, 10 comprised the second group and 16 comprised the third group.

A pertinent question is whether the data collected in this study can be classified as three-way data and thus amenable to modelling by means of three-way techniques. It is clear that this is indeed a three-way data set in which the infants participating in the study represent one way, the T cell markers measured as described in Table 3.1 represents the second way and the measurements taken at different time points represents the third way.

Group	Time Point (months)				Total
	3	6	9	12	
HIV infected	20	12	8	5	45
HIV exposed and uninfected	25	17	15	12	69
HIV unexposed and uninfected	23	22	22	19	86

Table 3.2: Number of subjects at each occasion.

3.3 Exploratory Data Analysis

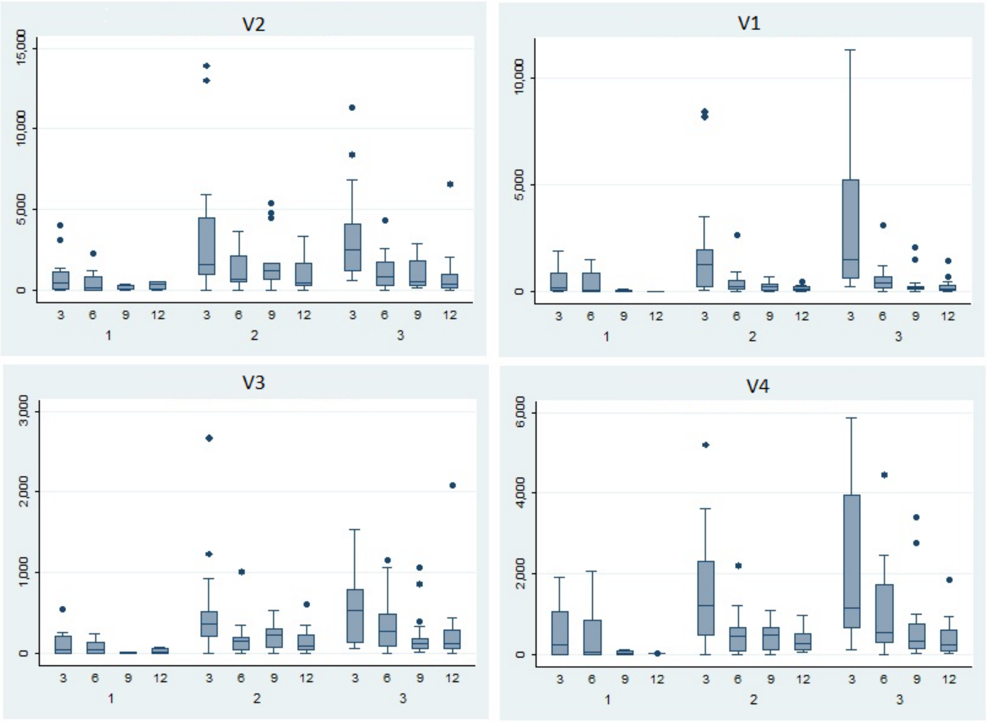


Figure 3.1: Boxplots representing four variables comprising data.

Data exploration is key in any statistical analysis and the boxplot and scatterplot are the traditional tools used for this task. When data comprises a large number of variables a myriad of these plots are needed to tease out the data structure. A brief rudimentary exploration of the data by means of boxplots serves to sketch a preliminary image of the structure inherent in the data. Biplots are fundamentally exploratory in spirit and this section is included in order to get a sense of the data but also to show that the biplot serves well to encompass most of the information that is gleaned from the separate plots used for the exploratory analysis.

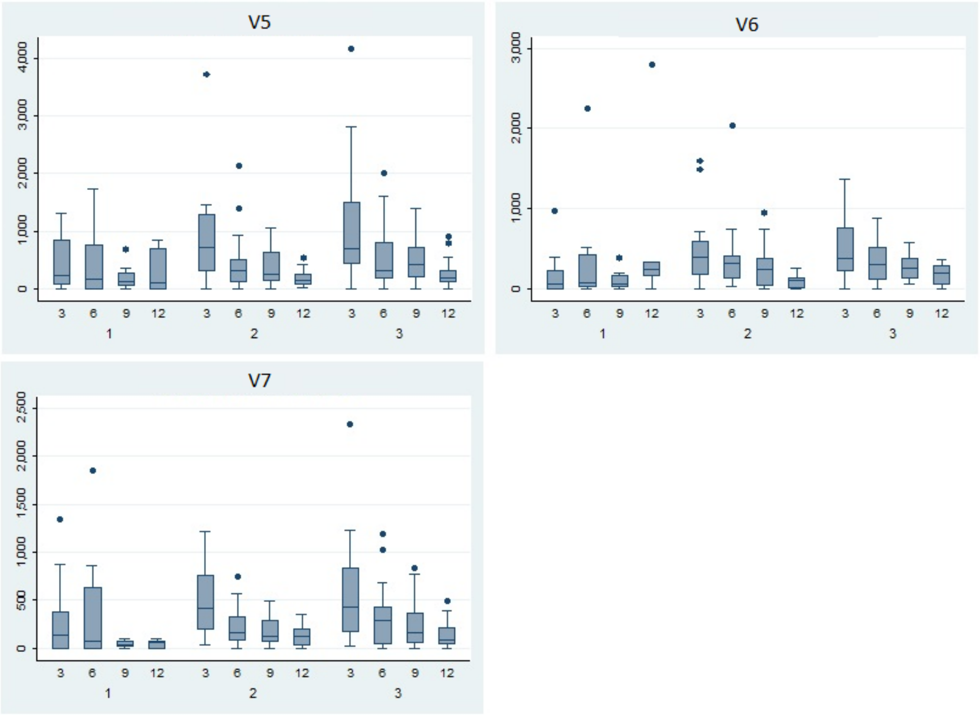


Figure 3.2: Boxplots for the remaining three variables comprising data.

Figures 3.1 and 3.2 represent boxplots constructed for the data. Each variable is represented and for each variable the grouped nature of the data is considered in the plot. It is immediately clear that there was a general decrease in variation in the data across occasion for all groups. This reduction was generally more distinct for infants comprising the HIV-uninfected groups across most variables. The group comprising the HIV^{+} infants demonstrated behaviour that was contrary to this trend particularly in Figure 3.1. The infants comprising the uninfected and unexposed group, group 3, showed the most variation on $V2$ and $V5$ in Figure 3.1, particularly at occasion 3. Another striking feature is the fact that the median score for the infants in groups 2 and 3 are similar on all variables across occasion. Furthermore, there was a general decreasing trend in the score on most variables across occasion. The scores for the subjects comprising group 1 was seen to be relatively smaller than the scores for the other two groups. These plots thus seem to indicate that BCG did not have the same effect in terms of producing the required immune response in each of the groups as opined by Mansoor *et al.*

(2009) with the uninfected groups scoring relatively higher on the variable coexpressing all three cytokines, $V4$. In fact, the infants comprising the first group seemed to obtain relatively lower scores on most of the variables with the difference becoming less marked over time. If the biplot can be used to condense this information into a single display it will be quite convenient. The next aspect to consider is that of the correlation between the variables and how that evolves over time. Although the pertinent information for the investigative questions posed by Mansoor *et al.* (2009) is contained in Figures 3.1 and 3.2, it is worth examining the associations between variables if only to see how well this information is captured in a biplot.

$$\begin{array}{cc} & \begin{array}{ccccccc} V1 & V2 & V3 & V4 & V5 & V6 & V7 \end{array} \\ \begin{array}{c} V1 \\ V2 \\ V3 \\ V4 \\ V5 \\ V6 \\ V7 \end{array} & \begin{pmatrix} 1 & 0.82 & 0.77 & 0.90 & 0.42 & 0.19 & 0.26 \\ 0.82 & 1 & 0.76 & 0.69 & 0.23 & 0.33 & 0.06 \\ 0.77 & 0.76 & 1 & 0.82 & 0.29 & 0.31 & 0.24 \\ 0.90 & 0.69 & 0.82 & 1 & 0.47 & 0.19 & 0.47 \\ 0.42 & 0.23 & 0.29 & 0.47 & 1 & 0.64 & 0.76 \\ 0.19 & 0.33 & 0.31 & 0.19 & 0.64 & 1 & 0.55 \\ 0.26 & 0.06 & 0.24 & 0.47 & 0.76 & 0.54 & 1 \end{pmatrix} \end{array} \quad \begin{array}{c} \begin{array}{ccccccc} V1 & V2 & V3 & V4 & V5 & V6 & V7 \end{array} \\ \begin{pmatrix} 1 & 0.58 & 0.64 & 0.79 & 0.70 & 0.69 & 0.48 \\ 0.58 & 1 & 0.73 & 0.53 & 0.23 & 0.63 & 0.18 \\ 0.64 & 0.73 & 1 & 0.70 & 0.35 & 0.61 & 0.20 \\ 0.79 & 0.537 & 0.70 & 1 & 0.79 & 0.46 & 0.67 \\ 0.70 & 0.23 & 0.35 & 0.79 & 1 & 0.43 & 0.83 \\ 0.69 & 0.63 & 0.61 & 0.46 & 0.43 & 1 & 0.20 \\ 0.48 & 0.18 & 0.20 & 0.67 & 0.83 & 0.20 & 1 \end{pmatrix} \end{array}$$

$$\begin{array}{cc} \begin{array}{c} V1 \\ V2 \\ V3 \\ V4 \\ V5 \\ V6 \\ V7 \end{array} & \begin{pmatrix} 1 & 0.35 & 0.87 & 0.95 & 0.43 & 0.33 & 0.57 \\ 0.35 & 1 & 0.45 & 0.34 & 0.09 & 0.36 & 0.22 \\ 0.87 & 0.45 & 1 & 0.84 & 0.25 & 0.37 & 0.52 \\ 0.95 & 0.34 & 0.84 & 1 & 0.35 & 0.28 & 0.73 \\ 0.43 & 0.09 & 0.25 & 0.35 & 1 & 0.51 & 0.36 \\ 0.33 & 0.36 & 0.37 & 0.28 & 0.51 & 1 & 0.49 \\ 0.57 & 0.22 & 0.52 & 0.73 & 0.36 & 0.491 & \end{pmatrix} \end{array} \quad \begin{pmatrix} 1 & 0.80 & 0.91 & 0.90 & 0.23 & 0.13 & 0.51 \\ 0.80 & 1 & 0.91 & 0.75 & 0.20 & 0.09 & 0.27 \\ 0.91 & 0.91 & 1 & 0.87 & 0.11 & 0.01 & 0.30 \\ 0.90 & 0.75 & 0.87 & 1 & 0.11 & 0.17 & 0.53 \\ 0.23 & 0.20 & 0.11 & 0.11 & 1 & 0.39 & 0.17 \\ 0.13 & 0.09 & 0.01 & 0.17 & 0.39 & 1 & 0.50 \\ 0.51 & 0.27 & 0.30 & 0.53 & 0.17 & 0.50 & 1 \end{pmatrix}$$

Figure 3.3: Correlation matrices for each occasion with occasion 1 top left, continuing clockwise.

Figure 3.3 illustrates the correlation matrices for the seven variables comprising the Mansoor data across occasion. $V1$ and $V4$ show a particularly strong correlation across occasion. $V1$ and $V3$ as well as $V3$ and $V4$ display similar behaviour across occasion. $V5$ and $V7$ show strong correlation at occasions 1 and 2 with a marked reduction in correlation at occasions 3 and 4. $V2$ and $V7$ display relatively weak correlation across occasion and the same can be said for $V3$ and $V5$. It is possible to continue in this vein in order to better understand the relationships between variables and how this evolves over time. In order to hail the biplot as an efficacious tool for exploratory data analysis, it should be possible to see some of these relationships.

3.4 Conclusion

This Chapter briefly detailed the aims of the Mansoor *et al.* (2009) investigation and digressed into a discussion on cytokines in order to facilitate the understanding of these aims. An exploratory data analysis was undertaken in order to gain a sense of the nature of the data set. This was done primarily to see whether the biplot, when used later on, will convey similar information regarding the nature of the data.

Chapter 4

Biplots

4.1 Introduction

As previously mentioned, data visualisation forms an integral part of the statistical analysis process. It affords the means to glean pertinent information about the data before delving into the more technical aspects of the analysis. The scatterplot is one graphical tool available for the purpose of data visualisation. This is a mathematical diagram which make use of Cartesian co-ordinates in order to show the values for two or three variables for a given set of data. Figure 4.1 is an illustration of a scatterplot. The axes are orthogonal and each represents a specific variable whilst the sample is represented as a collection of points, the position of each being determined by the values of the variables on the horizontal and vertical axes respectively. Naturally the diagram is two-dimensional, a dimension for each variable. Primarily it conveys information about the strength of the relationship between the variables. In this case Figure 4.1 conveys a strong positive linear relationship between $V1$ and $V2$. It also affords the means to make a statement about the magnitude of the variation for the variables represented by examining the spread of the data in the direction of each axis. Since the data are relatively dispersed in the direction of $V2$, it suggests that $V2$ displays large variation. When samples comprising three variables are collected such an accurate representation remains possible since three dimensional diagrams can be accurately depicted in two dimensions with modern computer graphics. However, in the realm of higher dimensions this becomes difficult. The question is thus how multivariate data comprising more than three variables can be visualised in a similar fashion. Biplots lie at the heart of the answer to this enigma.

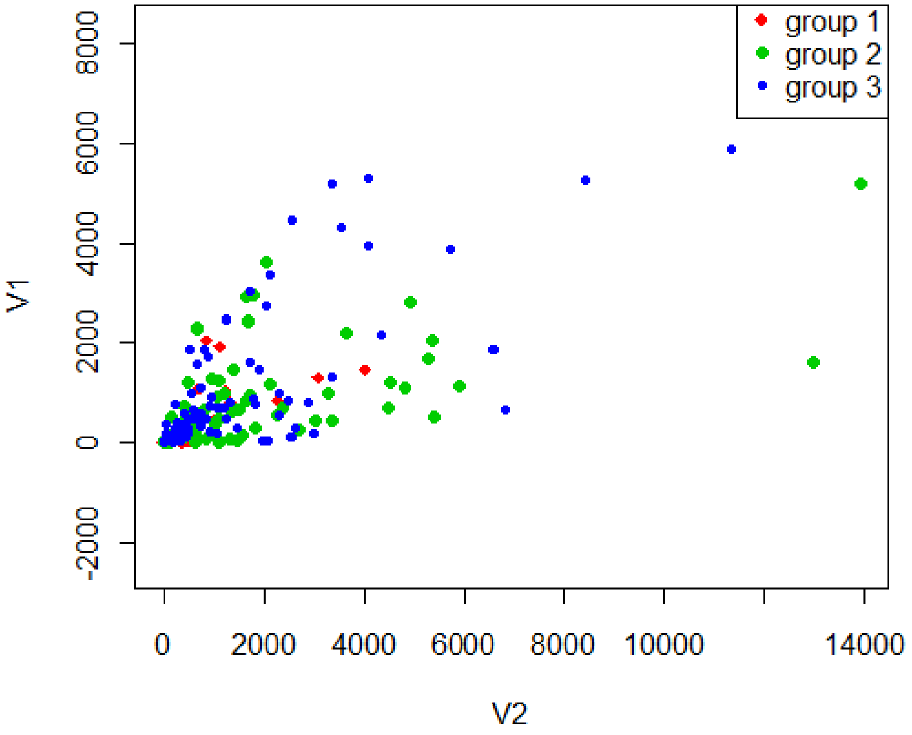


Figure 4.1: Scatterplot for variables one and two at time point three

Kroonenberg (2008) provided an eloquent description of the biplot as "allow[ing] for the analysis of two-way interaction in a table of n objects and p variables such that systematic patterns between rows, between columns and between rows and columns can readily be assessed and evaluated" (p. 492). The prefix *bi* is not a reference to the idea that these plots most often occur in two dimensions but rather to the fact that they allow for the simultaneous display of both the rows and columns of the data matrix $\boldsymbol{X}$. It is possible to distinguish between *symmetric* and *asymmetric* biplots. In the former instance, the biplots provide information on data matrices comprising sample units and variables and as such the rows and columns of the matrix are not interchangeable. Symmetric biplots provide information on two-way tables in which rows and columns are interchangeable since this would have no effect on the information contained in the table. Attention is focused on the construction of *asymmetric* biplots. The next section discusses the construction and interpretation of the most fundamental biplot, the Principal Component Analysis (PCA) biplot.

4.2 Basic tools for constructing a biplot

There are a number of mathematical tools that are vital in the process of constructing a biplot and more specifically a PCA biplot. This section offers a brief discussion of each and begins with the Singular Value Decomposition (SVD), moves on to the Eckart-Young Theorem and finally discusses Huygen's Principle.

4.2.1 Singular Value Decomposition

Suppose that a two way data matrix $\boldsymbol{X}$ comprising information of n objects on p variables with more objects than variables ($n > p$) has been collected. This data matrix $\boldsymbol{X}$ can be decomposed as:

$$\boldsymbol{X} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{\Sigma}}\tilde{\boldsymbol{V}}^{'},$$

where $\tilde{\boldsymbol{U}}_{n\times n}$ and $\tilde{\boldsymbol{V}}_{p\times p}$ are orthogonal matrices and $\tilde{\boldsymbol{\Sigma}}$ is an $n\times p$ matrix with singular values σ_j where $j = 1, \ldots, s$ are arranged in decreasing magnitude on the principle diagonal. The value s is equal to the rank of the matrix $\boldsymbol{X}$. Define

$$\tilde{\boldsymbol{\Sigma}} = \begin{array}{cc} & \begin{array}{cc} s & p-s \end{array} \\ \begin{array}{c} s \\ n-s \end{array} & \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \end{array}, \tag{4.1}$$

so that $\boldsymbol{\Sigma}$ is a diagonal $s \times s$ matrix with the singular values on its diagonal. $\tilde{\boldsymbol{\Sigma}}$ is referred to as a rectangular diagonal matrix and (4.1) makes it clear why this is the case. The column vectors comprising $\tilde{\boldsymbol{U}}$ are the n orthogonal eigenvectors of the matrix $\boldsymbol{X}\boldsymbol{X}'$ referred to as the left singular vectors and the columns comprising $\tilde{\boldsymbol{V}}$ are the p orthogonal eigenvectors of the matrix $\boldsymbol{X}'\boldsymbol{X}$ and are referred to as the right singular vectors. Both matrices are orthonormal. Gower *et al.* (2011) show that by defining matrices $\boldsymbol{U}_{n\times s}$ and $\boldsymbol{V}_{p\times s}$ comprising the first s columns of $\tilde{\boldsymbol{U}}$ and $\tilde{\boldsymbol{V}}$ respectively makes it possible to represent the SVD of $\boldsymbol{X}$ as

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{'}. \tag{4.2}$$

Note that the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthonormal. It is also pertinent to note that the decomposition of the data matrix $\boldsymbol{X}$ can be represented in summation notation as

$$x_{ij} = \sum_{t=1}^{s} \sigma_t u_{it} v_{jt}, \tag{4.3}$$

implying that s terms are generally required in order to reproduce the data matrix $\boldsymbol{X}$ perfectly. This notation becomes useful later on when the discussion turns to the actual construction of the biplot.

4.2.2 Eckart-Young Theorem

As discussed in section 4.1 , when data matrices comprises more than three variables, perfect graphical representation akin to a simple scatterplot is problematic. Failing the ideal of presenting a data set perfectly, the best option is to find a low dimensional approximation of the data matrix that lends itself to graphical display. The Eckart-Young theorem (Eckart and Young, 1936) provides a means to determine the best r-dimensional least squares approximation of the data matrix $\boldsymbol{X}$ where the value of r is usually 2. Formally, the Eckart-Young Theorem can be stated as follows

Theorem 4.2.1. *Given an $n \times p$ matrix $\boldsymbol{X}$ with a specific rank s, $\boldsymbol{X}$ can be approximated by an $n \times p$ matrix $\hat{\boldsymbol{X}}_{[r]}$ with rank r such that $r \leq s$. The approximation is based on the minimisation of the Frobenius Norm*

$$\|\boldsymbol{X} - \hat{\boldsymbol{X}}_{[r]}\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{p}(x_{ij} - \hat{x}_{ij})^2} = tr((\boldsymbol{X} - \hat{\boldsymbol{X}}_{[r]})'(\boldsymbol{X} - \hat{\boldsymbol{X}}_{[r]}))^{\frac{1}{2}},$$

under the constraint that rank($\hat{\boldsymbol{X}}_{[r]}$)=$r$ and the solution is given by the Singular Value Decomposition of $\boldsymbol{X}$

$$\hat{\boldsymbol{X}}_{[r]} = \boldsymbol{U}\boldsymbol{\Sigma}_{[r]}\boldsymbol{V}',$$

where $\boldsymbol{\Sigma}_{[r]}$ replaces the $s - r$ smallest singular values on the diagonal of $\boldsymbol{\Sigma}$ with 0.

Proof. The aim is to minimise $\|\boldsymbol{X} - \hat{\boldsymbol{X}}_{[r]}\|_F$ subject to the constraint that rank($\hat{\boldsymbol{X}}_{[r]}$)=$r$. Suppose that the Singular Value Decomposition of $\boldsymbol{X}$ is $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}'$ so that $\boldsymbol{\Sigma} = \boldsymbol{U}'\boldsymbol{X}\boldsymbol{V}$. The Frobenius Norm is unitarily invariant; that is for any matrix $\boldsymbol{A}$, $\|\boldsymbol{A}\|_F = \|\boldsymbol{U}\boldsymbol{A}\boldsymbol{V}\|_F$ where $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary matrices. A real matrix $\boldsymbol{U}$ is deemed unitary if $\boldsymbol{U}\boldsymbol{U}' = \boldsymbol{U}'\boldsymbol{U} = \boldsymbol{I}$. The property of unitary invariance implies that minimising $\|\boldsymbol{X} - \hat{\boldsymbol{X}}_{[r]}\|_F$ is equivalent to minimising $\|\boldsymbol{\Sigma} - \boldsymbol{U}'\hat{\boldsymbol{X}}_{[r]}\boldsymbol{V}\|_F$.

It is clear that since $\boldsymbol{\Sigma}$ is an $s \times s$ diagonal matrix, $\boldsymbol{U}'\hat{\boldsymbol{X}}_{[\boldsymbol{r}]}\boldsymbol{V}$ must also be diagonal in order to minimise the Frobenius norm. Define this diagonal matrix to be $\boldsymbol{S} = diag(s_i)$ for $i = 1, \dots, s$ so that $\boldsymbol{U}'\hat{\boldsymbol{X}}_{[\boldsymbol{r}]}\boldsymbol{V} = \boldsymbol{S}$ and $\hat{\boldsymbol{X}}_{[r]}$=$\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}'$. The problem thus reduces to finding the $\min_{s_i} \|\boldsymbol{\Sigma} - \boldsymbol{S}\|_F = min_{s_i}(\sum_{i=1}^{s}(\sigma_i - s_i)^2)^{\frac{1}{2}}$.

Due to the rank constraint the above expression has minimum

$$\min_{s_i}(\sum_{i=1}^{r}(\sigma_i - s_i)^2 + \sum_{i=r+1}^{s}(\sigma_i)^2)^{\frac{1}{2}} = \sqrt{\sum_{i=r+1}^{n}\sigma_i^2},$$

when $\sigma_i = s_i$ for $i = 1, \ldots, r$ and the corresponding singular vectors are the same as those for the matrix $\boldsymbol{X}$. □

Define the $s \times s$ matrix $\boldsymbol{J}$ as

$$\begin{pmatrix} \boldsymbol{I}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \tag{4.4}$$

where $\boldsymbol{I}_r$ is an $r \times r$ identity matrix. Note that $\boldsymbol{J}^2 = \boldsymbol{J}$ as well as that diagonal matrices commute. $\hat{\boldsymbol{X}}_{[r]}$ can be written as $\boldsymbol{U}(\boldsymbol{\Sigma J})\boldsymbol{V}^{'}$. Due to the properties mentioned, this expression can be represented as $(\boldsymbol{UJ})\boldsymbol{\Sigma}(\boldsymbol{VJ})^{'}$. The final $s - r$ columns of $\boldsymbol{UJ}$ and $\boldsymbol{VJ}$ comprise zeros though the matrices have dimension $n \times s$ and $p \times s$ respectively. The usefulness of this expression will become apparent later.

4.2.3 Huygen's Principle

At this point it is necessary to merely state and prove Huygen's Principle. It will become apparent at a later stage why this Principle is in fact so important in the process of constructing a biplot.

Result 4.2.1. *Let $\boldsymbol{c} \,\epsilon\, \Re^{\boldsymbol{p}}$ and $\boldsymbol{X}$ an $n \times p$ matrix then the sum of squares*

$$\|\boldsymbol{X} - \mathbf{1}\boldsymbol{c}^{'}\| = \|\boldsymbol{X} - \frac{1}{n}\mathbf{1}\mathbf{1}^{'}\boldsymbol{X}\| + n\|\mathbf{1}\boldsymbol{X}^{'} - \boldsymbol{c}^{'}\|,$$

is minimized when $\boldsymbol{c}^{'} = \frac{1}{n}\mathbf{1}^{'}\boldsymbol{X}$. More succinctly, the sum of squares about the mean is smaller than it is about any other point.

Proof. First consider the L.H.S

$$\begin{aligned}
\|\boldsymbol{X} - \mathbf{1}\boldsymbol{c}^{'}\| &= tr[(\boldsymbol{X} - \mathbf{1}\boldsymbol{c}^{'})(\boldsymbol{X} - \mathbf{1}\boldsymbol{c}^{'})^{'}] \\
&= tr[\boldsymbol{X}\boldsymbol{X}^{'} - \boldsymbol{X}\boldsymbol{c}\mathbf{1}^{'} - \mathbf{1}\boldsymbol{c}^{'}\boldsymbol{X}^{'} + \mathbf{1}\boldsymbol{c}^{'}\boldsymbol{c}\mathbf{1}^{'}] \\
&= tr[\boldsymbol{X}\boldsymbol{X}^{'} - \mathbf{1}^{'}\boldsymbol{X}\boldsymbol{c} - \boldsymbol{c}^{'}\boldsymbol{X}^{'}\mathbf{1} + \mathbf{1}^{'}\mathbf{1}\boldsymbol{c}^{'}\boldsymbol{c}] \\
&= tr[\boldsymbol{X}\boldsymbol{X}^{'} - \mathbf{1}^{'}\boldsymbol{X}\boldsymbol{c} - \boldsymbol{c}^{'}\boldsymbol{X}^{'}\mathbf{1} + n\boldsymbol{c}^{'}\boldsymbol{c}].
\end{aligned}$$

Now consider the expression on the R.H.S of the equation in Result 4.2.1.

$$\begin{aligned}
&\|\boldsymbol{X} - \frac{1}{n}\mathbf{1}\mathbf{1}^{'}\boldsymbol{X}\| + n\,\|\mathbf{1}\boldsymbol{X}^{'} - \boldsymbol{c}^{'}\| \\
&\quad = tr[(\boldsymbol{X} - \frac{1}{n}\underline{\mathbf{1}}\ \underline{\mathbf{1}}^{'}\boldsymbol{X})(\boldsymbol{X} - \frac{1}{n}\underline{\mathbf{1}}\ \underline{\mathbf{1}}^{'}\boldsymbol{X})^{'} + n(\underline{\mathbf{1}}\boldsymbol{X}^{'} - \underline{\boldsymbol{c}}^{'})(\underline{\mathbf{1}}\boldsymbol{X}^{'} - \underline{\boldsymbol{c}}^{'})^{'}] \\
&\quad = tr[\boldsymbol{X}\boldsymbol{X}^{'} - \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^{'}\mathbf{1}\mathbf{1}^{'} - \frac{1}{n}\mathbf{1}\mathbf{1}^{'}\boldsymbol{X}\boldsymbol{X}^{'} + \frac{1}{n^2}\mathbf{1}\mathbf{1}^{'}\boldsymbol{X}\boldsymbol{X}^{'}\mathbf{1}\mathbf{1}^{'} \\
&\qquad\quad + \frac{1}{n}\mathbf{1}^{'}\boldsymbol{X}\boldsymbol{X}^{'}\mathbf{1} - \boldsymbol{c}^{'}\boldsymbol{X}^{'}\mathbf{1} - \mathbf{1}^{'}\boldsymbol{X}\boldsymbol{c} + n\boldsymbol{c}^{'}\boldsymbol{c}] \\
&\quad = tr[\boldsymbol{X}\boldsymbol{X}^{'} - \frac{2}{n}\mathbf{1}^{'}\boldsymbol{X}\boldsymbol{X}^{'}\mathbf{1} + \frac{2}{n}\mathbf{1}^{'}\boldsymbol{X}\boldsymbol{X}^{'}\mathbf{1} - \boldsymbol{c}^{'}\boldsymbol{X}^{'}\mathbf{1} - \mathbf{1}^{'}\boldsymbol{X}\boldsymbol{c} + n\boldsymbol{c}^{'}\boldsymbol{c}] \\
&\quad = tr[\boldsymbol{X}\boldsymbol{X}^{'} - \mathbf{1}^{'}\boldsymbol{X}\boldsymbol{c} - \boldsymbol{c}^{'}\boldsymbol{X}^{'}\mathbf{1} + n\boldsymbol{c}^{'}\boldsymbol{c}].
\end{aligned}$$

This proves equality and inspecting the R.H.S reveals that setting $\boldsymbol{c}^{'} = \frac{1}{n}\mathbf{1}^{'}\boldsymbol{X}$ results in $n\,\|\mathbf{1}\boldsymbol{X}^{'} - \boldsymbol{c}^{'}\| = 0$. This minimises the R.H.S and implies that in order to obtain a minimum, $\boldsymbol{c}^{'} = \frac{1}{n}\mathbf{1}^{'}\boldsymbol{X}$. □

4.2.4 Factorisation of data matrix

This section seeks to succinctly express the ideas put forward by Gabriel (1971). More specifically, it introduces the notion that any matrix can be factorised and then considers how that factorisation is used in the process of constructing a biplot. The first result is a simple one and pertains to the factorisation of any $n \times p$ matrix.

Result 4.2.2. *Any $n \times p$ matrix $\boldsymbol{X}$ of rank r can be factorised as*

$$\boldsymbol{X} = \boldsymbol{G}\boldsymbol{H}^{'},$$

into an $n \times r$ matrix $\boldsymbol{G}$ and an $p \times r$ matrix $\boldsymbol{H}$, both necessarily of rank r. This factorisation is not unique (Rao, 1965).

Considering each element in the data matrix $\boldsymbol{X}$, Result 4.2.2 can be written as

$$x_{ij} = \boldsymbol{g_i^{'}}\boldsymbol{h_j},$$

where x_{ij} is the element in the i^{th} row and j^{th} column of the data matrix $\boldsymbol{X}$, $\boldsymbol{g_i}$ is the i^{th} row of the matrix $\boldsymbol{G}$ and $\boldsymbol{h_j}$ is the j^{th} row of the matrix $\boldsymbol{H}$. Each of the vectors $\boldsymbol{g_1}, \ldots, \boldsymbol{g_n}$ are assigned to the rows of $\boldsymbol{X}$, one for each row and are termed the row effects. Similarly each of the vectors $\boldsymbol{h_1}, \ldots, \boldsymbol{h_p}$ are assigned to the columns of $\boldsymbol{X}$, one for each column and are termed the column effects (Gabriel, 1971). Each of these vectors are of order r thus providing a means to represent the matrix $\boldsymbol{X}$ in the r-space by means of these $n+p$ vectors. Without loss of generality, the assumption is that all matrices $\boldsymbol{X}$ will be of rank two.

When considering a matrix $\boldsymbol{X}$ of rank two, the row and column effects are all vectors comprising two elements. As a result, these $n+p$ vectors can be plotted in the plane and each of the np elements of the matrix $\boldsymbol{X}$ can be represented by the inner products of the corresponding row and column effects. These plots are extremely useful in visually assessing the structure of the data matrix. The following example is taken from Gabriel (1971) and will emphasise an apparent obstacle in the graphical representation of a matrix of rank two.

Consider the following matrix with the associated factorisation:

$$\boldsymbol{X} = \begin{pmatrix} 2 & 2 & -4 \\ 2 & 1 & -3 \\ 0 & -1.5 & 1.5 \\ -1 & -0.5 & 1.5 \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ 2 & 1 \\ 0 & -1.5 \\ -1 & -0.5 \end{pmatrix} \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}.$$

An alternative factorisation of the matrix $\boldsymbol{X}$ is as follows:

$$\begin{pmatrix} 2 & -4 \\ 0 & -1 \\ -3 & 4.5 \\ 0 & 0.5 \end{pmatrix} \begin{pmatrix} -3 & -1 & 4 \\ -2 & -1 & 3 \end{pmatrix}.$$

In the first factorisation of the matrix $\boldsymbol{X}$, the row effects are $\boldsymbol{g_1}$ = (2,2), $\boldsymbol{g_2}$ = (2,1), $\boldsymbol{g_3}$ = (0,-1.5), $\boldsymbol{g_4}$ = (-1,-0.5) and the column effects are $\boldsymbol{h_1}$ = (1,0), $\boldsymbol{h_2}$ = (0,1), $\boldsymbol{h_3}$ = (-1,-1). These vectors can be plotted on the plane affording a visual appraisal of the structure of the data and this is shown in Figure 4.2.

The actual interpretation of these graphical representations will be deferred but what is poignant is the fact that the two possible factorisations produce disparate plots for the same matrix $\boldsymbol{X}$ as is evident from a quick study of the two diagrams in Figure 4.2. The disparity is evidence of the non-uniqueness of the factorisation of the matrix $\boldsymbol{X}$ and results from the fact that the factorisation stated in Result 4.2.2 can be replaced by

$$\boldsymbol{X} = (\boldsymbol{GR'})(\boldsymbol{HR^{-1}})' = \boldsymbol{GR}'(\boldsymbol{R'})^{-1}\boldsymbol{H'} = \boldsymbol{GH'}, \qquad (4.5)$$

for any non-singular matrix $\boldsymbol{R}$. Equation (4.5) makes it clear that although the matrices $\boldsymbol{G}$ and $\boldsymbol{H}$ are transformed to $\boldsymbol{GR'}$ and $\boldsymbol{HR^{-1}}$ respectively, the

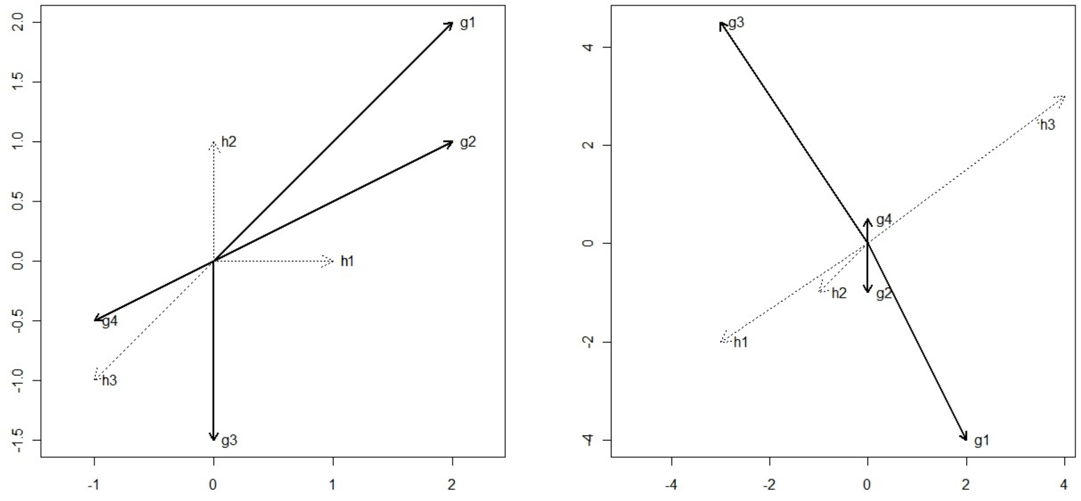


Figure 4.2: Gabriel Biplots illustrating effect of non-unique factorisation.

resulting matrix product of the transformed matrices is equal to that of the original $\boldsymbol{G}$ and $\boldsymbol{H}$ matrices. In order to understand how these transformations translate to the disparity in the graphical representations of the matrix $\boldsymbol{X}$, consider the SVD of the non-singular matrix $\boldsymbol{R}'$,

$$\boldsymbol{R}' = \boldsymbol{V}'\boldsymbol{\Sigma}\boldsymbol{W} \tag{4.6}$$

$\boldsymbol{V}$ and $\boldsymbol{W}$ are orthonormal matrices and will cause a rotation and possible reflection of the axes in the graphical representation of the matrix $\boldsymbol{X}$ where as the matrix $\boldsymbol{\Sigma}$ results in stretching or shrinking of the axes. The conclusion is thus that the graphical representation on the right of Figure 4.2 results from transforming the graphical representation in the left hand panel by the means described. This serves to emphasise the fact that the factorisation of the matrix is non-unique and that the graphical representation of the matrix $\boldsymbol{X}$ depends to a large extent on the factorisation. The question is thus how this apparent obstacle to producing a plot that affords the means to make meaningful inferences about the relations between the rows and columns of the matrix $\boldsymbol{X}$ can be conquered. The solution is rather simple and requires that a metric be imposed thus making the factorisation and the resulting plot unique (Gabriel,1971).

There are two possibilities to consider. The biplot can either accurately represent the relations between the samples which means that the the distances between the rows of the matrix $\boldsymbol{X}$ will be equivalent to the distances between the rows of the matrix $\boldsymbol{G}$. Alternatively, the relationship between the columns of the matrix $\boldsymbol{X}$ can be accurately represented which means that the correlations between the variables will be accurately represented on the biplot. Gabriel (1971) states that if it is the relations between the rows of the matrix $\boldsymbol{X}$ that are to be represented accurately by the corresponding relations between the rows of $\boldsymbol{G}$, then imposing the following requirement ensures that this is the case

$$\boldsymbol{H}'\boldsymbol{H} = \boldsymbol{I_2}. \tag{4.7}$$

This condition yields a number of different results, the first of which indicates that the inner products of the rows of $\boldsymbol{X}$ is equivalent to that of the matrix $\boldsymbol{G}$.

$$\boldsymbol{X}\boldsymbol{X}' = \boldsymbol{G}\boldsymbol{H}'\boldsymbol{H}\boldsymbol{G}' = \boldsymbol{G}\boldsymbol{G}', \tag{4.8}$$

implying that for any two rows $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ of the matrix $\boldsymbol{X}$

$$\boldsymbol{x}_i'\boldsymbol{x}_j = \boldsymbol{g}_i'\boldsymbol{g}_j. \tag{4.9}$$

It follows very simply from Equation (4.9) that

$$\|\boldsymbol{x}_i\| = \|\boldsymbol{g}_i\|, \tag{4.10}$$

implying that the inner product of any two row vectors from the matrices $\boldsymbol{X}$ and $\boldsymbol{G}$ is equivalent. Note that the inner product of two vectors can be written as $<\boldsymbol{x}_i, \boldsymbol{x}_j> = \|\boldsymbol{x}_i\|\|\boldsymbol{x}_j\| \cos(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Combining Equations (4.9) and (4.10) with this form of the inner product immediately implies that

$$\cos(\boldsymbol{x}_i, \boldsymbol{x}_j) = \cos(\boldsymbol{g}_i, \boldsymbol{g}_j). \tag{4.11}$$

The next result is perhaps the most pertinent because it shows that imposing the condition in (4.7) will ensure that the Euclidean distances between the rows of the matrix $\boldsymbol{X}$ are accurately represented by those of the matrix $\boldsymbol{G}$ which is what is meant by preserving the relations between the samples of the matrix $\boldsymbol{X}$. Recall that the Euclidean distance between two vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be written as

$$\| \boldsymbol{x}_i - \boldsymbol{x}_j \| = \sqrt{\| \boldsymbol{x}_i \| + \| \boldsymbol{x}_j \| - 2 < \boldsymbol{x}_i, \boldsymbol{x}_j >}, \tag{4.12}$$

but $\|\boldsymbol{x}_i\|=\|\boldsymbol{g}_i\|$ and $\|\boldsymbol{x}_i\|=\|\boldsymbol{g}_i\|$ and $< \boldsymbol{x}_i, \boldsymbol{x}_j >=< \boldsymbol{g}_i, \boldsymbol{g}_j >$ so that

$$\| \boldsymbol{x}_i - \boldsymbol{x}_j \| = \| \boldsymbol{g}_i - \boldsymbol{g}_j \| \, . \tag{4.13}$$

Though the relations between the rows of the matrix $\boldsymbol{X}$ are accurately presented by those of the matrix $\boldsymbol{G}$, the same cannot be said for the way in which the rows of the matrix $\boldsymbol{H}$ represent the columns of the matrix $\boldsymbol{X}$. In fact, the precise relationship between the inner product of the columns of $\boldsymbol{H}$, the vectors $\boldsymbol{h}_j$ where $j = 1, \ldots, p$, and those of the matrix $\boldsymbol{X}$ is

$$\boldsymbol{X}^{'}(\boldsymbol{X}\boldsymbol{X}^{'})^{-}\boldsymbol{X} = \boldsymbol{H}\boldsymbol{H}^{'}, \tag{4.14}$$

where $(\boldsymbol{X}\boldsymbol{X}^{'})^{-}$ is any conditional inverse of the matrix $\boldsymbol{X}\boldsymbol{X}^{'}$.

Proof.

$$\boldsymbol{X}^{'}(\boldsymbol{X}\boldsymbol{X}^{'})^{-}\boldsymbol{X} = \boldsymbol{H}\boldsymbol{G}^{'}(\boldsymbol{G}\boldsymbol{H}^{'}\boldsymbol{H}\boldsymbol{G}^{'})^{-}\boldsymbol{G}\boldsymbol{H}^{'}, \tag{4.15}$$

and $\boldsymbol{H}^{'}\boldsymbol{H} = \boldsymbol{I_2}$. A property of the conditional inverse of a matrix $\boldsymbol{A}$ is that $\boldsymbol{A}\boldsymbol{A}^{-}\boldsymbol{A} = \boldsymbol{A}$ and using $\boldsymbol{G}\boldsymbol{G}^{'}$ instead of $\boldsymbol{A}$ yields

$$\boldsymbol{G}\boldsymbol{G}^{'}(\boldsymbol{G}\boldsymbol{G}^{'})^{-}\boldsymbol{G}\boldsymbol{G}^{'} = \boldsymbol{G}\boldsymbol{G}^{'}.$$

Multiplying from the left with $\boldsymbol{G}^{'}$ and from the right with $\boldsymbol{G}$ yields

$$(\boldsymbol{G}^{'}\boldsymbol{G})\boldsymbol{G}^{'}(\boldsymbol{G}\boldsymbol{G}^{'})^{-}\boldsymbol{G}(\boldsymbol{G}^{'}\boldsymbol{G}) = \boldsymbol{G}^{'}\boldsymbol{G}\boldsymbol{G}^{'}\boldsymbol{G}. \tag{4.16}$$

Multiplying (4.16) by $(\boldsymbol{G}^{'}\boldsymbol{G})^{-1}$ from the left and right yields

$$\boldsymbol{G}^{'}(\boldsymbol{G}\boldsymbol{G}^{'})^{-}\boldsymbol{G} = \boldsymbol{I_r}, \tag{4.17}$$

where r indicates the rank restriction imposed and in this instance r is 2 so that $\boldsymbol{I}$ is in fact the 2×2 identity matrix. Substituting (4.17) into (4.16) produces the result in (4.14). □

The alternative is to ensure that the relations between the columns of the matrix $\boldsymbol{X}$ are accurately represented by those of the rows of $\boldsymbol{H}$ so that the inner products of the columns of $\boldsymbol{X}$ are reproduced by the inner products of the rows of $\boldsymbol{H}$. In order to achieve this, the necessary requirement is that

$$\boldsymbol{G}^{'}\boldsymbol{G} = \boldsymbol{I_2}, \tag{4.18}$$

so that

$$\boldsymbol{X}^{'}\boldsymbol{X} = \boldsymbol{H}\boldsymbol{H}^{'}. \tag{4.19}$$

In order to understand how the accurate representation of the relation between the columns of the matrix $\boldsymbol{X}$ can be interpreted geometrically, it is necessary to understand how the angle between two variables relates to their correlation. In the context of the biplot, the oblique axes represented by the $\boldsymbol{h}_j$ vectors represent the variables and the angle between these axes is equivalent to the correlation coefficient of these variables provided that the variables are centered. Consider two columns of $\boldsymbol{X}$, $\boldsymbol{x}$ and $\boldsymbol{y}$ and

$$r_{xy} = \frac{\sum_{i=1}^{p}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2}\sqrt{\sum (y_i - \bar{y})^2}}. \tag{4.20}$$

In the event that the variables are centred, (4.20) reduces to

$$r_{xy} = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2}\sqrt{\sum y_i^2}} = \frac{< \boldsymbol{x}, \boldsymbol{y} >}{\parallel \boldsymbol{x} \parallel \parallel \boldsymbol{y} \parallel} = \cos\theta_{\boldsymbol{xy}}. \tag{4.21}$$

The expression in (4.21) indicates that an angle $\theta_{\boldsymbol{xy}}$ of 0 degrees or 180 degrees between the vectors is equivalent to perfect correlation of 1 and -1 respectively. Equation (4.19) implies that the columns of $\boldsymbol{X}$ can be replaced by the rows of $\boldsymbol{H}$ in (4.21) and this shows that the relations between the columns of $\boldsymbol{X}$ are accurately represented by those between the corresponding rows of $\boldsymbol{H}$ (Gabriel, 1971).

Choosing to have the correlations between the p variables of the data matrix accurately represented implies that the Euclidean distance between the sample points will not be optimally presented by the inner products of *row* effects. Instead of the inner product of the rows of $\boldsymbol{G}$ reproducing that of the rows of $\boldsymbol{X}$, the precise relationship becomes

$$\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-}\boldsymbol{X}' = \boldsymbol{G}\boldsymbol{G}'. \tag{4.22}$$

The proof of (4.22) follows analagously to that of (4.14). This has a tangible interpretation in the context of Principal Component Analysis biplots which will be discussed later.

4.3 Principal Component Analysis biplot

The Principal Component Analysis (PCA) biplot is arguably the simplest of the asymmetric biplots and this section concerns itself with the technical aspects underpinning the construction of such a biplot before delving into its applications. For the purpose of illustration an example is taken from (Gower *et al.*, 2011). A data set comprising 25 observations on 3 variables is used in the process of examining how a multidimensional data set,which would ordinarily be represented by a multidimensional scatterplot, can in fact be represented by a two dimensional plot *viz.* a PCA biplot.

4.3.1 Principal Component Analysis and its biplot

Principal Component Analysis is one of the oldest techniques forming part of the arsenal of multivariate analysis (Joliffe, 2002). The technique was developed by Pearson in 1901 and then independently discovered by Hotelling in 1933. The context in which each of these prolific scientists made this discovery was vastly different. Pearson (1901) was concerned with finding the best line or plane to represent a system of points in p dimensional space and the solution came in the form of Principal Components. Hotelling (1933) was concerned with reducing the dimensionality of a data set by finding a smaller set of variables that expressed the original p variables. Hotelling's approach has become a popular definition for Principal Components. Joliffe (2002) defines PCA in the same vein as Hotelling, essentially stating that it is aimed at "reducing the dimensionality of a large set of interrelated variables, whilst retaining as much of the variation present in the data" (p.1). Ultimately these two approaches are different perspectives on the problem of dimension reduction where Pearson (1901) had a geometric interpretation and Hotelling (1933), a more statistical interpretation. Naturally in the context of graphically representing a multivariate data set, the geometric interpretation is the pertinent one. Furthermore, the construction of the biplot will emphasise a slightly different aspect of the PCA transformation to that mentioned in the definition provided by Joliffe (2002).

Formally, PCA seeks to approximate a data matrix $\boldsymbol{X}$, comprising n observations and p variables, by a matrix $\hat{\boldsymbol{X}}_{[r]}$ of rank r. The approximation is based on minimizing a least-squares criterion which can be represented as

$$\|\boldsymbol{X} - \hat{\boldsymbol{X}}_{[r]}\|^2 = tr[(\boldsymbol{X} - \hat{\boldsymbol{X}}_{[r]})'(\boldsymbol{X} - \hat{\boldsymbol{X}}_{[r]})]. \tag{4.23}$$

Geometrically, the n observations comprising the matrix $\boldsymbol{X}$ can be represented in p dimensions and PCA seeks to find the best fitting r dimensional plane, a subspace of the p dimensional space, containing the points with coordinates given by the rows of $\hat{\boldsymbol{X}}_{[r]}$. The best-fitting plane is the one that minimises the criterion in (4.23).

Section 4.2 outlined the tools necessary for the construction of a biplot and what follows is a description of how those tools apply by considering an example. The fictitious data set as well as the accompanying figures are taken from Gower *et al.* (2011). The data set used in the example comprises 25 observations on 3 variables, the values for n and p respectively. Figure 4.2 is a three dimensional plot of the data and is the geometric representation of the data matrix $\boldsymbol{X}$. The aim of PCA is to find a two dimensional plane that

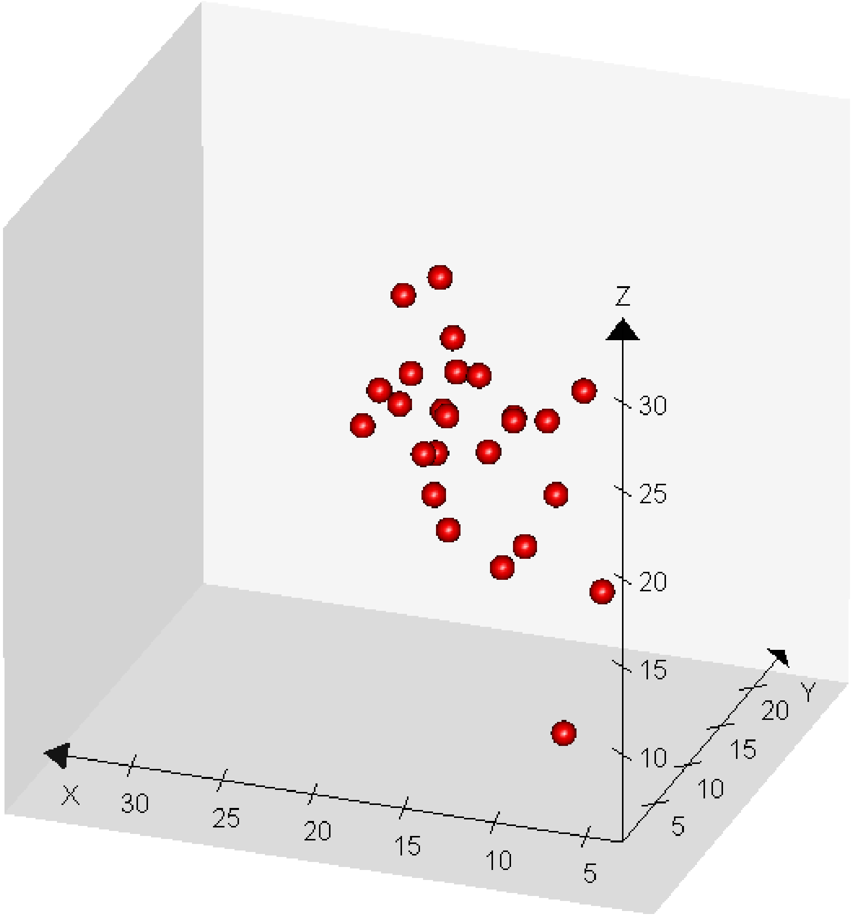


Figure 4.3: Three dimensional representation of the data, (Gower *et al.*, 2011).

bests fits the data as displayed in Figure 4.2. The significance of Huygen's Principle is that it implies that the best fitting plane should pass through the centroid of the points in $\boldsymbol{X}$ because the sum of squares about the mean is smaller than that about any other point. The expansion of the expression in (4.23) will include $tr(\boldsymbol{X}'\boldsymbol{X})$ which is effectively the sum of squares about the origin. Since the aim is to minimise (4.23), $\boldsymbol{X}$ is replaced by $(\boldsymbol{X}\text{-}\frac{1}{n}\mathbf{1}\mathbf{1}'\boldsymbol{X})$ thus ensuring that the centroid of the data is at the origin at that the term $tr(\boldsymbol{X}'\boldsymbol{X})$ is minimised. Ultimately, Huygen's Principle requires that the data be centered.

What remains is to determine the direction of the best-fitting plane (Gower *et al.*, 2011). The Eckart-Young Theorem (Eckart and Young, 1936) holds the solution to this problem. The Eckart-Young Theorem gives the approximation of $\boldsymbol{X}$ as

$$\hat{\boldsymbol{X}}_{[r]} = \boldsymbol{U}\boldsymbol{\Sigma}_{[r]}\boldsymbol{V}'. \tag{4.24}$$

Equation (4.24) can be represented in $\boldsymbol{J}$ notation as

$$\hat{\boldsymbol{X}}_{[r]} = \boldsymbol{(UJ)}\Sigma\boldsymbol{(VJ)}' = \boldsymbol{U}_{[r]}\Sigma\boldsymbol{V}'_{[r]}, \tag{4.25}$$

where $\boldsymbol{U}_{[r]}$ and $\boldsymbol{U}_{[r]}$ comprise the first r columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ respectively. Introducing a parameter α leads to (4.25) being represented as

$$\hat{\boldsymbol{X}}_{[r]} = \boldsymbol{U}_{[r]}\boldsymbol{\Sigma}^{\alpha}\boldsymbol{\Sigma}^{1-\alpha}\boldsymbol{V}'_{[r]}. \tag{4.26}$$

The equation in (4.26) can be related to the representation that Gabriel (1971) put forward where $\boldsymbol{G} = \boldsymbol{U}_{[r]}\boldsymbol{\Sigma}^{\alpha}$ and $\boldsymbol{H}' = \boldsymbol{\Sigma}^{1-\alpha}\boldsymbol{V}'_{[r]}$. The parameter α commonly takes on the value 0, $\frac{1}{2}$ or 1. This determines whether the Euclidean distances between sample points or correlations between variables is optimally approximated on the biplot. In the case of the PCA biplot, α is set equal to 1 and condition (4.7) is satisfied so that the relationship between the rows of $\boldsymbol{G}$ accurately represent the relationship between the corresponding rows of $\boldsymbol{X}$. Rudimentary algebraic manipulation yields $\boldsymbol{G} = \boldsymbol{X}\boldsymbol{V}_{[r]}$ and $\boldsymbol{H} = \boldsymbol{V}_{[r]}$. This representation provides some insight into the geometrical interpretation. $\boldsymbol{G}$ indicates that the n rows of $\boldsymbol{X}$ are projected onto the column space of $\boldsymbol{V}_{[r]}$ to produce the row markers or sample point representations. In this form, $\boldsymbol{G}$ comprises n row vectors with r elements; it is a representation of the sample point relative to the orthogonal axes which underly the best-fitting plane. This can be understood geometrically by considering Figure 4.4 which shows how each of the observations in three dimensional space are orthogonally projected onto the best-fitting two dimensional plane. This process is referred to as interpolation. The orientation of this best-fitting plane is determined by the columns of $\boldsymbol{V}_{[r]}$. As a direct result of the fact that interpolating sample points is based on orthogonal projection and that $\boldsymbol{V}_{[r]}$ is orthogonal, a representation of the sample points relative to the original p orthogonal axes is given by

$$\boldsymbol{X}\boldsymbol{V}_{[r]}\boldsymbol{V}'_{[r]}. \tag{4.27}$$

The direction of the p variable axes are given by rows of $\boldsymbol{V}_{[r]}$. In effect, the first r columns of the matrix $\boldsymbol{V}$ represent the direction cosines of the best fitting plane and $\boldsymbol{X}\boldsymbol{V}_{[r]}$ represents the projection of the n sample points onto the best fitting plane. It is assumed that r is equal to 2 since a two dimensional representation of the data is sought. This is shown in Figure 4.4. Note that the correlation between the variables comprising the data matrix are no

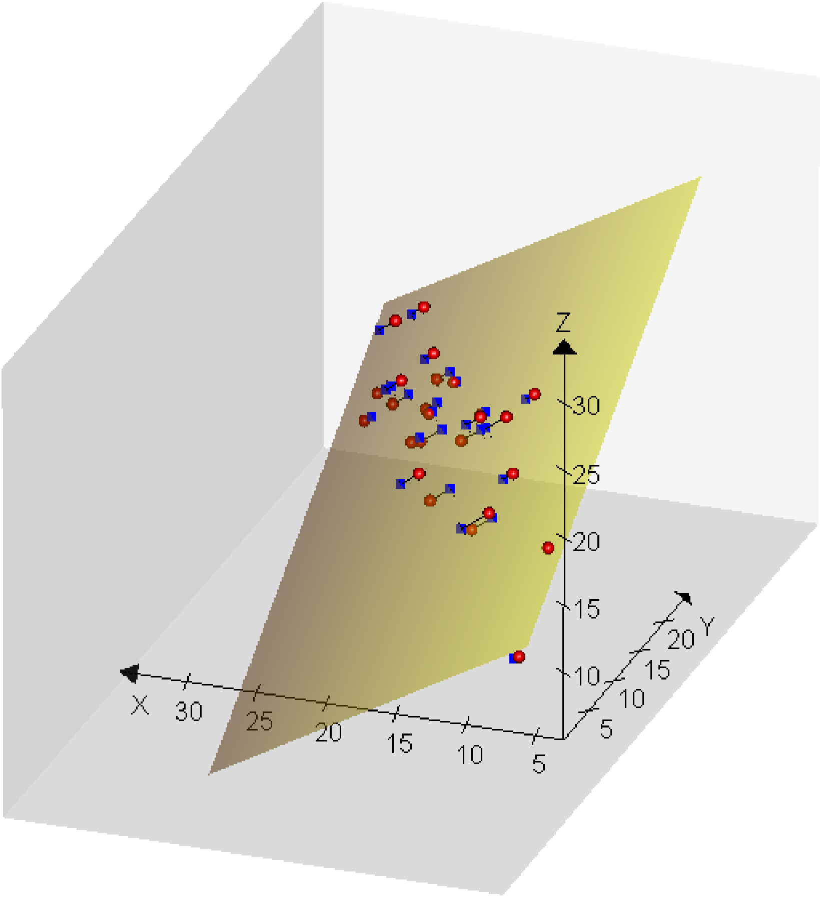


Figure 4.4: Geometric illustration of the rows of $\boldsymbol{G}$, (Gower *et al.*, 2011).

longer optimally approximated by the column effects but it is still possible to make a statement about the extent of the correlation between the variables in a qualitative sense; the smaller the angle between the column effects, the greater the correlation between the corresponding variables.

The alternative of setting α to 0 results in condition (4.20) being satisfied and consequently the cosine of the angles between the column effects optimally approximate the correlations between the p variables comprising the data matrix $\boldsymbol{X}$. In this case the inner products of the row effects have a meaningful interpretation in that the Euclidean distance between any two row effects $\boldsymbol{g}_i$ and $\boldsymbol{g}_j$ is proportional to the Malahanobis (1936) distance between the i^{th} and j^{th} observations in the data matrix $\boldsymbol{X}$. Combining the fact that $\boldsymbol{X}$ is centered with definition of the Malahanobis distance together with (4.22) is sufficient to proof this statement.

Attention can now be directed to a providing a more comprehensive understanding of the biplot axes. Notice that the Gabriel biplot represents the biplot axes as arrowed vectors. The length of this vector from the origin to the tip of the arrowhead typically indicates one standard deviation of the variable considered. Gower and Hand (1996) contend that though this is an acceptable form of representation especially useful when the purpose is to approximate variance, covariance and correlation, even then it is not entirely satisfactory. This is because in treating the biplot as the multivariate analog to the scatterplot, one would require that the axes be calibrated and that orthogonal projections onto the axes can be done simply. In the vector representation the axes need to be extended at times in order to facilitate the orthogonal projection of sample points onto the axes and they are not calibrated either. This dissertation follows the convention described by Gower and Hand (1996) thus the discussion on biplot axes starts with the process of calibration.

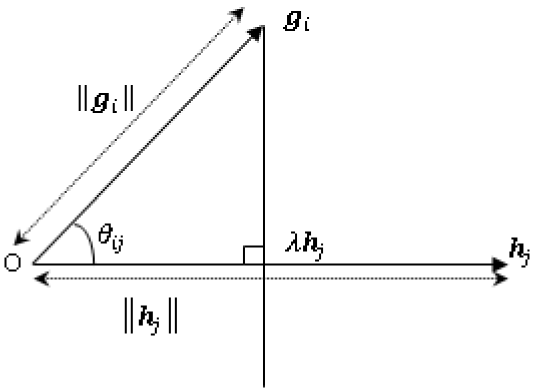


Figure 4.5: Illustrating the process of axes calibration.

Figure 4.5 provides a simple means to explain the process of calibrating the biplot axis. Recall that the data matrix can be approximated with $\hat{\boldsymbol{X}}_{[2]}$ based on the SVD of $\boldsymbol{X}$. This can be factorised into the product of two matrices $\boldsymbol{G}$ and $\boldsymbol{H}$ with each of the rows of $\boldsymbol{G}$ representing the row effects and each of the rows of $\boldsymbol{H}$ representing column effects as described in section 4.2.4.

Figure 4.5 is a representation of a hypothetical row effect $\boldsymbol{g}_i$ as well as a column effect $\boldsymbol{h}_j$ associated with $\hat{\boldsymbol{X}}$ for an $n \times p$ data matrix $\boldsymbol{X}$. In the context of the physical plot, the points $\boldsymbol{g}_i$ and $\boldsymbol{h}_j$ are termed the *row markers* and *column markers* respectively. The vector $\boldsymbol{h}_j$ is commonly referred to as a *biplot axis* since the column markers represent the variables. Furthermore, each element of $\hat{\boldsymbol{X}}_{[2]}$ is represented by the inner product of the row and column effects. This can be represented by

$$\hat{x}_{ij} = \parallel \boldsymbol{g}_i \parallel \parallel \boldsymbol{h}_j \parallel \cos\theta_{ij}. \tag{4.28}$$

The inner product is integral in calibrating the biplot axis because $\boldsymbol{g}_i^{'} \boldsymbol{h}_j$ is constant for all points on the locus that project $\boldsymbol{g}_i$ onto $\boldsymbol{h}_j$ (Gower and Hand, 1996). The point of projection is calibrated with a value μ, the inner product. The point of projection $\lambda\boldsymbol{h}_j$ lies on the locus and thus it must satisfy the condition

$$\lambda\boldsymbol{h}_j^{'}\boldsymbol{h}_j = \mu. \tag{4.29}$$

This indicates that $\lambda = \frac{\mu}{\boldsymbol{h}_j^{'}\boldsymbol{h}_j}$ which is a scalar and in order to obtain the co-ordinates of the point to be calibrated with the μ, $\boldsymbol{h}_j$ is multiplied by lambda. Generally, μ will take on various integer values such as 1,2,3,... or any values that are convenient for the calibration which will be determined by the size of the inner products. Generally the mean of the variable in question is used to conveniently calibrate the axis. This gives a basic understanding of the process of calibration of the biplot axes however there is a further complication to consider in the form of *prediction* and *interpolation.*

It is necessary to ponder whether a single set of axes can serve both as predictive and interpolative axes. By virtue of the fact that prediction uses a process of orthogonal projection onto the biplot axes where as interpolation uses vector addition by means of the parallelogram method to add a new sample point to the biplot display it should be immediately obvious that one set of axes cannot suffice for both purposes. An interpolated point will not yield the values used for the interpolation when read off of the axes in the manner prescribed by the process of prediction. In the process of interpolation, the biplot axes are used for the purpose of adding new sample points to the plot by means of vector addition and are thus termed *interpolative biplot axes.* In effect, the process of interpolation gives co-ordinates of the sample points in the lower dimensional space $\mathcal{L}$. In context, the interpolated point is the two dimensional representation in $\mathcal{L}$ of the p dimensional representation in the original space. Recall that the point $\boldsymbol{x}$ in p-dimensional space is interpolated into the two dimensional plane $\mathcal{L}$ to the point $\boldsymbol{z}$ as follows

$$\boldsymbol{z} = \boldsymbol{x}\boldsymbol{V}_{[2]}. \tag{4.30}$$

Consider the k $1 \times p$ vectors $\boldsymbol{e}_k^{'}$ as representing the unit vectors in the directions of each of the p axes in the original space, then equation 4.30 can be represented as

$$\boldsymbol{z} = \sum_{k=1}^{p} x_k \boldsymbol{e}_k^{'} \boldsymbol{V}_{[2]}. \tag{4.31}$$

Graphically the original axes are simply being projected onto the best fitting plane $\mathcal{L}$ which is represented by $\boldsymbol{e}_k^{'} \boldsymbol{V}_{[2]}$. This is represented in Figure 4.6. In order to calibrate the axes, consider the expression $\mu \boldsymbol{e}_k^{'} \boldsymbol{V}_{[2]}$. Varying the value μ as described previously will provided the co-ordinates for the points to be calibrated with the value μ.

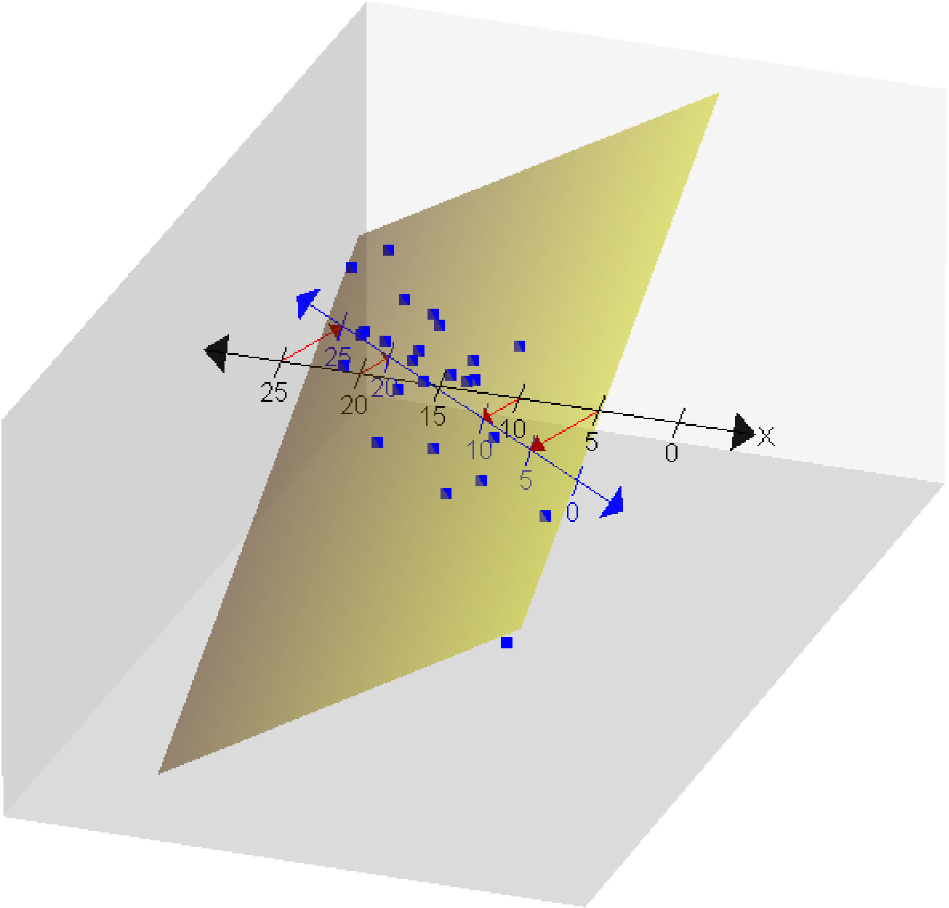


Figure 4.6: Constructing interpolative biplot axes (Gower *et al.*, 2011).

Predictive biplot axes are more commonplace and coincide with the notion that the biplot is an abstraction of the simple scatterplot. More specifically, projections onto the biplot axes can be used as approximate values for the variables comprising the data set. The basis for calibrating these axes is closely related to the general discussion on calibrating biplot axes. It was shown that the co-ordinates for calibrating the $r \times 1$ biplot axis $\boldsymbol{h}_j$ with the value μ could be found by calculating

$$\frac{\mu}{\boldsymbol{h}_j^{'}\boldsymbol{h}_j}\boldsymbol{h}_j. \tag{4.32}$$

In the case of the PCA biplot, the direction of the axes are given by the 1×2 vectors $\boldsymbol{e}_k^{'}\boldsymbol{V}_{[2]}$ which is identical to the the direction of the interpolative axes, however the co-ordinates will be given by the expression

$$\frac{\mu}{\boldsymbol{e}_k^{'}\boldsymbol{V}_{[2]}\boldsymbol{V}_{[2]}^{'}\boldsymbol{e}_k}\boldsymbol{e}_k^{'}\boldsymbol{V}_{[2]}, \tag{4.33}$$

where the value of μ is varied across suitable integer values as determined by the data and $\boldsymbol{h}_j$ is replaced by $\boldsymbol{V}_{[2]}^{'}\boldsymbol{e}_k$ in (4.32) .

4.4 Application to Mansoor data

This section concerns itself with applying the PCA biplot methodology to matricised forms of the Mansoor data. More specifically the biplot methodology will be applied to the averaged data set, the tall combination as well as the wide combination of the data. The merits and shortcomings of each biplot will be discussed.

Before delving into constructing each of the biplots it is necessary to consider how to assess the efficacy of biplots as exploratory tools. In attempting to answer this question one must be cognisant of the types of questions that three way analysis is capable of answering and evaluate whether the biplot affords the means to answer these questions. Kroonenberg (2008) provides an eloquent discussion on the power of three-way analysis by way of example. He uses the example of children being tracked over a number of years on a number of attributes and defines questions that might be of central importance. The researcher is bound to ask questions like

- What are the relationships between the variables?
- What trends may be discovered over time?

- Is there any structure to the children being tracked?

These questions in effect ask something about each way comprising the data. Three way analysis is capable of answering questions in which the various ways are combined. For instance, do the relationships between the variables change over time? Do the children score differently on the various attributes over time? The most complex questions combine all three ways. The assessment framework thus becomes whether these PCA biplots afford the means to answer complex questions in which various ways of the data are combined.

To begin, the wide combination of the data will be considered. Recall that the wide combination of the data matrix combines the variable and time modes so that their effects are confounded in the context of formal statistical modelling procedures. It might be that in the process of data visualisation the possibility to make a statement about time and variable effects is possible. In other words, the question at hand is whether the PCA biplot of the wide combination matrix provides the means to make a statement about how the relationships between variables evolve over time. In constructing the associated PCA biplot the $n \times kp$ matrix was constructed by placing each of the data matrices for each time point next to one another. In this instance n is equal to 29 and since k is 4 and p is 7, the matrix has size 29×28. Figure 4.7 illustrates the wide combination PCA biplot for the Mansoor data and it was constructed so that the correlation between variables is optimally represented. Before evaluating this biplot use as a tool for visualising threeway data attention will be given to understanding what it reveals about the structure of the data. It is logical to link the interpretation to the questions that Mansoor *et al.* (2009) were seeking to answer in their investigation and thus a succinct reminder is provided. Mansoor *et al.* (2009) was interested in determining whether the vaccine BCG induced the required immune response for TB protection in HIV^+ infants and whether the induced immune strength in the two groups of HIV negative infants was the same. The axes with the labels including $V4$ represent the T-cells that co-expressed all three cytokines, which is thought to be the required immune response. It is perfectly clear that the observations comprising group 1 are clustered close to the zero points for each of the axes. This implies that all these sample points having relatively low scores for the number of T-cells co-expressing all three cytokines, variable four. At first glance this seems to suggest that BCG does not induce the required immune response in HIV positive infants. Although a fair number of infants in group 3, children born to HIV negative mothers, also score low on the number of cells co-expressing all three genes it is members of this group that show the highest scores on

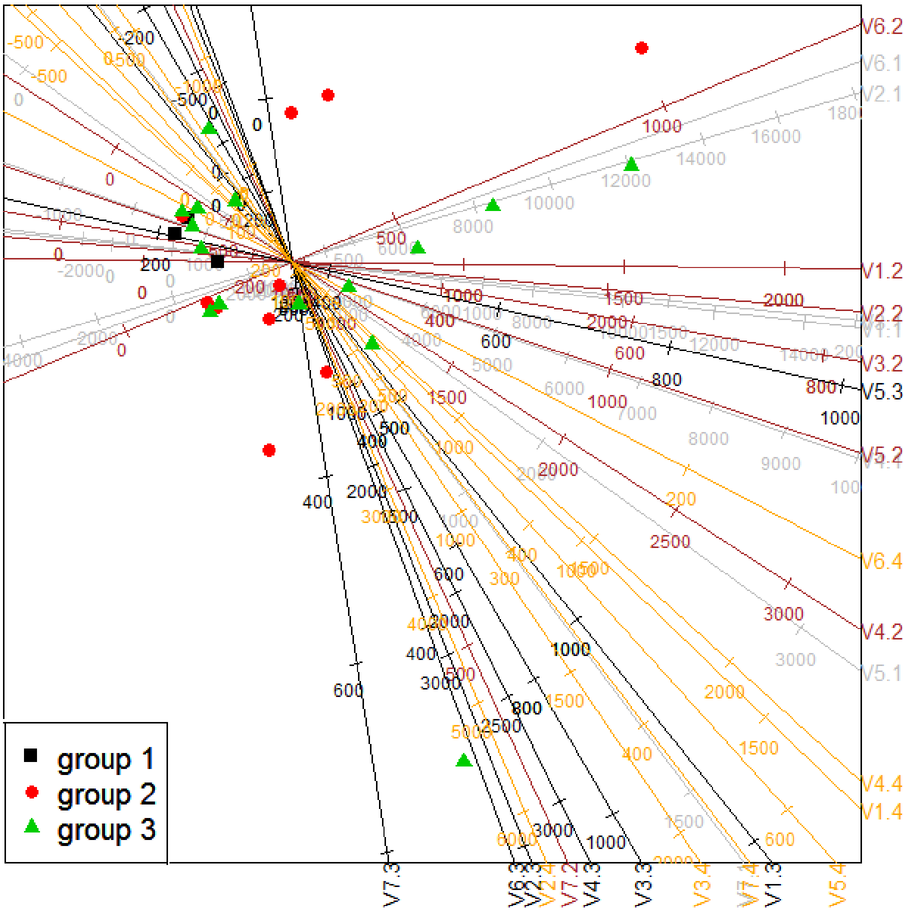


Figure 4.7: PCA biplot for the wide combination of the Mansoor Data with correlation optimally represented.

this variable also. Furthermore, there is a lot more variation present in group 3 as indicated by the spread of the sample points in the display.

If the vaccine induced similar immune strength in groups 2 and 3 then infants in these groups would have similar scores on most of the variables. Graphically this would translate to a similar spread of the observations comprising these groups. The observations in group 2 are more tightly clustered relative to those in group 3. In general infants in group 3 tend to score higher than those in group 2 particularly for the polyfunctional T-cells. It may well be the case that infants in group 3 enjoy greater immune strength induced by the BCG vaccine. It is interesting to note that of the highest and lowest scores of all the infants belong to members in group 3. It is necessary to reiterate that this is an exploratory process and the data should be subjected to more formal statistical modelling but it is clear that the biplot provides a sense of the structure of the data. It should be noted that to get a sense of changes in variability across occasion one need only project the sample points onto the relevant axes to see how dispersed the variables scores are. For example projecting all the sample points onto $V1.1$ and $V1.2$ and comparing the spread gives a sense of how the variability has changed on this variable.

A statement can also be made about the relationship between the variables as indicated by the acute angles between the variable axes. Notice that statements can be made about relationships between variables across times points which is an extra element unique to three mode data. All the angles between axes can be interpreted as giving information about the strength of association between variables. As an example, consider the relationship between $V1.1$. and $V1.2$ where $V1.1$ refers to $V1$ at occasion 1 and so forth. The angle between these variables axes is relatively small indicating a strong relationship between the variables and this correlation coefficient was found to be 0.63. The plot also gives an indication of how the relationship between variables evolved over time. The relationship between variables one and two can be examined over time. It is important to note that this plot is based on approximation and that the PCA technique is scale invariant. The scale invariance means that if the data are not standardised before applying the PCA technique then directions with a lot of variation will be influential in determining the orientation of the scaffolding axes and will thus be better represented than other directions. Standardising the data prevents this from happening but it also makes it difficult to see differences in the variation in the data over time. The fact that the plot is based on a two dimensional approximation of the data means that not all information is accurately conveyed. $V6.1$ and $V2.1$ have a correlation of 0.32 and examining Figure 4.7, the angle between these two axes suggests a stronger relationship than this. This is attributable to the issues of scale invariance and approximation. This does not render the tool useless since it still indicates a relationship between the variables albeit overstated. In effect, variables that show relatively small angles between them should be inspected by considering the correlation matrices because these variables are bound to have an association.

In the context of the investigation, the biplot provided answers and when assessed as a tool for answering research questions that are three-way in nature it serves well, giving information on how observation scores evolve over time as well as how relationships between variables evolve over time.

The next consideration is that of a tall combination of the data in which the sample and time ways are confounded. It is precisely this problem in formal modelling that allows the associated biplot to shed light on questions of a three-way nature. Here the separate data matrices are combined into a single matrix with dimensions 116×7, the number of rows being the product of the number of observations per occasion, 29, and the number of occasions, 4. This combined matrix is used to construct the PCA biplot. Figure 4.8 illustrates the PCA biplot for the tall combination of the Mansoor data and the Euclidean distance between observations is optimally represented. The most notable difference between this biplot and that in Figure 4.7 is that there are fewer axes but more sample points on the plot. In terms of the assessment framework described, this biplot would thus serve well in answering questions about the evolution of an attribute over time but fails in providing information about how the relationships of the variables changes over time. Careful inspection of the plot reveals much the same information regarding the distribution of the sample points as in the wide combination biplot. The power of this plot lies in the fact that it affords the means to visually appraise how observation scores evolve over time. Where as the wide PCA biplot allowed the researcher to read off the scores for a given observation over time, the tall PCA biplot allows the researcher to see whether the scores have changed substantially by looking at the Euclidean distance between points corresponding to the same observation. One key aspect of the data that is revealed in Figure 4.8 is that the variation in the data decreases over time. It also indicates that the mean of the variables tended to decrease over time.

It is important to be aware of the fact that the biplot has been constructed so that the Euclidean distances between the sample points are optimally represented. The implication is that the angles between variable axes do not optimally approximate the correlations between variables but it is still possible to make a statement about the strength of association between variables; smaller angles suggest strong association. It is not completely clear what the strength of association represented in Figure 4.8 actually means and this must be considered. Thinking about the construction of the tall PCA biplot from a geometric perspective may shed some light on this matter. Constructing the tall PCA biplot amounts to finding the best fitting plane in $\Re^p$ where $p = 7$. It is very similar to the construction process that occurs when PCA biplots are constructed for each occasion separately but for the fact that in this context all observations across occasions are used in the construction process. The dispersion of the observations in $\Re^p$ determines the orientation of the best fitting plane and in the case of the tall PCA biplot the dispersion of all the observations comprising the data set influence the orientation of the scaffolding. If there are associations between variables that persist across occasion and this is represented in the separate biplots then it will certainly appear in the biplot constructed from all the observations collectively. The separate PCA biplots for the four occasions show persistent associations between variables one and four, variables five and seven and variables three and six. A quick examination of Figure 4.8 indicates that these variables are shown to have an association. The strength of association indicated on the tall PCA biplot is thus that which persists across occasion. The contention here is that this biplot is better for understanding the evolution of observations over occasion not only because it provides a visual appraisal but also because of the construction process. If directions of greatest variability are very similar across occasion then this is preserved and the biplot should represent the observations well.

Finally the averaged data set is considered where the data are collapsed along the third way. In this instance the data are averaged across the third way(time) so that every entry in the averaged data matrix $\boldsymbol{M}$ is calculated as

$$m_{ij} = \frac{1}{4}\sum_{k=1}^{4} x_{ijk}. \tag{4.34}$$

A PCA biplot is then constructed from this matrix $\boldsymbol{M}$. The result of this process can be seen in Figure 4.9. The biplot constructed from the averaged data is the least useful of the three biplots considered here. It still provides a modicum of information regarding the spread of the observations and paints a similar picture to the previous displays. What is interesting about the plot is that the associations between $V1$ and $V4$ as well as $V5$ and $V7$ are similar to that seen in Figure 4.8, the tall PCA biplot. In fact, the orientation of the variable axes in Figure 4.9 looks very similar to that of the PCA biplot for occasion 1. The variation was greatest at occasion 1 and so it stands to reason that this occasion would have substantial influence in the construction process.

4.5 Conclusion

This chapter detailed the construction of the PCA biplot, the simplest of the asymmetric biplots. PCA biplots were then constructed for the matricised Mansoor data. It was seen that *BCG* did not seem to induce the required immune response in HIV^+ infants with similar immune strength being displayed by the groups comprising the uninfected infants. The wide combination PCA biplot afforded the means to make a statement about the relationship between variables over time. It was argued that the tall PCA biplot was preferred for making a statement about the evolution of observation scores over time and this was due to the construction process and the fact that it provided a visual means of gleaning this information. This plot also gave information regarding the change in variability in the data. Finally the PCA biplot for the averaged data was discussed and this was deemed to be the least useful of the array of biplots discussed. The PCA biplots proved to be useful as an exploratory tool for providing answers to questions of a three way nature. More than that, these displays were able to reveal much about the structure of the data. It is thus useful to use specifically the tall and wide PCA biplot in conjunction with one another to understand the structure of the data as opposed to a myriad scatterplots and boxplots which is traditional.

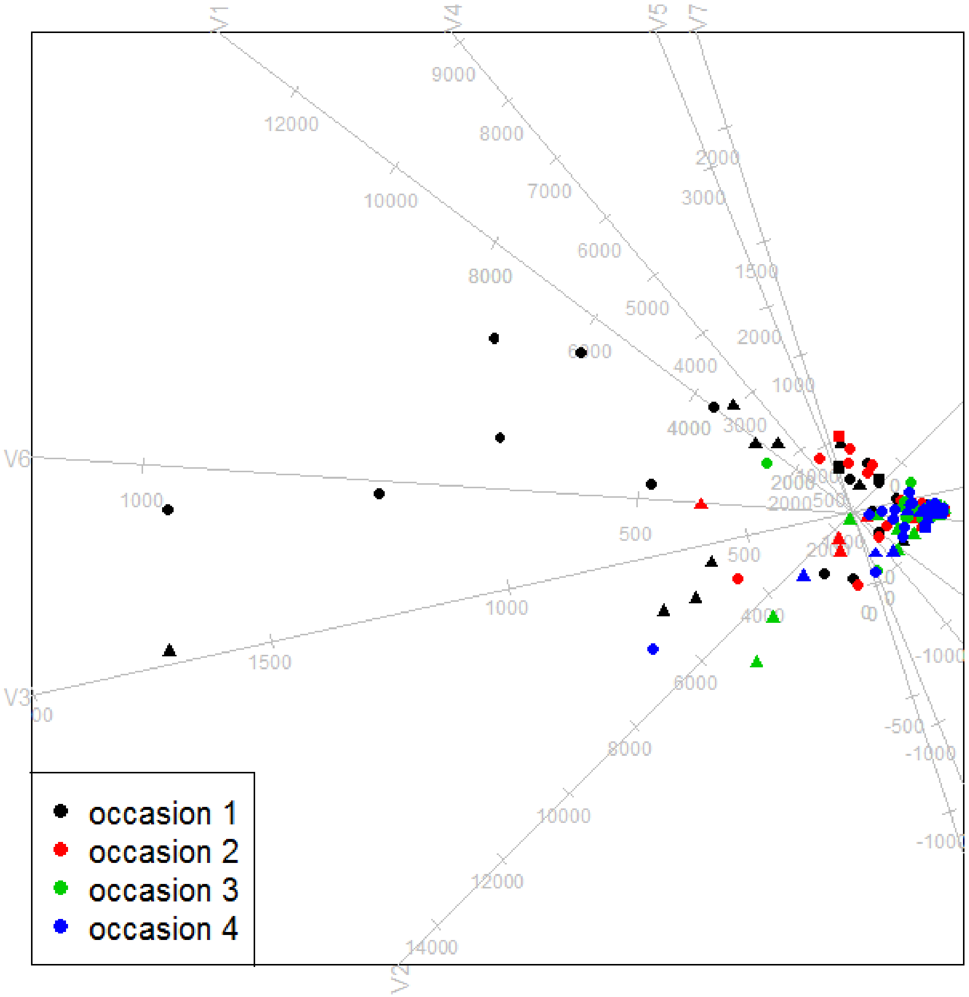


Figure 4.8: PCA biplot for the tall combination of the Mansoor Data.

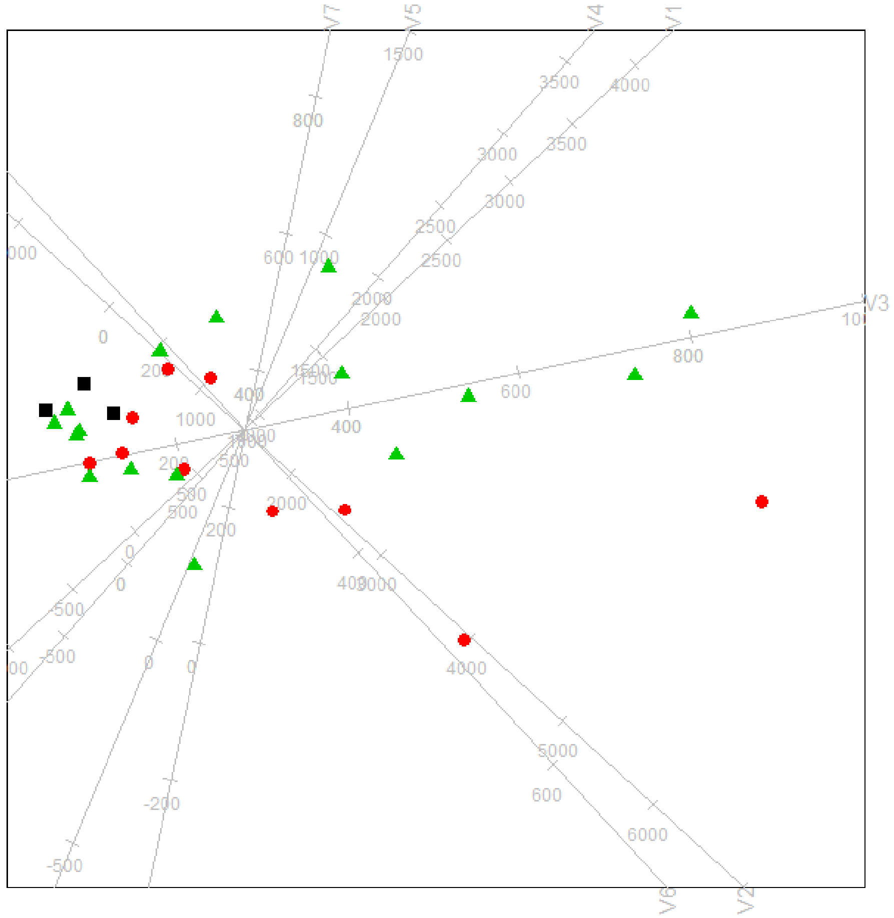


Figure 4.9: PCA biplot for the aggregated Mansoor Data.

Chapter 5

Canonical Variate Analysis Biplots

5.1 Introduction

In a seminal paper, Hotelling (1936) introduced the ideas of understanding the relations between two sets of variables which form the foundations of Canonical Variate Analysis (CVA); a technique specifically concerned with data that are grouped into g classes. In this instance the two sets of variables of interest comprise the group indicators as well as the variables measured in the dataset. When data are grouped in this fashion then there is both between and within group variation that is of interest and the primary question is how to exhibit this graphically. CVA biplots provide the means to represent this type of data and visually appraise the separation between the groups comprising the data. Gabriel (1972) introduced what he termed a MANOVA biplot in order to analyse meteorological data and this is the first instance of constructing a biplot affording the means to visually appraise the separation of groups in the data. A likely question might be how CVA biplots differ from their PCA counterparts and the answer predominantly lies in the fact that although the grouped nature of the data can be represented on the PCA biplot in a number of ways it is not considered in the process of constructing the biplot where as this group structure is specifically taken into consideration in the process of constructing the CVA biplot. The grouped nature of the data can only be represented by means of colour and symbols in the PCA biplot. Group means can also be interpolated onto the PCA biplot. Gower *et al.* (2011) mention some of the differences between CVA and PCA biplots but go on to mention what is arguably the fundamental difference; in PCA biplots the interpolated group means make no contribution to the scaffolding axes of the biplot where as in the CVA biplot the scaffolding axes are *determined* by the group means. This chapter briefly explains the construction of such biplots as well as considers its applications to the Mansoor data set.

5.2 Canonical Variate Analysis

Consider an $n \times p$ centered data matrix $\boldsymbol{X}$ grouped into g classes. Define the diagonal $g \times g$ matrix $\boldsymbol{N} = diag(n_1, \ldots, n_k)$ as well as $\overline{\boldsymbol{X}}$ as the $g \times p$ matrix of group means. $\boldsymbol{N}$ is the matrix with the sample size of each of the g groups on the diagonal. These matrices afford the means to construct a between and within analysis of variance as illustrated in Table 5.1. Gower and Hand (1996) provide an elegant means of posing the question that CVA seeks to answer. In essence, CVA is concerned with finding the $p \times 1$ vector $\boldsymbol{m}$ such that the linear combination $\boldsymbol{Xm}$ of the p variables maximizes the between-to-within groups variance ratio. Mathematically the ratio is represented as

Between groups	$g-1$	$\boldsymbol{B} = \overline{\boldsymbol{X}}^{'}\boldsymbol{N}\overline{\boldsymbol{X}}$
Within groups	$n-g$	$\boldsymbol{W} = \boldsymbol{X}^{'}\boldsymbol{X} - \overline{\boldsymbol{X}}^{'}\boldsymbol{N}\overline{\boldsymbol{X}}$
Total	$n-1$	$\boldsymbol{T} = \boldsymbol{X}^{'}\boldsymbol{X}$

Table 5.1: Decomposition of the Total Sums of Squares and Products.

$$\frac{\boldsymbol{m}^{'}\boldsymbol{Bm}}{\boldsymbol{m}^{'}\boldsymbol{Wm}}. \tag{5.1}$$

As a result of the fact that the scaling of $\boldsymbol{m}$ is not unique, the solution to the maximisation problem will not be unique. To circumvent this problem, a constraint is placed on the scaling of $\boldsymbol{m}$ which is given as $\boldsymbol{m}^{'}\boldsymbol{Wm} = 1$. The problem is thus to maximise (5.1) subject to the imposed constraint and the process of solving this yields

$$\boldsymbol{Bm} = \lambda \boldsymbol{Wm}. \tag{5.2}$$

Given the constraint that was imposed, (5.2) implies that $\boldsymbol{m}^{'}\boldsymbol{Bm} = \lambda$. This shows immediately that the ratio in equation (5.1) to be maximized is in fact equal to λ. Equation (5.2) represents the two-sided eigenvalue problem (Gower and Hand, 1996). The solution to the CVA problem is given when $\boldsymbol{m}$ is chosen to be the eigenvector associated with the largest eigenvalue of (5.2). There are p solutions to the problem, $p-1$ of which are associated with sub-optimal values of λ. All p solutions are important in the process of constructing the CVA biplot. Of fundamental importance is the fact that the Euclidean distance between the canonical means, the representation of the means after applying the transformation, is equivalent to the Mahalanobis distance between the group means. In order to see this, first all p eigenvectors are gathered in the form

$$\boldsymbol{BM} = \boldsymbol{WM\Lambda}, \tag{5.3}$$

where $\boldsymbol{M}$ is the $p \times p$ matrix with columns comprising the p eigenvalues that solve (5.2) and $\boldsymbol{\Lambda}$ is the diagonal $p \times p$ matrix with the associated eigenvalues on the diagonal in descending order. (5.3) is the matrix representation of the two-sided eigenvalue problem. It can be shown that $\boldsymbol{M}$ is orthogonal in $\boldsymbol{W}$ and this implies that

$$\boldsymbol{M}^{'}\boldsymbol{WM} = \boldsymbol{I}. \tag{5.4}$$

An equivalent representation of (5.4) is given by $\boldsymbol{MM}^{'} = \boldsymbol{W}^{-1}$. The data matrix $\boldsymbol{X}$ as well as the group means $\overline{\boldsymbol{X}}$ are represented as $\boldsymbol{XM}$ and $\overline{\boldsymbol{X}}\boldsymbol{M}$ respectively in the canonical space. The canonical means for the k^{th} group can be represented as $\overline{\boldsymbol{x}^*}^{'}_{k} = \overline{\boldsymbol{x}}^{'}{}_{k}\boldsymbol{M}$ where $\overline{\boldsymbol{x}}^{'}{}_{k}$ is a $1 \times p$ vector. Consider finding the Euclidean distance between canonical means $\overline{\boldsymbol{x}^*}^{'}_{k}$ and $\overline{\boldsymbol{x}^*}^{'}_{h}$ for the k^{th} and h^{th} group respectively.

$$\begin{aligned}(\overline{\boldsymbol{x}^*}_k - \overline{\boldsymbol{x}^*}_h)^{'}(\overline{\boldsymbol{x}^*}_k - \overline{\boldsymbol{x}^*}_h) &= (\overline{\boldsymbol{x}}_k - \overline{\boldsymbol{x}}_h)^{'}\boldsymbol{MM}^{'}(\overline{\boldsymbol{x}}_k - \overline{\boldsymbol{x}}_h) \\ &= (\overline{\boldsymbol{x}}_k - \overline{\boldsymbol{x}}_h)^{'}\boldsymbol{W}^{-1}(\overline{\boldsymbol{x}}_k - \overline{\boldsymbol{x}}_h),\end{aligned} \tag{5.5}$$

which is equivalent to the Mahalanobis distance between the group means in Euclidean space. The Mahalanobis distance, introduced by Mahalanobis in 1936, differs from Euclidean distance in that the latter effectively treats all variables as having equal variances and being uncorrelated where as the former gives relatively lower weights to those variable with large variances and groups of variables that are highly correlated (Joliffe, 2002). If there is pronounced separation between the groups in the data then it is expected that variables are not highly correlated across groups. This results in higher weights being assigned to these variables which leads to the separation between groups that is seen in the CVA biplot. It is thus not surprising that the Mahalanobis distance should reveal itself in the process of constructing a solution to the problem posed by CVA.

Armed with the means of answering the question CVA poses it is possible to turn attention to the construction of the biplot. The canonical means have rank less than or equal to the minimum of $g - 1$ and p. Furthermore, given that multiplication by a non-singular matrix leaves the rank of a matrix unchanged, the canonical means will have the same rank as the means in the original space. Geometrically this implies that the canonical means can be represented in at most $min(g-1,p)$ dimensions. The objective is to represent the canonical means in ρ dimensions, more specifically in two dimensions. The two dimensional space in which the canonical means are to be represented is given by $\mathcal{L}$. The approximation of the canonical means in $\mathcal{L}$ is simply obtained by considering the first two columns of the transformation matrix $\boldsymbol{M}$, denoted by $\boldsymbol{M}_{[2]}$. Mathematically this is represented as

$$\boldsymbol{Z} = \overline{\boldsymbol{X}}\boldsymbol{M}_{[2]}. \tag{5.6}$$

The fact that the first two columns of $\boldsymbol{M}$ span the subspace $\mathcal{L}$ can be better understood if the process of constructing the CVA biplot is thought of as a two-step process (Gower *et al.*, 2011). The first step concerns itself with finding a non-singular $p \times p$ matrix transformation $\boldsymbol{L}$ so that the Euclidean distances between the group means of the transformed variables is equivalent to the Mahalanobis distance between the original group means. Working from the definition of the Mahalanobis distance as illustrated in (5.5), this implies that the matrix $\boldsymbol{L}$ is such that $\boldsymbol{L}\boldsymbol{L}^{'} = \boldsymbol{W}^{-1}$. Solving the eigenvector equation $\boldsymbol{W}\boldsymbol{L} = \boldsymbol{L}\boldsymbol{\Lambda}$ subject to the the scaling $\boldsymbol{L}\boldsymbol{W}\boldsymbol{L}^{'} = \boldsymbol{I}$ provides a feasible solution for the matrix $\boldsymbol{L}$. In effect this represents the linear transformation to the canonical space. The second step is concerned with constructing the biplot and this is based on performing PCA on the canonical means $\overline{\boldsymbol{X}}\boldsymbol{L}$ and employing the methodology for PCA biplot construction detailed in Chapter 3. The methodology described by Gower and Hand (1996) essentially combines these two steps into a single calculation represented by the two-sided eigenvalue problem. The eigenvalue decomposition used to solve the second step in the process can be represented as

$$(\boldsymbol{L}^{'}\overline{\boldsymbol{X}}^{'}\boldsymbol{C}\overline{\boldsymbol{X}}\boldsymbol{L})\boldsymbol{V} = \boldsymbol{V}\boldsymbol{\Lambda}, \tag{5.7}$$

where $\boldsymbol{C}$ is a centering operation to be discussed shortly. Left-multiplying both sides of (5.7) by $\boldsymbol{L}$ yields

$$(\overline{\boldsymbol{X}}^{'}\boldsymbol{C}\overline{\boldsymbol{X}})(\boldsymbol{L}\boldsymbol{V}) = \boldsymbol{W}(\boldsymbol{L}\boldsymbol{V})\boldsymbol{\Lambda}. \tag{5.8}$$

The matrix $\boldsymbol{M}$ is thus equivalent to $\boldsymbol{L}\boldsymbol{V}$. When the matrix $\boldsymbol{C} = \boldsymbol{N}$ then (5.8) becomes the two sided eigenvalue problem in (5.3) which provided the solution to the problem that CVA seeks to solve. From this perspective it is clear that the solution to the two-sided eigenvalue problem is an amalgamation of the two-step process so that the first two columns of the associated eigenvector matrix, $\boldsymbol{M}$, spans the subspace $\mathcal{L}$ in the canonical space. The construction of the ordinary PCA biplot relies on the $\boldsymbol{V}$ matrix obtained from the SVD of the centred data matrix $\boldsymbol{X}$ and $\boldsymbol{V}$ also diagonalizes $\boldsymbol{X}^{'}\boldsymbol{X}$. For an accurate PCA interpretation the matrix $\boldsymbol{V}$ must meet this requirement yet in this case in does not since it is derived from the matrix $\overline{\boldsymbol{X}}^{'}\boldsymbol{N}\overline{\boldsymbol{X}}$. Redefining $\boldsymbol{B}$ as $\overline{\boldsymbol{X}}^{'}\overline{\boldsymbol{X}}$ and solving (5.3) leads to a different matrix $\boldsymbol{V}$ however the Mahalanobis distance property is preserved with the added benefit that an exact interpretation for the principal components is established.

This opens the discussion on the centering operation $\boldsymbol{C}$ defined in Gower *et al.* (2011). PCA requires that the data matrix be centered so as to ensure that the best fitting plane passes through the centroid of the data. In the context of CVA, there is a choice of weighting the groups means by their sample sizes so that the best fitting plane passes through the weighted centroid. The Malanahobis distance property remains regardless of whether the group means are weighted or not. According to Gower *et al.* (2011). the choice depends on the context and specifies that if the goal is to get a sense of the Mahalaonbis distances between samples then the unweighted form is the better since the weighting will lead to groups comprising larger samples being better represented. The approach adopted here is to consider the unweighted case if only because the biplot is an exploratory tool and one would want a general appreciation of the distances between samples.

5.2.1 CVA biplot axes

Having come to understand the fundamentals of constructing the best fitting plane $\mathcal{L}$ it is now necessary to consider the process of how the original variables are displayed in this space. CVA biplots present the simplest case in which the variable axes for interpolation and prediction are both linear but differ in direction (Gower and Hand, 1996). This is different to the PCA biplot where interpolative and predictive axes differed only in calibration. Interpolation axes are considered first. For the purpose of interpolation, the process of representing the original variables is very similar to that used in the case of PCA biplots. Any sample point $\boldsymbol{x}$ in $\Re^p$ can be represented as $\sum_{k=1}^{p} x_k \boldsymbol{e}_k^{'}$ where $\boldsymbol{e}_k^{'}$ is the $1 \times p$ unit vector in the direction of the k^{th} Cartesian axis in $\Re^p$. Interpolating any point into the space $\mathcal{L}$ is achieved by multiplying by the matrix $\boldsymbol{M}_{[2]}$ so that the interpolated point $\boldsymbol{z}$ is represented as $\sum_{k=1}^{p} x_k \boldsymbol{e}_k^{'}\boldsymbol{M}_{[2]}$. The Cartesian axes are represented by $e_k^{'}\boldsymbol{M}_{[2]}$ in $\mathcal{L}$. This representation is used for the purpose of interpolation. Calibrating the axes is simply a matter of considering $\mu\boldsymbol{e}_k^{'}\boldsymbol{M}_{[2]}$ and plotting the co-ordinates for varying values of μ.

The process of prediction is different and it is important to be cognisant of the fact that the objective is to use the axes in order to find the co-ordinates for the the sample points in the *original space* and not in the canonical space. This means that the objective is to be able to read off the values for the sample point $\boldsymbol{x}^{'}$ in the original space which is represented as $\boldsymbol{y}^{'} = \boldsymbol{x}^{'}\boldsymbol{M}$ in the canonical space. From this expression it is evident that

$$\boldsymbol{x}^{'} = \boldsymbol{y}^{'}\boldsymbol{M}^{-1}. \tag{5.9}$$

The direction for the predictive axes will depend on the rows of the matrix $\boldsymbol{M}^{-1}$ since (5.9) makes it clear that the original observed value $\boldsymbol{x}^{'}$ is represented as a linear combination of the rows of $\boldsymbol{M}^{-1}$. The value on the k^{th} original axis is given by $\boldsymbol{y}^{'}\boldsymbol{M}^{-1}\boldsymbol{e}_k$ making it clear that the directions of the predictive axes will be determined by $\boldsymbol{M}^{-1}\boldsymbol{e}_k$. In order to calibrate the k^{th} predictive axis, a plane $\mathcal{N}$ is constructed so that it is orthogonal to the k^{th} axis in the canonical space at an arbitrary value μ. The equation for $\mathcal{N}$ is given by

$$\mu = \boldsymbol{y}^{'}\boldsymbol{M}^{-1}\boldsymbol{e}_k. \tag{5.10}$$

Any point on the plane $\mathcal{N}$ predicts the value μ for the k^{th} original axis. Assume that the point $\boldsymbol{z}$ lies in the plane $\mathcal{L}$ represented relative to the vectors spanning $\mathcal{L}$ so that $\boldsymbol{z}$ is 2×1. The plane $\mathcal{N}$ intersects the best-fitting plane $\mathcal{L}$ where

$$\mu = \boldsymbol{z}^{'}(\boldsymbol{J}\boldsymbol{M}^{-1})\boldsymbol{e}_k, \tag{5.11}$$

where $\boldsymbol{J}$ is the matrix defined for $\boldsymbol{J}$-notation in Chapter 4. In order to facilitate orthogonal projection onto the predictive biplot axis for the k^{th} variable, the axis will be defined so that it is perpendicular to the line (5.11) which represents the point on the biplot axis that predicts μ for the k^{th} original variable. The direction of the k^{th} predictive biplot axis is thus specified by $\boldsymbol{e}_k^{'}(\boldsymbol{J}\boldsymbol{M}^{-1})^{'}$. The location of the point to be calibrated as μ on the axis is given by

$$\boldsymbol{z}_\mu^{'} = \sigma\boldsymbol{e}_k^{'}(\boldsymbol{J}\boldsymbol{M}^{-1})^{'}. \tag{5.12}$$

In order to find the point in (5.12) on the line of intersection between $\mathcal{L}$ and $\mathcal{N}$, it is a simple matter of substituting (5.12) into (5.11) and solving for σ. This process yields

$$\boldsymbol{z}_\mu^{'} = \frac{\mu}{\boldsymbol{e}_k^{'}(\boldsymbol{J}\boldsymbol{M}^{-1})'(\boldsymbol{J}\boldsymbol{M}^{-1})\boldsymbol{e}_k}\boldsymbol{e}_k^{'}(\boldsymbol{J}\boldsymbol{M}^{-1})^{'}. \tag{5.13}$$

Equation (5.13) represents the location of the point on the k^{th} predictive axis to be labelled as μ. Varying values of μ completes the calibration process.

It is also evident that the predictive and interpolative axis do not share direction albeit that both are linear. This provides the foundations for the construction of the CVA biplot.

5.3 Application to matricised data

The CVA biplot is used predominantly to ascertain the extent of the separation between the groups comprising the data. Gower and Hand (1996) contend that if variables are strongly correlated then this will reveal itself in the biplot, however the angle between variable axes is not directly related to the strength of association between variables. It is also instructive to consider whether any of the resulting CVA biplots will look similar. Denote the matricised data as $\boldsymbol{X}_{tall}$, $\boldsymbol{X}_{avg}$ and $\boldsymbol{X}_{wide}$ for the tall combination, aggregated data and the wide combination respectively. $\boldsymbol{X}_{avg}$ is calculated as $\frac{1}{k}\sum_{i=1}^{k}\boldsymbol{X}_i$ where $\boldsymbol{X}_i$ denotes the i^{th} data set. The separate datasets are not centred before combining into the tall and wide combinations. A moment's thought reveals that $\overline{\boldsymbol{X}}_{avg}$ is equivalent to $\overline{\boldsymbol{X}}_{tall}$. It is thus immediately clear that although $\boldsymbol{W}_{avg}$ and $\boldsymbol{W}_{tall}$ will not be equivalent, $\boldsymbol{B}_{avg}$ and $\boldsymbol{B}_{tall}$ as defined in order to maintain the PCA interpretation will be equivalent. This raises questions about the similarity of the CVA biplots associated with the aggregated and tall data and this will be explored. The next aspect to consider is the extent of the separation between the groups comprising the data, how this changes over occasion and how it is represented in the matricised CVA biplot. All CVA biplots are constructed with predictive variable axes.

Time point 1	Time point 2	Time point 3	Time point 4
$\boldsymbol{\Sigma}$	$\boldsymbol{\Sigma}$	$2\boldsymbol{\Sigma}$	$2\boldsymbol{\Sigma}$
$\boldsymbol{\mu}_1' = (0\ 0\ 0)$	$\boldsymbol{\mu}_1' = (2\ 2\ 2)$	$\boldsymbol{\mu}_1' = (0\ 0\ 0)$	$\boldsymbol{\mu}_1' = (2\ 2\ 2)$
$\boldsymbol{\mu}_2' = (1\ 0\ 0)$	$\boldsymbol{\mu}_2' = (5\ 0\ 0)$	$\boldsymbol{\mu}_2' = (1\ 0\ 0)$	$\boldsymbol{\mu}_2' = (5\ 0\ 0)$
$\boldsymbol{\mu}_3' = (0\ 1\ 1)$	$\boldsymbol{\mu}_3' = (0\ 1\ 1)$	$\boldsymbol{\mu}_3' = (0\ 1\ 1)$	$\boldsymbol{\mu}_3'$=(0 1 1)

Table 5.2: Parameter Values for the Simulation.

$$\boldsymbol{\Sigma} = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{pmatrix}$$

In order to explore this, data were simulated from the multivariate normal distribution. The data comprised 45 observations with scores on three variables at four different occasions, classified into three groups of equal size. This afforded the means to manipulate the variation and degree of separation between the groups at each of the four occasions. Table 5.2 contains the parameters used for the simulation of the data. Figure 5.1 is a representation of the separate CVA biplots constructed for each occasion with occasion 1 in the top left panel, moving in a clockwise fashion to occasion 4 in the bottom left panel. The black markers represent group 1, the red markers represent group 2 and the green markers represent group 3. It is immediately clear that the separation between groups changed with occasions 2 and 4 showing clear separation between the groups. Group 2 is particularly separate from the other two groups. Occasions 1 and 4 show less separation with the observations comprising groups 1 and 2 overlapping on variables 2 and 3. When all three variables are considered simultaneously there is still some modicum of separation between the groups at occasions 1 and 3. Is is worth mentioning that the CVA biplot makes it possible to consider the degree of separation of the groups not only collectively but also on particular variables by projecting the observations onto the variable of interest and studying the differences between groups.

Having come to understand the separation that is inherent in the data it is now possible to explore how this is represented on each of the matricised CVA biplots. Figure 5.2 illustrates the CVA biplot for the aggregated data and as one would expect it is impossible to appreciate the evolution of the extent of the separation between the groups in this plot. The information that can be gleaned from Figure 5.2 is that group 2 is quite different from the remaining groups on all variables since projecting onto the axes reveals that the separation occurs on each of the variables. The within-group variation is also relatively small across each of the groups as evidenced by the fact that points are so clustered around the interpolated means, represented by the unfilled symbols. In an attempt to understand what informs the construction, consider two arbitrary data matrices $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. The within-group and between-group variation matrices for the aggregated data $\boldsymbol{W}_{avg}$ and $\boldsymbol{B}_{avg}$ are calculated as

$$\begin{aligned} \boldsymbol{W}_{avg} &= \frac{1}{4}[\boldsymbol{W}_1 + \boldsymbol{W}_2 + (\boldsymbol{X}_1^{'}\boldsymbol{X}_2 - \overline{\boldsymbol{X}}_1^{'}\overline{\boldsymbol{X}}_2) + (\boldsymbol{X}_2^{'}\boldsymbol{X}_1 - \overline{\boldsymbol{X}}_2^{'}\overline{\boldsymbol{X}}_1)], \\ \boldsymbol{B}_{avg} &= \frac{1}{4}(\overline{\boldsymbol{X}}_1^{'}\overline{\boldsymbol{X}}_1 + \overline{\boldsymbol{X}}_1^{'}\overline{\boldsymbol{X}}_2 + \overline{\boldsymbol{X}}_2^{'}\overline{\boldsymbol{X}}_1 + \overline{\boldsymbol{X}}_2^{'}\overline{\boldsymbol{X}}_2), \end{aligned} \tag{5.14}$$

where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are the within-group variation matrices calculated from $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ respectively. What is striking is that the cross product terms play a role in determining both $\boldsymbol{W}_{avg}$ and $\boldsymbol{B}_{avg}$. In order to gain some sense of

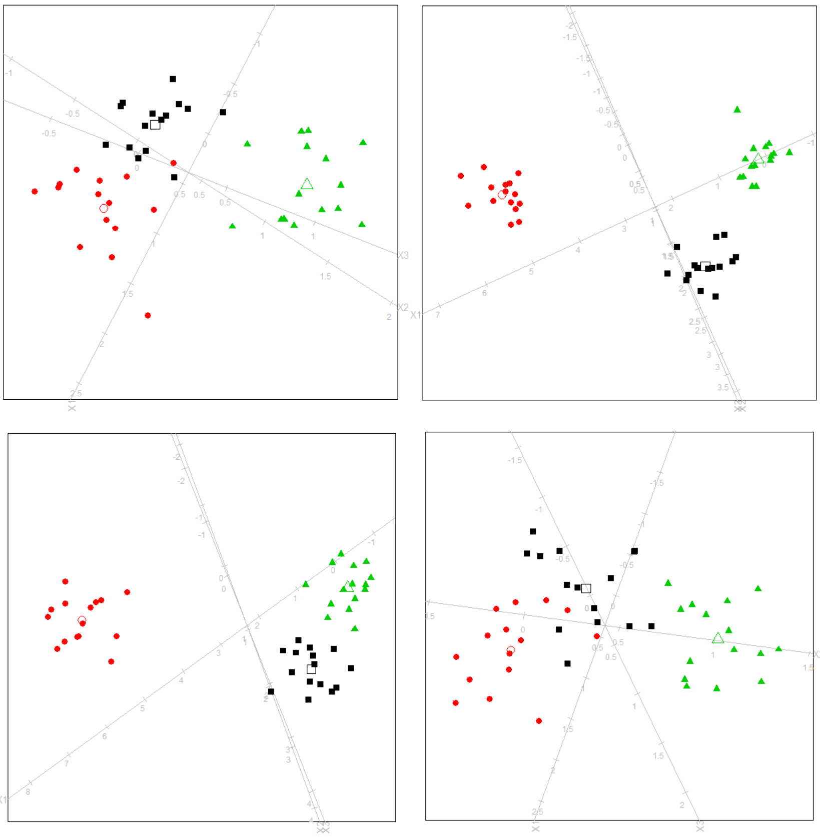


Figure 5.1: Separate CVA biplots for each occasion.

the effect of these cross product terms, the matrices resulting from excluding these terms in (5.14) are computed and the result is compared with $\boldsymbol{W}_{avg}$ and $\boldsymbol{B}_{avg}$. For the simulated data,

$$\boldsymbol{B}_{avg} = \begin{pmatrix} 67.71 & -25.02 & -24.33 \\ -25.02 & 10.35 & 10.23 \\ -24.33 & 10.23 & 10.12 \end{pmatrix} \boldsymbol{W}_{avg} = \begin{pmatrix} 1.68 & 0.22 & 0.63 \\ 0.22 & 1.88 & -0.32 \\ 0.63 & -0.32 & 1.67 \end{pmatrix}. \tag{5.15}$$

The matrices that result from excluding all the cross product terms, denoted by $\boldsymbol{W}^*_{avg}$ and $\boldsymbol{B}^*_{avg}$ are as follows

$$\boldsymbol{B}^*_{avg} = \begin{pmatrix} 24.21 & -6.38 & -6.89 \\ -6.38 & 4.67 & 4.89 \\ -6.89 & 4.89 & 1.58 \end{pmatrix} \boldsymbol{W}^*_{avg} = \begin{pmatrix} 1.63 & 0.11 & 0.37 \\ 0.11 & 1.78 & -0.16 \\ 0.37 & -0.16 & 1.58 \end{pmatrix}. \tag{5.16}$$

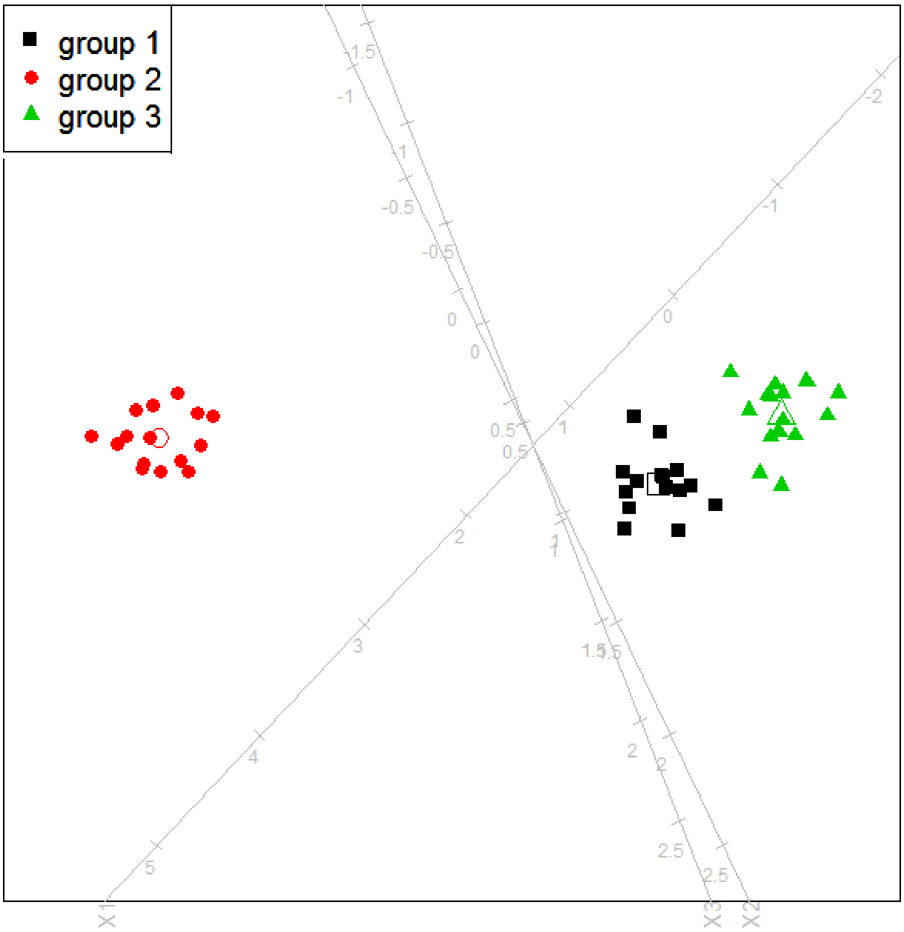


Figure 5.2: CVA biplot for the aggregated Mansoor data.

A comparison of (5.15) and (5.16) reveals that the difference between $\boldsymbol{W}^*_{avg}$ and $\boldsymbol{B}^*_{avg}$ is much more pronounced than the difference between $\boldsymbol{W}_{avg}$ and $\boldsymbol{B}_{avg}$ and this is attributable to the fact that the crossproduct terms seem to make the entries in $\boldsymbol{B}_{avg}$ relatively bigger than those in $\boldsymbol{B}^*_{avg}$. This increased difference could possibly account for the fact that observations are more tightly clustered around their respective group means and that the separation is quite pronounced. A similar result was observed when this comparison was done for a combination of $\boldsymbol{X}_1$ and $\boldsymbol{X}_3$ as well as for $\boldsymbol{X}_2$ and $\boldsymbol{X}_4$. The inclusion of the crossproduct terms tended to exaggerate the difference between $\boldsymbol{B}$ and $\boldsymbol{W}$.

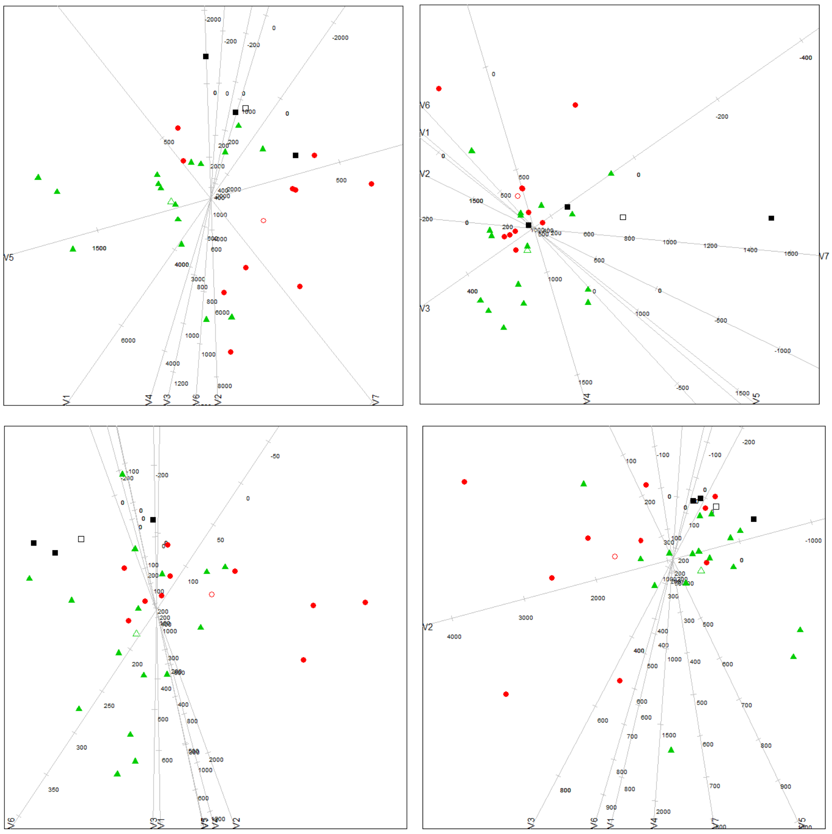


Figure 5.3: Separate CVA biplots for each occasion.

When considering the Mansoor data, Figure 5.3 shows that there is hardly any separation between the groups comprising the Mansoor data and that this is the case across occasion. The within-group variation matrix contained entries much larger than those in the between-variation matrices. Again the crossproduct terms seemed to exaggerate this with the difference between $\boldsymbol{W}$ and $\boldsymbol{B}$ being more pronounced for the aggregated data when compared to the same matrices calculated without the inclusion of the crossproduct terms. Although these are but two numerical examples they seem to suggest that the structure inherent in the data tends to be exaggerated when constructing a CVA biplot for the aggregated data. Figure 5.4, the CVA biplot for the aggregated Mansoor data, captures this structure. It is important to note the fact that because the separation between groups did not vary over time, the CVA biplot for the aggregated data can be deemed useful. This notion is corroborated by examining Figure 5.4 and noticing that the separation reflected is similar to that seen across occasion in Figure 5.3.

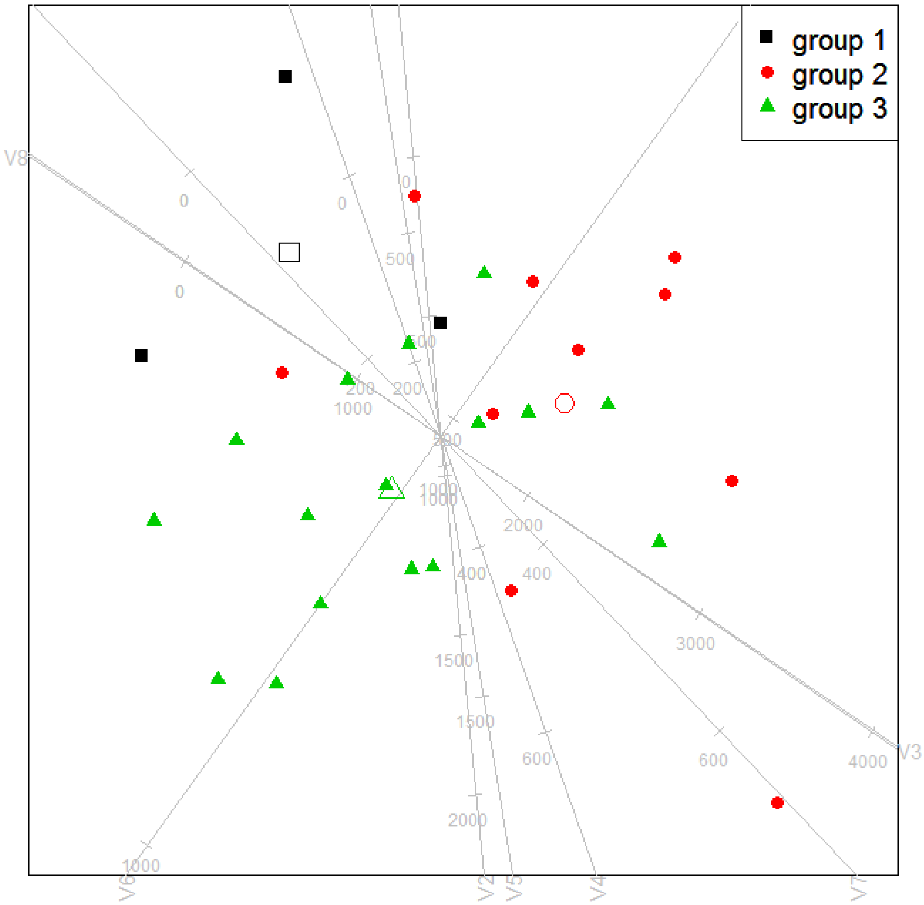


Figure 5.4: CVA biplot for the aggregated Mansoor data.

Attention now turns to examining the CVA biplot resulting from the tall combination of the data. Firstly, the interpolated means represent the overall means and are equivalent to the means determined in the aggregated data case. Furthermore, the directions of the variable axes in Figure 5.5, the tall combination CVA biplot for the simulated data, are similar to those in Figure 5.2. The same can be said for the Mansoor data when Figures 5.4 and 5.7 are compared. This is attributable to the fact that the matrix $\boldsymbol{B}$ used in the second step of the CVA biplot construction process is equivalent for both these cases. The similarity of the variable axes is less notable when the interpolative axes are used although this in not illustrated here. An interesting

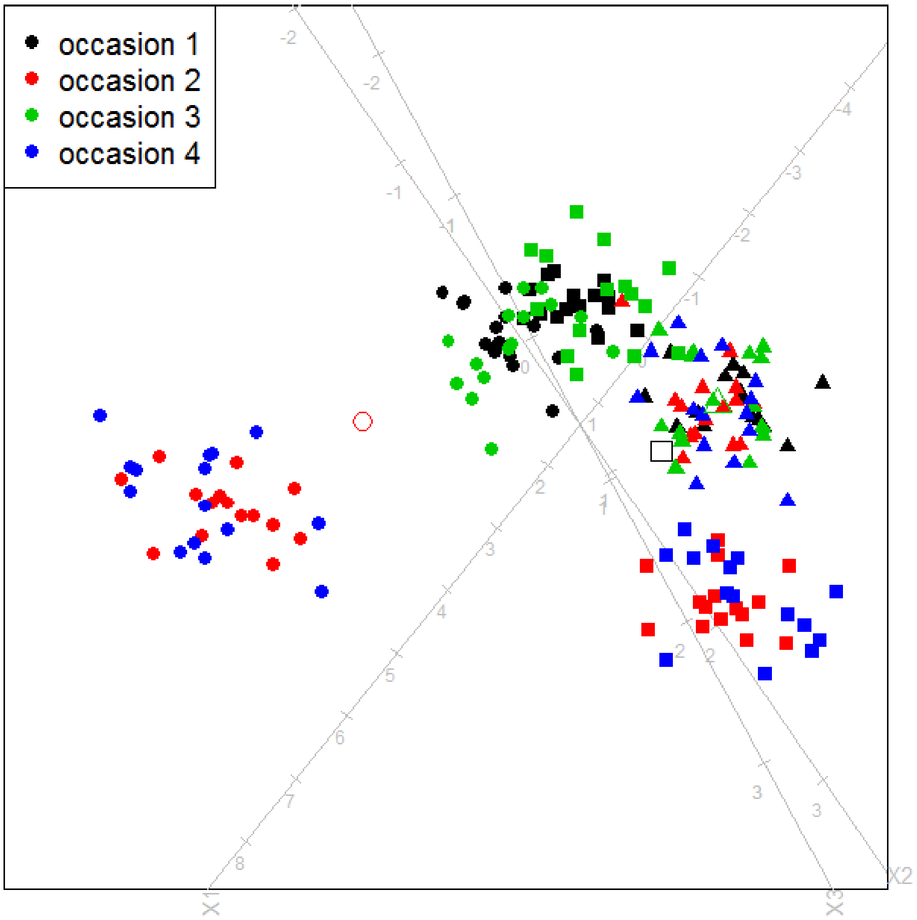


Figure 5.5: CVA biplot for the tall combination of the Simulated data.

observation is made on closer examination of Figure 5.5. It is clear that the change in the separation between the groups is represented here. Occasions 1 and 3 show less separation between groups whilst the separation is marked for occasions 2 and 4. When compared with Figure 5.1 it is clear that the separation between groups at occasions 1 and 3 is less pronounced where as occasions 1 and 4 seem well represented. The tall combination CVA biplot in Figure 5.7 also indicates that the observations are less dispersed than is illustrated in Figure 5.3. The CVA biplot can be used to interpolate new samples for the purpose of classification and this is based on the nearest mean to the interpolated observation (Gower *et.al.*, 2011). Using an overall mean as is done here is not useful for this purpose because the mean evolves over time. Although the tall combination CVA biplot can indicate separa-

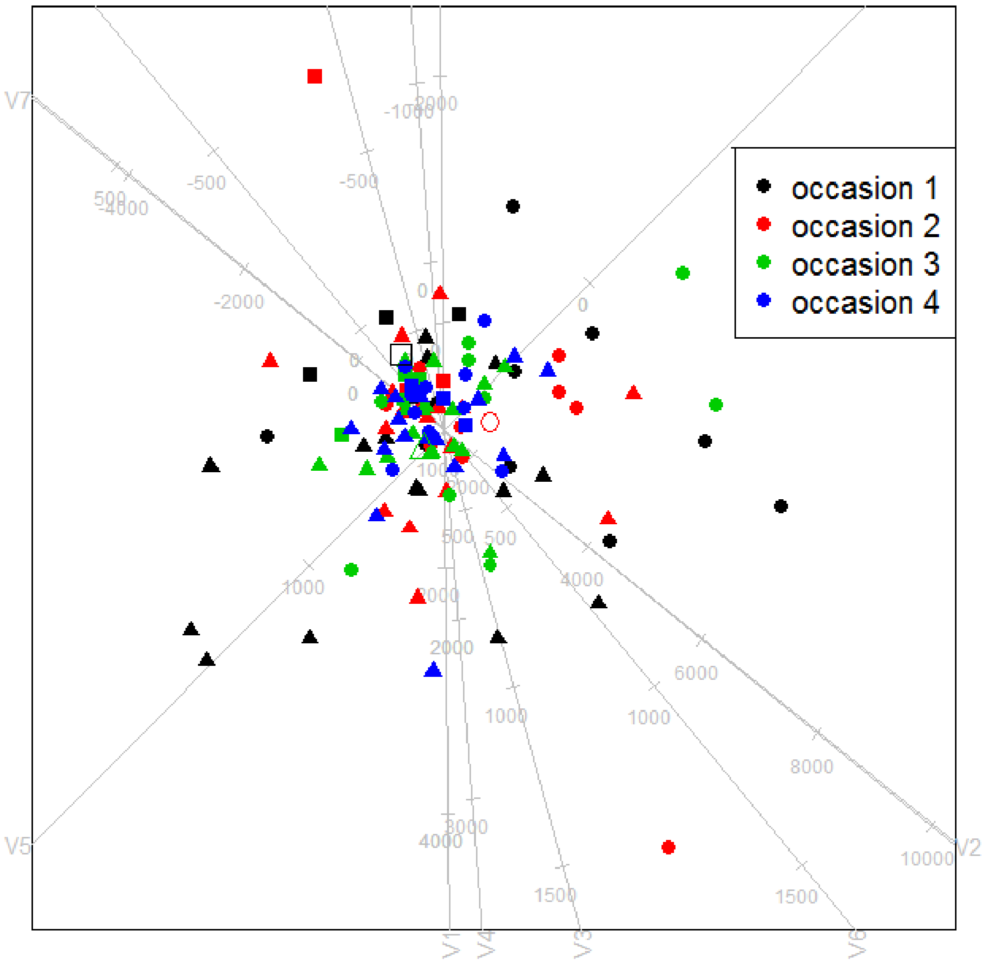


Figure 5.6: CVA biplot for the tall combination of the Mansoor Data.

tion in the data and separate group means can be interpolated in order to visualise the evolution of the group means over time, it cannot serve well to facilitate the classification of new sample points because this would be based on an overall mean.

Finally the wide combination CVA biplot is examined. Figure 5.7, the CVA biplot for the simulated data, indicates that the separation between groups is distinct with observations clustered very closely around the interpolated means. Evidently the wide combination CVA biplot suffers from the same shortcoming as the aggregated data CVA biplot in that the changes in group

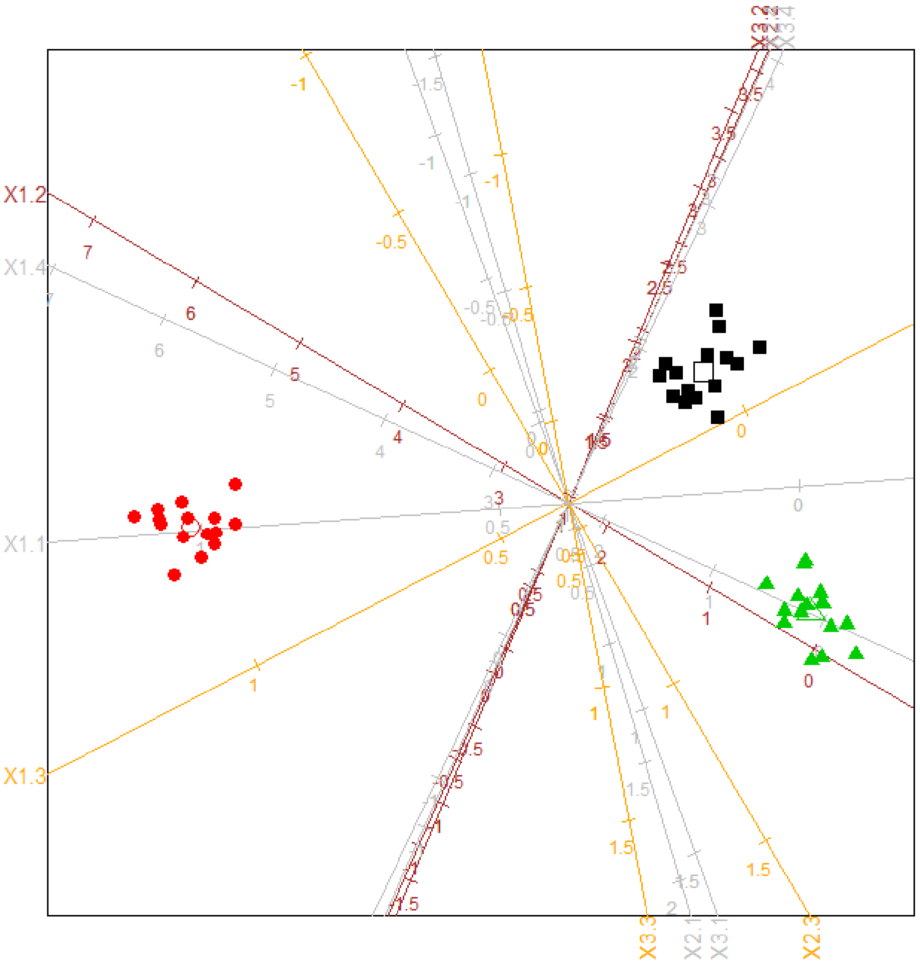


Figure 5.7: CVA biplot for the wide combination of the Simulated data.

separation over time cannot be visualised. When compared to Figure 5.1 the observations are more closely clustered around their respective interpolated means. The wide combination CVA biplot could not be constructed for the complete Mansoor data because there were too few observations for the number of variables thus the first 6 variables at each occasion were used to construct the wide combination data set and consequently the biplot is displayed in Figure 5.8. It can be seen that this biplot overstates the separation between the groups comprising the Mansoor data. A comprehensive reason for this is elusive however the answer may lie in examining the linear transformation to the canonical space, $\boldsymbol{L}$. Consider the case of a single data matrix from the simulated data $\boldsymbol{X}_1$ with corresponding 3×3 matrix of

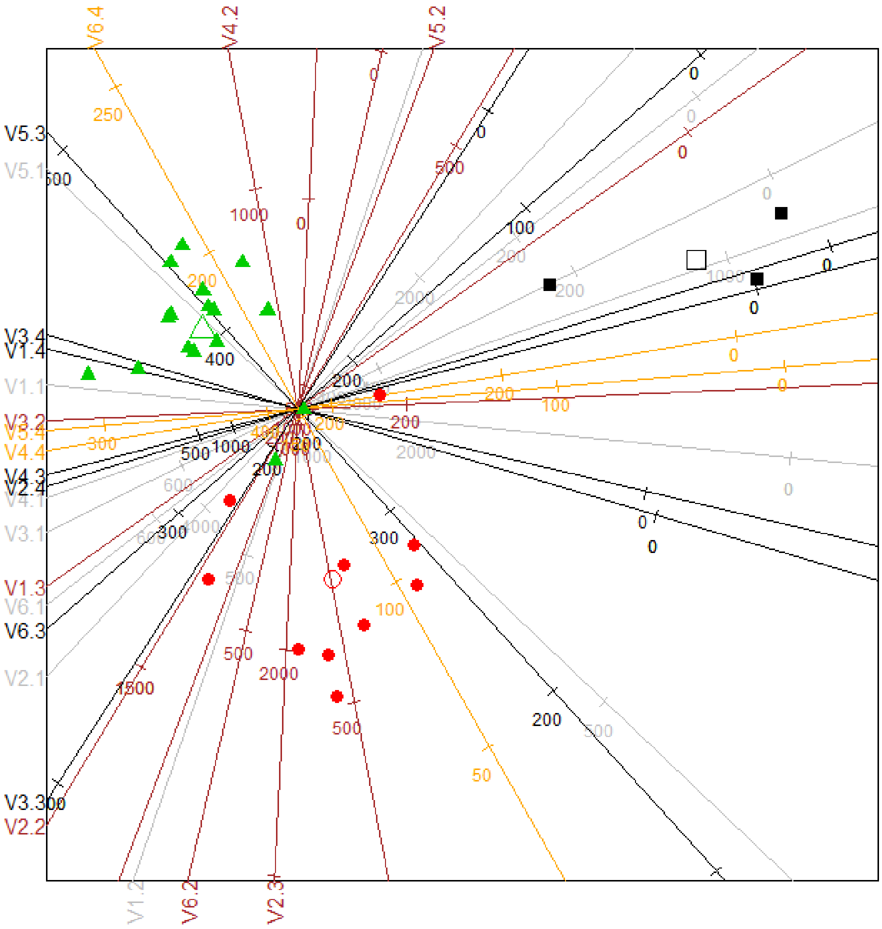


Figure 5.8: CVA biplot for the wide combination of the Mansoor data.

group means $\overline{\boldsymbol{X}}$. The SVD of the transformation matrix $\boldsymbol{L} = \boldsymbol{PDQ}'$ and the transformation to the canonical space is given by $\overline{\boldsymbol{X}}\boldsymbol{PDQ}'$. Since $\boldsymbol{P}$ and $\boldsymbol{Q}$ are orthogonal matrices, geometrically this results in a rotation of the coordinate axes. The diagonal matrix $\boldsymbol{D}$ results in the stretching or shrinking of each of the dimensions. For the purpose of example, consider two data matrices combined into a wide combination matrix $\boldsymbol{X}_{wide} = [\boldsymbol{X}_1 \ \boldsymbol{X}_2]$ with dimensions $n \times 2p$. The SVD of the transformation to the canonical space

$\boldsymbol{L}_{wide}$ can be represented as

$$\boldsymbol{L}_{wide} = \left(\begin{array}{c|c} \boldsymbol{U}_{11} & \boldsymbol{U}_{12} \\ \hline \boldsymbol{U}_{21} & \boldsymbol{U}_{22} \end{array}\right) \left(\begin{array}{c|c} \boldsymbol{D}_{11} & \mathbf{0} \\ \hline \mathbf{0} & \boldsymbol{D}_{22} \end{array}\right) \left(\begin{array}{c|c} \boldsymbol{V}'_{11} & \boldsymbol{V}'_{21} \\ \hline \boldsymbol{V}'_{12} & \boldsymbol{V}'_{22} \end{array}\right) \tag{5.17}$$

$$= \left(\begin{array}{c|c} \boldsymbol{U}_{11}\boldsymbol{D}_{11}\boldsymbol{V}'_{11} + \boldsymbol{U}_{12}\boldsymbol{D}_{22}\boldsymbol{V}'_{12} & \boldsymbol{U}_{11}\boldsymbol{D}_{11}\boldsymbol{V}'_{21} + \boldsymbol{U}_{12}\boldsymbol{D}_{22}\boldsymbol{V}'_{22} \\ \hline \boldsymbol{U}_{21}\boldsymbol{D}_{11}\boldsymbol{V}'_{11} + \boldsymbol{U}_{22}\boldsymbol{D}_{22}\boldsymbol{V}'_{12} & \boldsymbol{U}_{21}\boldsymbol{D}_{11}\boldsymbol{V}'_{21} + \boldsymbol{U}_{22}\boldsymbol{D}_{22}\boldsymbol{V}'_{22} \end{array}\right) \tag{5.18}$$

where each of $\boldsymbol{P}_{wide}$, $\boldsymbol{D}_{wide}$ and $\boldsymbol{Q}_{wide}$ are partitioned into matrices of dimension $p \times p$. It is thus clear that the transformation to the canonical space in this context is quite complex with the terms $\boldsymbol{D}_{11}$ and $\boldsymbol{D}_{22}$ both affecting the transformation and this might account for the fact that the separation is overstated in the context of the wide combination CVA biplot.

5.4 Conclusion

This chapter detailed the basic theory underlying Canonical Variate Analysis and how this technique can be used in the process of constructing a biplot in order to optimally represent the separation between the groups comprising the data in a visual context. The technique was then applied to simulated data as well as to the Mansoor data in order to construct CVA biplots for the matricised forms of these data sets. In general in was determined that regardless of the matricised form of the data being used, the resulting biplot had some deficiencies. Both the aggregated and wide combination CVA biplots did not perform well when the separation between groups changed over time and simply represented the dominant structure in the data. Furthermore, the wide CVA biplot tended to overstate the separation regardless of the structure of the data and the aggregated CVA biplot tended to emphasise the group structure inherent in the data. The tall combination CVA biplot provided a sense of the time profile of the group structure in the data but did not afford to classify observations. The matricised CVA biplots thus did not perform well and it is best to just consider the separate CVA biplots for each occasion. The ideal would be to represent the separate CVA biplots in a single comprehensive biplot where all of the group means are considered in the construction and a novel way to do this is developed in Chapter 8.

Chapter 6

Procrustes Analysis and Biplots

6.1 Introduction

Gower and Dijksterhuis (2004) describe Procrustes Analysis in its simplest form as a technique that seeks to solve the problem concerned with finding a matrix $\boldsymbol{T}$ such that

$$\|\boldsymbol{X}_1\boldsymbol{T} - \boldsymbol{X}_2\|^2, \tag{6.1}$$

is minimised over $\boldsymbol{T}_{p_1 \times p_2}$ for given matrices $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ with dimensions $n \times p_1$ and $n \times p_2$ respectively. Procrustes problems come in various forms depending largely on the constraint that is placed on the transformation matrix $\boldsymbol{T}$. In its application to biplots, $\boldsymbol{T}$ is constrained to be a general rotation or rather an orthogonal matrix.

The name of the techique is attributable to Hurley and Cattell (1962) who used it in the context of relating a factor structure $\boldsymbol{X}_1$ obtained from Factor Analysis to a hypothesised factor structure $\boldsymbol{X}_2$ by means of estimating a transformation matrix $\boldsymbol{T}$ that would transform $\boldsymbol{X}_1$ to fit $\boldsymbol{X}_2$. It is taken from Greek mythology where it is alleged that the murderer Procrustes had an iron bed on which he would place his victims with the aim of ensuring that their bodies fit the length of the bed. In a similar fashion, "$\boldsymbol{X}_1$ is transformed by the matrix $\boldsymbol{T}$ to fit the 'bed' of $\boldsymbol{X}_2$" (Gower and Dijksterhuis, 2004, p. 2). The technique as employed in this context can be traced back even earlier to Mosier (1939).

The focus here is on Orthogonal Procrustes Analysis where the transformation matrix $\boldsymbol{T}$ is required to be an orthogonal matrix $\boldsymbol{Q}$. According to Gower and Dijksterhuis (2004), the solution to this Procrustes problem is attributable to Green (1952) though the assumption was that both $\boldsymbol{X}_1$ and

$\boldsymbol{X}_2$ were both of full column rank. This assumption was relaxed by Schöne (1966). Ultimately it was Gower (1971) who made the first strides in extending the technique for use with K matrices as opposed to just two, $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. This is the technique that is employed in this section.

Generalised Orthogonal Procrustes Analysis (GOPA) is used in this context to effectively superimpose the PCA biplots produced for each of the four separate occasions. The reason for choosing orthogonal transformations should be fairly obvious in light of the fact that each biplot can be though of as a configuration and the aim is really to superimpose each of these configurations optimally without altering them materially. This means that the distances between the sample points as well as the angles between the axes remain unchanged after the rotation.

6.2 GOPA

Before delving into the specifics of Generalised Orthogonal Procrustes Analysis it is instructive to consider the simplest orthogonal Procrustes problem and its solution first. In the simple case, the aim is to find a matrix $\boldsymbol{T}$ such that (6.1) is minimised and $\boldsymbol{T}$ is constrained to be an orthogonal matrix. Orthogonality implies that the inner product of any of the columns of the matrix is equal to zero and the norm of each column is one. Henceforth, the transformation matrix $\boldsymbol{T}$ will be represented by $\boldsymbol{Q}$ to emphasise that it is an orthogonal matrix. Note that $\boldsymbol{Q}$ is a general rotation matrix which includes the possibility of reflection.

Figure 6.1 illustrates the procedure of rotating $\boldsymbol{X}_1$ to best fit $\boldsymbol{X}_2$ where both represent dissimilar triangles. $\mathrm{X}_{11}\mathrm{X}_{12}\mathrm{X}_{13}$ and $\mathrm{X}_{21}\mathrm{X}_{22}\mathrm{X}_{23}$ represent two dissimilar triangles with the same centroid. $\mathrm{X}^1_{21}\mathrm{X}^1_{22}\mathrm{X}^1_{23}$ represents the position of the second triangle after rotating it to best fit the first triangle in accordance with the orthogonal Procrustes criterion. It must be said that Procrustes solution to the problem illustrated in Figure 6.1 employed isotropic scaling. Isotropic scaling refers to the application of scaling factors to entire configurations in order to magnify or reduce its size. The rotated triangle in Figure 6.1 was also scaled to be smaller. Each of the co-ordinates defining the triangle were multiplied by the same scaling factor. Isotropic scaling is not considered in this dissertation. Figure 6.1 a visual idea of how the process works. It is also necessary to consider the mathematical solution to the simple orthogonal Procrustes problem since it is instrumental when considering the solution to the generalised problem.

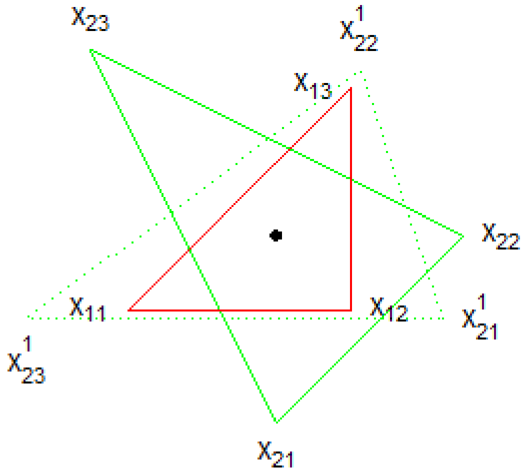


Figure 6.1: Graphical illustration of Orthogonal Procrustes Analysis using two dissimilar triangles

The criterion that is to be minimised can be written as

$$\begin{aligned}\|\boldsymbol{X}_1\boldsymbol{Q}-\boldsymbol{X}_2\| &= tr(\boldsymbol{Q}'\boldsymbol{X}_1'\boldsymbol{X}_1\boldsymbol{Q}+\boldsymbol{X}_2'\boldsymbol{X}_2)-tr(\boldsymbol{X}_2'\boldsymbol{X}_1\boldsymbol{Q})-tr(\boldsymbol{Q}'\boldsymbol{X}_1'\boldsymbol{X}_2)\\ &= tr(\boldsymbol{X}_1'\boldsymbol{X}_1+\boldsymbol{X}_2'\boldsymbol{X}_2)-2tr(\boldsymbol{X}_2'\boldsymbol{X}_1\boldsymbol{Q}). \qquad (6.2)\end{aligned}$$

The first equality follows from the definition of the norm. The second equality relies on the use of a couple of properties of the trace operator, namely that $tr(\boldsymbol{AB}) = tr(\boldsymbol{BA})$ and that $tr(\boldsymbol{A}') = tr(\boldsymbol{A})$. The significance of this result is that the first term does not include $\boldsymbol{Q}$ so in order to minimize (6.2) we simply need to maximize $2tr(\boldsymbol{X}_1'\boldsymbol{X}_1\boldsymbol{Q})$ which simplifies the problem somewhat. Consider the SVD of the matrix $\boldsymbol{X}_2'\boldsymbol{X}_1$ to be $\boldsymbol{U\Sigma V}'$. Using this result leads to the following:

$$\begin{aligned}tr(\boldsymbol{X}_2'\boldsymbol{X}_1\boldsymbol{Q}) &= tr(\boldsymbol{U\Sigma V}'\boldsymbol{Q})\\ &= tr(\boldsymbol{\Sigma V}'\boldsymbol{QU})\\ &= tr(\boldsymbol{\Sigma H})\\ &= \sum_{i=1}^{p}\sigma_{ii}h_{ii},\end{aligned}$$

where $\boldsymbol{H} = \boldsymbol{V}^{'}\boldsymbol{Q}\boldsymbol{U}$ and is orthogonal by virtue of the fact that it is the product of orthogonal matrices. Since all the singular values are positive, a maximum is obtained when each h_{ii} is equal to 1 for $i = 1, \dots, p$. The implication is thus that $\boldsymbol{H}$ is in fact the $p \times p$ identity matrix suggesting that the $\boldsymbol{Q}$ matrix which solves the Procrustes problem is equal to

$$\boldsymbol{Q} = \boldsymbol{V}\boldsymbol{U}^{'}. \tag{6.3}$$

This provides a firm foundation for understanding the solution to the generalised problem and so focus is shifted to defining the orthogonal Procrustes problem in the general sense and discussing the solution. Mathematically the Generalised Orthogonal Procrustes Analysis (GOPA) problem can be stated as follows: GOPA seeks to solve the problem of finding the matrices $\boldsymbol{Q}_1, \dots, \boldsymbol{Q}_k$ such that the norm

$$K\sum_{k=1}^{K} \|\boldsymbol{X}_k\boldsymbol{Q}_k - \boldsymbol{G}\| = \left(\frac{K-1}{K}\right)^2 \sum_{k=1}^{K} \| \boldsymbol{X}_k\boldsymbol{Q}_k - \boldsymbol{G}_k \| \tag{6.4}$$

is minimized over all $\boldsymbol{Q}_k$ which are constrained to be orthogonal matrices,

$$\boldsymbol{G} = K^{-1}\sum_{k=1}^{K}(\boldsymbol{X}_k\boldsymbol{Q}_k) \tag{6.5}$$

and

$$\boldsymbol{G}_k = \frac{1}{K-1}\sum_{i \neq k}^{K}(\boldsymbol{X}_i\boldsymbol{Q}_i). \tag{6.6}$$

The matrices $\boldsymbol{G}$ and $\boldsymbol{G}_k$ are often referred to as the *group-average configuration* and *k-excluded group average configuration* (Gower and Dijksterhuis, 2004). In effect the problem is much like the simple problem but in this instance each of the matrices $\boldsymbol{X}_1, \dots, \boldsymbol{X}_k$ is being transformed to best fit the group-average configuration according to the least squares criterion specified in (6.4).

The problem has no closed form solution and thus it must be solved by means of an algorithm. A simple alternating least squares algorithm can be specified in order to solve the problem which is guaranteed to converge at least to a local minimum. The algorithm is specified in the following way:

1. Initialise $\boldsymbol{G}$, setting it equal to $\boldsymbol{X}_1$

2. Update the current setting of $\boldsymbol{G}$

3. Find $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_k$ for each term in the summation (6.4) with Orthogonal Procrustes Analysis

4. Test for convergence and if it is not attained return to 2.

The importance of the solution to the simple orthogonal problem has yet to be revealed. In the second step of the algorithm, each of the transformation matrices need to be evaluated and this can be done considering the simple case where each $\boldsymbol{X}_k$ is being fit to $\boldsymbol{G}$. As an example, the solution to $\boldsymbol{Q}_1$ comes from the SVD of $\boldsymbol{G}^{'}\boldsymbol{X}_1$ as discussed previously in the context of the simple problem. Consideration is given to proving this assertion and the process begins with understanding the necessary and sufficient conditions for an optimal solution in the simple orthogonal problem. The "necessary condition" can be stated as follows: if $tr(\boldsymbol{X}^{'}_2\boldsymbol{X}_1\boldsymbol{Q})$ is maximised then $\boldsymbol{X}^{'}_2\boldsymbol{X}_1\boldsymbol{Q}$ is a symmetric positive semi-definite matrix (psd). To see this, recall that $\boldsymbol{Q} = \boldsymbol{V}\boldsymbol{U}^{'}$ maximises $tr(\boldsymbol{X}^{'}_2\boldsymbol{X}_1\boldsymbol{Q})$. Substituting for $\boldsymbol{Q}$ implies that $\boldsymbol{X}^{'}_2\boldsymbol{X}_1\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^{'}$. This serves to show that at a maximum, $\boldsymbol{X}^{'}_2\boldsymbol{X}_1\boldsymbol{Q}$ is a symmetric positive semi definite matrix (psd). The "sufficient" condition for optimality is embodied in the converse of the "necessary condition". In order to prove this consider the fact that the spectral decomposition of $\boldsymbol{X}^{'}_2\boldsymbol{X}_1\boldsymbol{Q}$ is given by $\boldsymbol{L}\boldsymbol{\Lambda}\boldsymbol{L}^{'}$ where $\boldsymbol{L}$ is an orthogonal matrix and $\boldsymbol{\Lambda}$ is diagonal with non-negative eigenvalues. It is immediately clear that

$$\boldsymbol{X}^{'}_2\boldsymbol{X}_1 = \boldsymbol{L}\boldsymbol{\Lambda}(\boldsymbol{Q}\boldsymbol{L})^{'}, \tag{6.7}$$

which is the SVD of $\boldsymbol{X}^{'}_2\boldsymbol{X}_1$. It follows that for any arbitrary orthogonal matrix $\boldsymbol{Q}^*$, $tr(\boldsymbol{X}^{'}_2\boldsymbol{X}_1\boldsymbol{Q}^*)$ is maximised when $\boldsymbol{Q}^* = (\boldsymbol{Q}\boldsymbol{L})\boldsymbol{L}^{'} = \boldsymbol{Q}$, showing that the symmetric psd matrix $\boldsymbol{X}_2\boldsymbol{X}^{'}_1\boldsymbol{Q}$ maximises $tr(\boldsymbol{X}_2\boldsymbol{X}^{'}_1\boldsymbol{Q}^*)$ thus proving the "sufficient" condition. Differentiating (6.4) with respect to $\boldsymbol{Q}_k$ yields

$$(\boldsymbol{X}^{'}_k\boldsymbol{X}_k)\boldsymbol{Q}_k - \boldsymbol{X}^{'}_k\boldsymbol{G}_k = \boldsymbol{\Lambda}_Q, \tag{6.8}$$

where $\boldsymbol{\Lambda}_Q$ is a Langrangian term expressing the orthogonality constraint on $\boldsymbol{Q}_k$. It can be shown that $\boldsymbol{\Lambda}_Q = \boldsymbol{Q}_k\boldsymbol{\Lambda}$ where $\boldsymbol{\Lambda}$ is a symmetric matrix and this follows from the orthogonality of $\boldsymbol{Q}_k$ (Gower and Dijksterhuis, 2004). Pre-multiplying (6.8) by $\boldsymbol{Q}^{'}_k$ gives

$$\boldsymbol{Q}_k(\boldsymbol{X}^{'}_k\boldsymbol{X}_k)\boldsymbol{Q}_k - \boldsymbol{Q}^{'}_k\boldsymbol{X}^{'}_k\boldsymbol{G}_k = \boldsymbol{\Lambda}. \tag{6.9}$$

Equation (6.8) shows that $\boldsymbol{Q}^{'}_k\boldsymbol{X}^{'}_k\boldsymbol{G}_k$ must be symmetric and thus by virtue of the necessary condition $\boldsymbol{Q}^{'}_k$ is derived from the SVD of $(\boldsymbol{X}^{'}_k\boldsymbol{G}_k)$. It is thus obvious that $\boldsymbol{Q}_k$ is derived from the SVD of $(\boldsymbol{G}^{'}_k\boldsymbol{X}_k)$. Furthermore, by equation(6.4) replacing $\boldsymbol{G}_k$ with $\boldsymbol{G}$ is acceptable since the solution $\boldsymbol{Q}_k$

not only provides the best fit to the the group average $\boldsymbol{G}$ but also to $\boldsymbol{G}_k$. This justifies the simple alternating least squares algorithm used to find the optimal solution. Convergence of the algorithm is guaranteed to a local minimum (Gower and Dijksterhuis, 2004).

Two issues need to be addressed before discussing the biplot construction *viz.* data preprocessing and translation. Each will be discussed in turn beginning with matter of data preprocessing which is done in the following way:

$$x^*_{ijk} = \frac{x_{ijk} - \overline{x}_{.j.}}{s_{.j.}},$$

where $\overline{x}_{.j.} = \frac{1}{nk}\sum_i \sum_k x_{ijk}$ and $s_{.j.} = \frac{1}{nk-1}\sum_i \sum_k (x_{ijk} - \overline{x}_{.j.})^2$. If the data are arranged in an array, these quantities represent the mean and standard deviation calculated by for the values on the lateral slice obtained by slicing the array across variable j where $j = 1, \ldots, p$. This form of preprocessing, slice centering and scaling is mentioned by Kroonenberg (2008). Gower and Dijksterhuis (2004) speak of the importance of ensuring that variables are commensurate before employing the Procrustes technique. It is thus important to ensure that variables are commensurate across the occasion mode but that it is still possible to see the change in variability across occasions. Choosing the slice transformation ensures that changes in variation in the data can still be captured in the biplot. Had each matrix $\boldsymbol{X}_k$ been standardised it would not have been possible to see the spread of the sample points change over time since the variance on all variables would be one. One must be weary of the fact that PCA is not scale invariant implying that the inclusion of $s_{.j.}$ in the preprocessing will affect the orientation of the best-fitting plane. This results in the seperate PCA biplots looking different to those that would have been obtained using the raw data. It does serve to mitigate the phenomenon of variables showing considerable variation dominating the orientation of the best fitting plane and can result in relationships between variables being better approximated. Full detail on the issue of scale invariance can be found in Gower *et al.* (2011). This justifies the choice of preprocessing and attention shifts to ensuring that PCA biplots are comparable after rotation.

Define $\boldsymbol{Z}_{k(2)}$ and $\boldsymbol{V}_{k(2)}$ to be the two dimensional representations of the sample points and variable points in the best-fitting plane respectively. The GOPA technique can be applied to the sample point representations $\boldsymbol{Z}_k$, the variable axes representations $\boldsymbol{V}_k$ or a combination of the two $\boldsymbol{A}_k^{'} = [\boldsymbol{Z}_k^{'}\boldsymbol{V}_k^{'}]$. Cox and Cox (2008) show that before determining the rotation, it is optimal

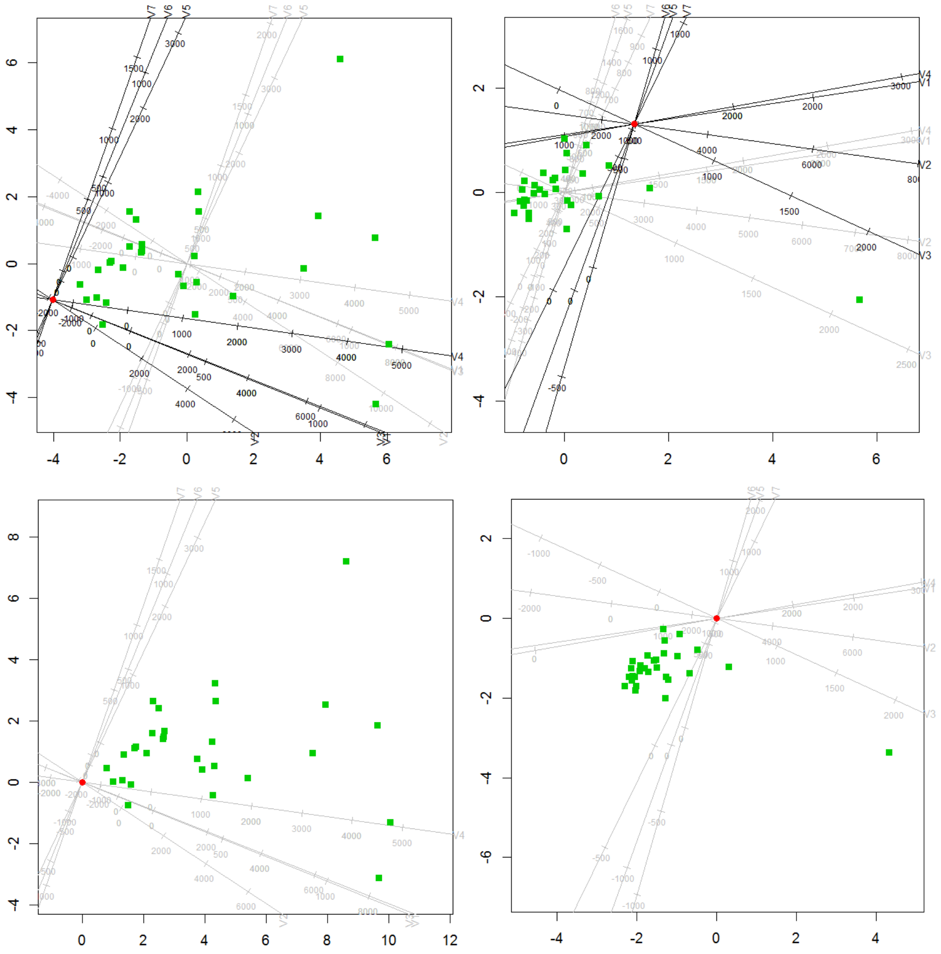


Figure 6.2: Illustration of OPT and translation of biplots for occasions 1 on the left and occasion 4 on the right.

to have the configurations translated so that they have their centroids at the origin. Since this occurs in the process of PCA, no translation is performed on the configurations before determining the rotation matrices $\boldsymbol{Q}_k$. Once these matrices have been determined and applied accordingly, what remains to be done is to superimpose the optimally rotated PCA biplots for each of the k occasions. Figure 6.2 will aid in describing this process which begins with interpolating the mean vector $(\overline{x}_{.1.}, \ldots, \overline{x}_{.p.})$ onto each of the k PCA biplots after the application of the rotation matrices. The interpolated mean is defined as

$$\boldsymbol{z}'_{k(2)} = [\frac{\overline{x}_{.1.} - \overline{x}_{.1.}}{s_{.1.}} - \overline{x}^*_{.1k}, \ldots, \frac{\overline{x}_{.p.} - \overline{x}_{.p.}}{s_{.p.}} - \overline{x}^*_{.pk}]\boldsymbol{V}_{k(2)}\boldsymbol{Q}_k,$$

where $\overline{x}^*_{.jk} = \frac{1}{n}\sum_{l=1}^{n} x^*_{ljk}$. This interpolated mean is represented by the red marker in each of the panels comprising Figure 6.2. The variable axes in each of the k PCA biplots undergo orthogonal parallel translation(OPT) such that the axes coincide in the interpolated mean. This is illustrated in the top left and right panel of Figure 6.2. The grey axes represent the original variable axes passing through the centroid for occasion 1 and occasion 4 respectively. The black variable axes result after application of OPT and it is evident that these axes pass through the interpolated mean. OPT entails moving each of the variable axes parallel to themselves whilst ensuring that the markers are moved in a consisent fashion. This means that any line joining the same marker on each of the parallel variable axes be orthogonal to these axes. The next step in the process requires that the scaffolding axes for each of the k PCA biplots be moved so that the interpolated point is at the origin $(0, 0)$. This step effectively sets the k rotated PCA biplots on one another. The k sets of scaffolding axes are then used to produce the combined PCA biplot of all k occasions. In order to accomplish this, $\boldsymbol{z}^{'}_{k(2)}$ is subtracted from each of the rows of $\boldsymbol{Z}_k\boldsymbol{Q}_k$ to produce $T(\boldsymbol{Z}_k\boldsymbol{Q}_k)$ as well as from the translated variable axes representations for the k^{th} occasion to produce $T(\boldsymbol{V}_k\boldsymbol{Q}_k)$. The effect of this translation is shown in the bottom left and right panels of Figure 6.2. Notice that when compared to the top panels, the red marker in the bottom panels coincide with the $(0, 0)$ point of the scaffolding axes. These quantities are then used in order to construct the combined PCA biplot. It is this process that ensures that the differences in means over occasion can be visualised in the final plot.

A final matter of interest before moving to the application is the interpretation of the biplots that result from employing this technique. More specifically the question of import is what visual information can this biplot provide about changes over time in the samples and variables. Since a biplot conveys information about the strength of association between variables as well as the distribution of sample points, each is considered in turn. Suppose that the biplot is constructed by fitting sample points optimally. Recall that GOPA works on the basis of fitting each of the k configurations to a group-average configuration, $\boldsymbol{G}$. This means that observations for the first subject at each occasion are going to be as close as possible to the representation of observation one in the group-average configuration, for example. The implication is that visually such a plot will give a distorted image of the Euclidean distances between the representations for subject one over time and as such is not the plot to use when seeking an accurate visual representation of the time profile for subjects. The PCA biplot that results from fitting the variable axes optimally is not necessarily better in this regard particularly for sample points that are very close to the variable axes. In fitting the variable axes optimally, variable axis one at each occasion will be rotated to be as close as possible to the group configuration representation of variable axis one, for example. If a particular observation is close to variable axis one across occasion then the Euclidean distances between the representations of this observation at each occasion will be distorted. Furthermore, the fact that the mean changes over time means that variable axes are calibrated differently across time and sample points that are close together in the combined PCA biplot might not have similar scores in reality. It can thus be argued that the combined PCA biplot does not give an accurate representation of the Euclidean distances between sample points across time. It is possible to see how the relationship between particular sample points, one and two for example, changes over occasion. Similarly it is only possible to say how the strength of association between particular variables changes over occasion. The PCA biplot also does not convey information regarding how variable one is related to itself across occasion for example. It is possible however to see how the variation in the data changes over time. The plot also preserves the separation between the data across occasion so that it is possible to get a sense of how the mean has changed over occasion.

The PCA biplot can be constructed so that either the Euclidean distances between sample points or the correlations between the variables are optimally represented and there are three possible configurations, $\boldsymbol{Z}_{k(2)}$, $\boldsymbol{V}_{k(2)}$ and $\boldsymbol{A}_{k(2)}$ to which GOPA can be applied. This implies that there are six possible Procrustes biplots that can be constructed. Furthermore, the biplots constructed using $\boldsymbol{A}_{k(2)}$ and $\boldsymbol{Z}_{k(2)}$ are likely to look very similar. This is because there are often more observations than variables in each of the datasets thus the observations will make a much bigger contribution to the least squares convergence criterion. In this chapter attention has only been given to the Procrustes biplot in which Euclidean distances between sample points is optimally represented although the the technique can be applied to biplots in which the correlation between variables is optimally represented.

6.3 Application to Mansoor data

Given the nature of the investigation undertaken by Mansoor *et al.* (2009), the PCA biplot in which the distance between sample points is optimally represented is studied given that the researchers were largely concerned with variation between group members. In the first instance, the sample points,

$\boldsymbol{Z}_{k(2)}$ are rotated to fit optimally so that the criterion to be minimised is

$$\min_{\boldsymbol{Q}_1 \ldots \boldsymbol{Q}_k} 4 \sum_{k=1}^{4} \|\boldsymbol{Z}_{k(2)}\boldsymbol{Q}_k - \boldsymbol{G}\|, \tag{6.10}$$

where

$$\boldsymbol{G} = \frac{1}{4}\sum_{k=1}^{4}(\boldsymbol{Z}_{k(2)}\boldsymbol{Q}_k). \tag{6.11}$$

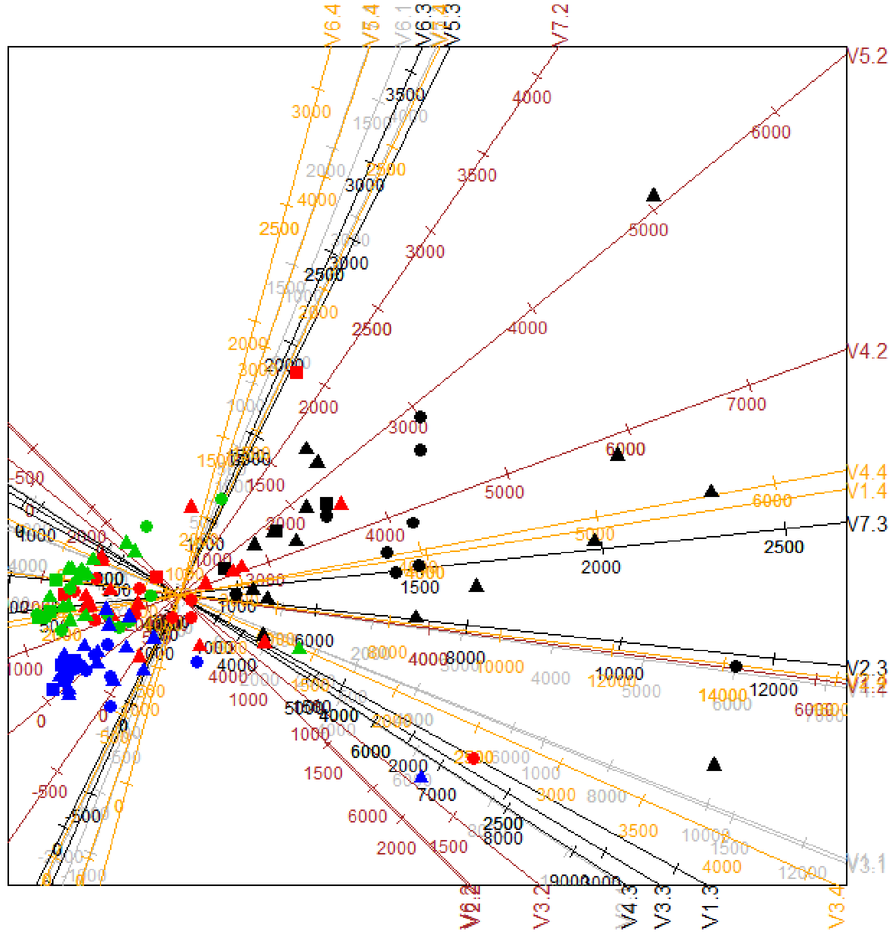


Figure 6.3: Procrustes PCA biplot in which sample points have been optimally fitted.

Figure 6.3 shows the resulting biplot. The most striking feature of the plot is the fact that variation within the data decreases from time one to time

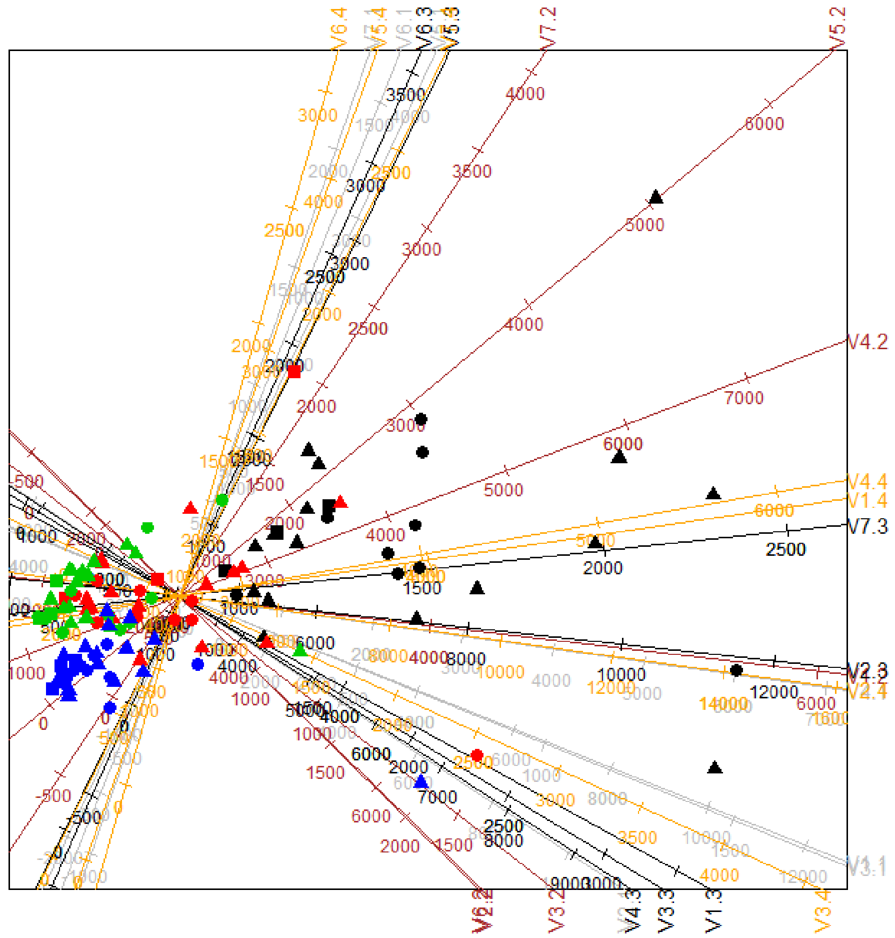


Figure 6.4: Procrustes PCA biplot in which both sample points and variable axes have been optimally fitted.

four with the reduction from time one to time two being particularly noticeable. The separation of the sample points across occasion is also palpable with occasion one having the largest mean. It is also clear that the members comprising the group of HIV^+ infants scored relatively low on all variables across occasion particularly on variable 4. The distribution of members comprising the remaining two groups is quite similar across time. Although a visual appraisal of the Euclidean distances between sample points is not accurate it is possible to read off the scores for sample points to get an idea of how an observation has evolved over time. It is clear that the scores for all observations tended to decrease over time. The next aspect to consider is how the strength of association between variables is represented. Taking variables 1 and 4 for example it can be seen that the variables display relatively strong association at occasions 3 and 4 with relatively weaker association at occasions one and two. This type of comparison can be done for all variables. The last aspect of the application is to ascertain whether the data corroborates the thought that the biplot constructed by optimally fitting the samples should look similar to that constructed by optimally fitting both samples and variables. The latter biplot is illustrated in Figure 6.4 and comparing this to Figure 6.3 reveals that these two biplots are indeed remarkably similar.

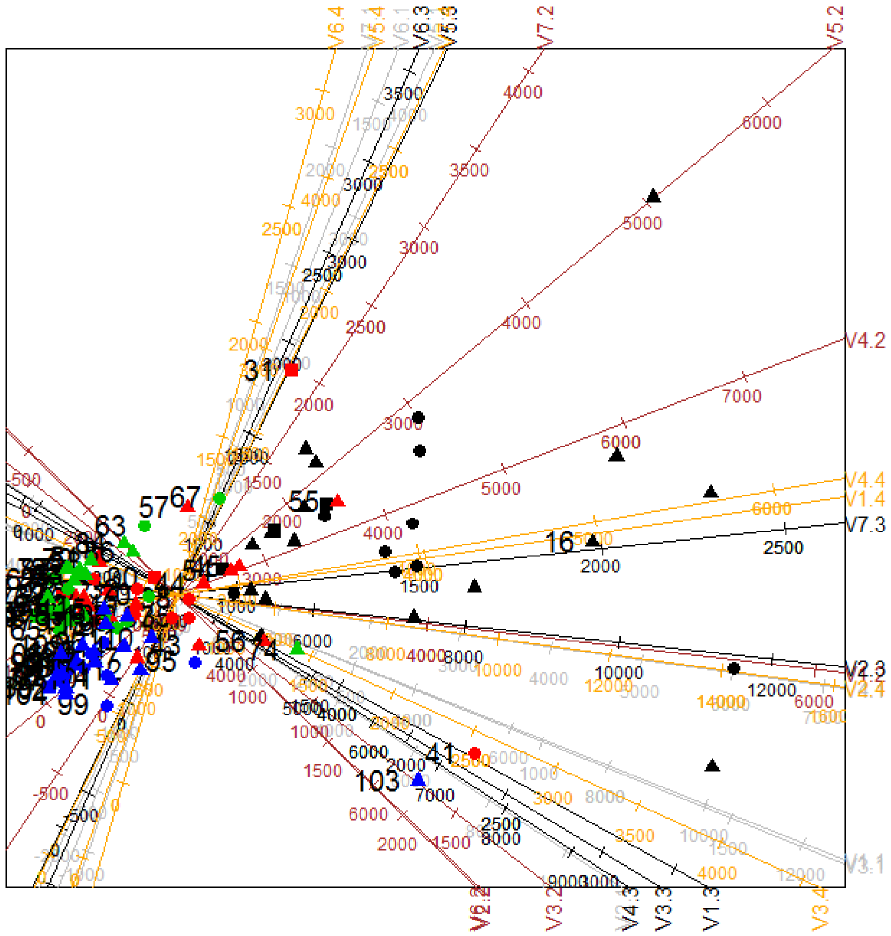


Figure 6.5: Procrustes PCA biplot in which both sample points optimally fitted and labelled.

The last aspect to consider is whether the Euclidean distances between observations across occasions are not well represented. Table 6.1 contains the

	Sample 41	Sample 74
Sample 103	30.48	16.94

Table 6.1: Euclidean distances between observations.

Euclidean distances between some of the observations. Studying Figure 6.5 reveals that the Euclidean distances are indeed not well represented. Although observation 74 is closer in proximity to observation 103 than observation 41, the biplot does not represent this. The implication is that Euclidean distances between points across occasions are indeed distorted in the combined display.

6.4 Conclusion

This chapter detailed the application of GOPA to PCA biplots. The theoretical foundation of GOPA was detailed as well as the process of constructing the combined PCA biplot. Although six possible biplots could be constructed attention was only given to the one in which Euclidean distances were optimally represented. The biplots constructed from optimally fitting $\boldsymbol{A}_{k(2)}$ and $\boldsymbol{Z}_{k(2)}$ were shown to be very similar. It was established that the combined biplot did not provide an accurate visual appraisal of the Euclidean distances between sample points across occasion but afforded the means to read off scores for observations across time. Furthermore, the plot immediately gives a sense of the separation in the data across time as well as changes in variation. In terms of the application to the Mansoor data it was seen that variation in the data reduced over occasion. Furthermore, it was seen that across all occasions the groups comprising the uninfected infants scored better than those comprising the infected group. The plot thus serves well as an exploratory tool.

Chapter 7

Common Principal Components Biplots

7.1 Introduction

Common Principal Components Analysis (CPC) is a technique introduced by Flury (1988) and generalizes PCA to several groups. The CPC model was in fact motivated by biometrical applications where it is commonplace to observe a pattern of similar principal components derived from covariance matrices for different species that have different variances (Neuenschwander and Flury, 2000). The idea is to take the covariance matrices of k independent groups, $\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_k$ and find an orthogonal matrix that simultaneously diagonalises the matrices so that $\boldsymbol{\Psi}_i = \boldsymbol{B}\boldsymbol{\Lambda}_i\boldsymbol{B}'$. An extension of this technique is described by Flury and Neuenschwander (2000) to include instances where the assumption of independence between the k groups is violated. One such case is that of repeated measures studies where p measurements are taken on the same subject over k different time points as is the case in the Mansoor *et al.* (2009) investigation. This chapter is something of an interlude and begins by discussing the theoretical foundations of CPC and CPC for dependent random vectors (DCPC). Its inclusion at this stage is two fold. CPC can be considered a three mode technique and it is thus interesting to consider whether the technique lends itself to constructing a biplot and how such a biplot should be interpreted. A biplot constructed from the CPC technique affords the means to produce a parsimonious display which comprises p variable axes as opposed to kp variable axes as is the case when Procrustes Analysis was used in the construction of biplots. This can be very useful especially when both k and p tend to be large. Furthermore, this technique is critical in a later novel development allowing the display of k

CVA biplots on a single plot.

7.2 Common Principal Components Analysis

7.2.1 Foundations of CPC

CPC seeks to find an orthogonal matrix that simultaneously diagonalises covariance matrices $\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_k$ for each of the k independent groups so that

$$\boldsymbol{\Psi}_i = \boldsymbol{B}\boldsymbol{\Lambda}_i\boldsymbol{B}^{'}. \tag{7.1}$$

Maximum Likelihood estimation is the technique employed in determining the matrix $\boldsymbol{B}$ and as such an assumption needs to be made regarding the distribution from which the data is drawn. It is thus assumed that the p variate random vectors $\boldsymbol{X}_i$ are independently distributed as $N_p(\boldsymbol{\mu}_i, \boldsymbol{\Psi}_i)$ where $\boldsymbol{\mu}_i \,\epsilon\, \Re^p$ and $\boldsymbol{\Psi}_i$ is a positive definite symmetric covariance matrix. The focus is on $\boldsymbol{\Psi}_i$ since the CPC model is concerned with the covariance matrices. Assuming a sample of size n, the covariance matrices can be represented by the sample covariance matrices $\boldsymbol{S}_i$ due to the fact that it is a sufficient statistic for the covariance matrix. Furthermore, $(n-1)\boldsymbol{S}_i$ follows a Wishart Distribution with $n-1$ degrees of freedom. This makes it possible to construct the common likelihood function of $\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_k$ given $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_k$ as

$$L(\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_k) = C \times \prod_{i=1}^{k} etr\big(-\frac{n}{2}\boldsymbol{\Psi}_i^{-1}\boldsymbol{S}_i\big)|\boldsymbol{\Psi}_i|^{-\frac{n}{2}}, \tag{7.2}$$

where *etr* represents the natural exponent of the trace. Flury (1984) suggests that instead of maximising the likelihood function, the function $g(\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_k)$ be minimised. This function is defined as

$$\begin{aligned} g(\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_k) &= -2logL(\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_k) + 2logC \\ &= \sum_{i=1}^{k} n_i(log|\boldsymbol{\Psi}_i| + tr(\boldsymbol{\Psi}_i^{-1}\boldsymbol{S}_i). \end{aligned} \tag{7.3}$$

Since the log likelihood is multiplied by -2 the problem becomes one of minimising the function. The addition of $2logC$ removes the constant term from the log likelihood function so that the function to be minimised depends only on the parameters of interest. Equation (7.3) must be altered so that the matrix $\boldsymbol{B}$ is included since this is what is to be estimated. Each term comprising (7.3) is considered in turn. Assume that (7.1) holds for some matrix

$\boldsymbol{B}$. It is a well known fact that given an $n \times n$ matrix $\boldsymbol{A}$ with eigenvalues $\lambda_1, \ldots, \lambda_n$, the determinant can be calculated as $det(\boldsymbol{A}) = \prod_{i=1}^{n} \lambda_i$. This implies that

$$log|\boldsymbol{\Psi}_i| = \sum_{i=1}^{p} log\lambda_{ij}, \qquad i = 1, \ldots, k. \tag{7.4}$$

Using the fact that $tr(\boldsymbol{AB}) = tr(\boldsymbol{BA})$, it is a simple matter to show that

$$\begin{aligned} tr(\boldsymbol{\Psi}_i^{-1}\boldsymbol{S}_i) &= tr(\boldsymbol{B}\boldsymbol{\Lambda}_i^{-1}\boldsymbol{B}^{'}\boldsymbol{S}_i) = tr(\boldsymbol{\Lambda}_i^{-1}\boldsymbol{B}^{'}\boldsymbol{S}_i\boldsymbol{B}) \\ &= \sum_{j=1}^{p} \frac{\boldsymbol{b}_j^{'}\boldsymbol{S}_i\boldsymbol{b}_j}{\lambda_{ij}}, \qquad i = 1, \ldots, k \end{aligned} \tag{7.5}$$

where $\boldsymbol{b}_j$ represents the j^{th} column of the matrix $\boldsymbol{B}$. Substituting (7.4) and (7.5) into (7.3) results in

$$g(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_p, \lambda_{11}, \ldots, \lambda_{1p}, \lambda_{21}, \ldots, \lambda_{kp}) = \sum_{i=1}^{k} n_i \Big(\sum_{j=1}^{p} (log\lambda_{ij} + \frac{\boldsymbol{b}_j^{'}\boldsymbol{S}_i\boldsymbol{b}_j}{\lambda_{ij}}) \Big). \tag{7.6}$$

Equation (7.6) is to be minimised subject to the constraint that the matrix $\boldsymbol{B}$ is orthogonal. Mathematically this is represented as

$$\begin{aligned} \boldsymbol{b}_h^{'}\boldsymbol{b}_j &= 0 \qquad \text{if } h \neq j \\ \boldsymbol{b}_h^{'}\boldsymbol{b}_j &= 1 \qquad \text{if } h = j. \end{aligned} \tag{7.7}$$

Since the estimation of $\boldsymbol{B}$ is a constrained optimisation problem, the solution lies in constructing the Langrangian function and minimising this function which takes the form

$$G(\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_k) = g(\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_k) - \sum_{h=1}^{p} \gamma_h(\boldsymbol{b}_h^{'}\boldsymbol{b}_h - 1) - 2\sum_{h<j}^{p} \gamma_{hj}\boldsymbol{b}_h^{'}\boldsymbol{b}_j. \tag{7.8}$$

Incorporating the constraints requires the inclusion of $\frac{p(p+1)}{2}$ Langrange multipliers. Differentiating with respect to the λ_{ij} and setting the result equal to zero yields

$$\lambda_{ij} = \boldsymbol{b}_j^{'}\boldsymbol{S}_i\boldsymbol{b}_j \qquad i = 1, \ldots, k, \; j = 1, \ldots, p. \tag{7.9}$$

This result, when combined with (7.5) implies that $tr(\boldsymbol{\Psi_i}^{-1}\boldsymbol{S}_i) = p$. Taking the partial derivative with respect to $\boldsymbol{b}_h^{'}$ yields

$$\sum_{i=1}^{k} n_i \frac{\boldsymbol{S}_i\boldsymbol{b}_j}{\lambda_{ij}} - \gamma_j\boldsymbol{b}_j - \sum_{\substack{h=1 \\ h\neq j}}^{p} \gamma_{jh}\boldsymbol{b}_h = 0 \qquad j = 1, \ldots, p. \tag{7.10}$$

where $\gamma_{jh} = \gamma_{hj}$ if $h > j$. Premultiplying by $\boldsymbol{b}_j^{'}$ and remembering the orthogonality constraint together with (7.9) leads to

$$\gamma_j = \sum_{i=1}^{k} n_i \qquad j = 1, \ldots, p. \tag{7.11}$$

Substituting (7.11) into (7.10) yields

$$\sum_{i=1}^{k} n_i \frac{\boldsymbol{S}_i \boldsymbol{b}_j}{\lambda_{ij}} - \Big(\sum_{i=1}^{k} n_i\Big)\boldsymbol{b}_j - \sum_{\substack{h=1 \\ h \neq j}}^{p} \gamma_{jh}\boldsymbol{b}_h = 0 \qquad j = 1, \ldots, p. \tag{7.12}$$

Now consider premultiplying (7.12) by $\boldsymbol{b}_l^{'}$ where $l \neq j$. Due to the orthogonality constraint this yields

$$\sum_{i=1}^{k} n_i \frac{\boldsymbol{b}_l^{'} \boldsymbol{S}_i \boldsymbol{b}_j}{\lambda_{ij}} = \gamma_{jl} \qquad j = 1, \ldots, p,\ l \neq j. \tag{7.13}$$

At this stage it is necessary to consider the effect of interchanging the indices j and l so that j is replaced with l and vice versa in (7.13). Noting that $\boldsymbol{b}_l^{'}\boldsymbol{S}_i\boldsymbol{b}_j = \boldsymbol{b}_j^{'}\boldsymbol{S}_i\boldsymbol{b}_l$ as well as the fact that $\gamma_{jl} = \gamma_{lj}$, interchanging the indices implies that

$$\sum_{i=1}^{k} n_i \frac{\boldsymbol{b}_l^{'} \boldsymbol{S}_i \boldsymbol{b}_j}{\lambda_{il}} = \gamma_{jl} \qquad l = 1, \ldots, p,\ j \neq l. \tag{7.14}$$

The final step in determining the system of equations to solve in order to estimate the matrix $\boldsymbol{B}$ is simply to equate (7.13) and (7.14) which produces the system of equations

$$\boldsymbol{b}_l^{'} \sum_{i=1}^{k} \Big(n_i \frac{\lambda_{il} - \lambda_{ij}}{\lambda_{il}\lambda_{ij}} \boldsymbol{S}_i\Big)\boldsymbol{b}_j = 0 \qquad l = 1, \ldots, p,\ l \neq j. \tag{7.15}$$

Equation (7.15) represents the system of equations that needs to be solved in order to estimate $\boldsymbol{B}$ subject to the orthogonality constraint as well the condition in (7.8). Flury and Gautschi (1984) developed an efficient algorithm to solve this problem called the FG algorithm. This is an iterative algorithm that is guaranteed to converge to a solution for (7.15) which minimises (7.8). This provides the means to find $\hat{\boldsymbol{B}}$ as well as $\hat{\boldsymbol{\Lambda}}_i$ for $i = 1, \ldots, k$. The mechanism of the FG algorithm will be detailed.

7.2.2 The FG algorithm

Since the algorithm is iterative in nature the most pertinent question is which quantity is being used in order to determine convergence. Consider the positive definite symmetric matrix $\mathbf{\Lambda}$ and define a measure of deviation from diagonality as

$$\phi(\mathbf{\Lambda}) := \frac{det(diag(\mathbf{\Lambda}))}{det(\mathbf{\Lambda})}. \tag{7.16}$$

Flury (1988) proves that $\phi(\mathbf{\Lambda}) \geq 1$ with equality resulting from $\mathbf{\Lambda}$ being exactly diagonal. Furthermore, the function $\phi(\mathbf{\Lambda})$ is monotonically increasing as $\mathbf{\Lambda}$ shifts from a diagonal matrix to a full positive definite symmetric matrix $\mathbf{\Lambda}$. $\phi(\mathbf{\Lambda})$ is a function in one symmetric positive definite matrix $\mathbf{\Lambda}$ but the FG algorithm seeks to simultaneously diagonalise k matrices, each of which ought to be considered in the convergence criterion. Flury (1988) thus defined

$$\Phi(\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_k; n_1, \ldots, n_k) := \prod_{i=1}^{k} [\phi(\mathbf{\Lambda}_i)]^{n_i}, \tag{7.17}$$

which is a measure of simultaneous deviation from diagonality. Representing each of $\mathbf{\Lambda}_i$ as $\boldsymbol{B}^{'}\boldsymbol{\Psi}_i\boldsymbol{B}$ and substituting this into (7.17) yields a function in $\boldsymbol{\Psi}_i$. Flury (1988) defines this as a measure of *simultaneous diagonalisability* of the matrices $\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_k$

$$\Phi_0(\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_k; n_1, \ldots, n_k) = \underset{\boldsymbol{B} \in \mathcal{O}(p)}{min} \Phi(\boldsymbol{B}^{'}\boldsymbol{\Psi}_1\boldsymbol{B}, \ldots, \boldsymbol{B}^{'}\boldsymbol{\Psi}_k\boldsymbol{B}; n_1, \ldots, n_k), \tag{7.18}$$

where $\mathcal{O}(p)$ is the set of orthogonal $p \times p$ matrices. Due to the fact that $\phi(\mathbf{\Lambda}_i) \geq 1$ with equality when $\mathbf{\Lambda}$ is exactly diagonal, it follows immediately that $\Phi_0 \geq 1$ with equality resulting if all the matrices $\boldsymbol{\Psi}_i$ are simultaneously diagonalisable with the same orthogonal matrix $\boldsymbol{B}$. It may seem odd that the convergence criterion for the algorithm is different from the likelihood estimation procedure that has dominated the discussion however, when considering (7.9) it is clear that the matrix $\boldsymbol{B}$ that maximises the likelihood function also minimises (7.17). The FG algorithm thus solves the system of equations in (7.15) and minimises (7.17) in the process. A solution $\hat{\boldsymbol{B}}$ which minimises (7.17) always exists due to the fact that the set of $p \times p$ orthogonal matrices $\mathcal{O}(p)$ is compact implying that the sequence of matrices $\{\boldsymbol{B}^j\}$ will converge to a matrix in the set $\mathcal{O}(p)$. Given that the equations which the FG algorithm seeks to solve has been described, it is now possible to discuss the mechanism by which the algorithm works. In order to appreciate the foundation of the FG algorithm the Jacobi algorithm must be explained.

Flury (1988) states that the oldest known method for diagonalising symmetric matrices can be attributed to Jacobi (1846). This algorithm is fundamental to the *FG* algorithm since the latter can be considered a generalisation of the Jacobi algorithm to several groups (Flury, 1988). The idea underpinning the Jacobi algorithm is simply to pre- and post- mulitply a symmetric matrix with orthogonal matrices so as to annihilate off-diagonal elements. Consider the $p \times p$ symmetric matrix $\boldsymbol{\Psi}$ to be diagonalised. Flury (1988) defines a Jacobi rotation as a $p \times p$ matrix

$$\boldsymbol{J} = \boldsymbol{J}(m,j,\theta) = \begin{array}{cc} & \begin{array}{cccccc} & & m & & j & \end{array} \\ \begin{array}{c} \\ \\ m \\ \\ j \\ \\ \\ \end{array} & \begin{pmatrix} 1 & \dots & 0 & & & \\ & & \vdots & & \vdots & \\ 0 & \dots & c & \dots & -s & \dots \\ & & \vdots & & \vdots & \\ & \dots & s & \dots & c & \dots \\ & & \vdots & & \vdots & \\ & & & & & 1 \end{pmatrix} \end{array}, \tag{7.19}$$

where $c = cos(\theta)$ and $s = sin(\theta)$. The matrix $\boldsymbol{J}$ is effectively an identity matrix in which the entries with indices m and j are replaced as shown. It is also orthogonal. To see how this rotation can result in zero off-diagonal elements consider the example taken from Flury (1988). Consider the transformation

$$\boldsymbol{H} = \boldsymbol{J}^{'}\boldsymbol{A}\boldsymbol{J}, \tag{7.20}$$

for a given value of m and j such that $1 \leq m < j \leq p$. The matrices $\boldsymbol{H}$ and $\boldsymbol{A}$ are the same but for the differences in the elements with indices m and j. These entries are determined by

$$\begin{aligned} h_{mm} &= c^2 a_{mm} + s^2 a_{jj} + 2csa_{mj}, \\ h_{jj} &= c^2 a_{mm} + s^2 a_{jj} - 2csa_{mj}, \\ h_{mj} &= h_{jm} = (c^2 - s^2)a_{mj} + cs(a_{jj} - a_{mm}). \end{aligned} \tag{7.21}$$

The objective is to get $h_{mj} = h_{jm} = 0$. Dividing the expression by a_{mj} and c^2 then setting $t = \frac{s}{c} = tan(\theta)$ allows h_{mj} to be expressed as

$$t^2 + \frac{a_{mm} - a_{jj}}{a_{mj}} t - 1 = 0. \tag{7.22}$$

If it happens that $a_{mj} = 0$ then Flury (1988) states that c should be set to one and s should be set to zero. It is clear that (7.22) has two real roots since the discriminant is greater than zero. Denote these roots as $t_1 = tan(\theta_1)$ and $t_2 = tan(\theta_2)$. Notice that expressing these in terms of the quadratic formula and taking their product yields $t_1 t_2 = -1$. This implies that the corresponding angles θ_1 and θ_2 differ by $\frac{\pi}{2}$. It is important to choose the solution that corresponds with $|\theta| \leq \frac{\pi}{4}$ due to the nature of the *FG* algorithm which will be explained shortly. Flury (1988) frames this problem in a different light when he states that determining the matrix $\boldsymbol{J}(m, j, \theta)$ to make the $(m, j)^{th}$ element of the matrix $\boldsymbol{A}$ zero is equivalent to finding the eigenvectors of the 2×2 matrix

$$\begin{pmatrix} a_{mm} & a_{mj} \\ a_{jm} & a_{jj} \end{pmatrix}. \tag{7.23}$$

This perspective is important in constructing the *FG* algorithm. This process described here is performed iteratively in the Jacobi algorithm and the choice of m and j is based on finding the largest off-diagonal element in absolute value in the matrix $\boldsymbol{A}$. In order to avoid finding the largest off-diagonal element in the matrix $\boldsymbol{A}$, Flury (1988) suggests sweeping through all possible pairs in a cyclic fashion; this means considering pairs in the order $(1, 2), (1, 3), \ldots, (1, p), (2, 3), \ldots, (p-1, p)$ for example. Rotating through each of these $\binom{p}{2}$ pairs is referred to as a *sweep* by Flury (1988). It is precisely because of this cyclic rotation that it is important to choose the angle of rotation $|\theta| \leq \frac{\pi}{4}$. This prevents large off-diagonal elements from constantly being moved ahead of the $(m, j)^{th}$ element under consideration so that repeated sweeps lead to convergence. The *FG* algorithm is a generalisation of this cyclic approach to the Jacobi algorithm. This background affords the means to describe the workings of the *FG* algorithm.

The *FG* algorithm effectively comprises two separate algorithms *viz.* the *F* algorithm and the *G* algorithm, the latter nested within the former. The *F* algorithm, on the outer level, comprises a cyclic rotation through all $\binom{p}{2}$ columns of the matrix $\hat{\boldsymbol{B}}$. Every pair of columns of the current approximation $\hat{\boldsymbol{B}}$ is rotated so as to satisfy the corresponding equation in (7.15). The *G* algorithm, on the inner level, seeks to find an orthogonal 2×2 matrix that solves a two dimensional analogue of (7.15) by means of an iterative process. The resulting solution determines the rotation of the pair of columns of $\hat{\boldsymbol{B}}$ that is currently under consideration in the *F* algorithm. In effect, the *F* algorithm sweeps through all pairs of columns of $\hat{\boldsymbol{B}}$, making use of the *G* algorithm to determine the appropriate rotation for each of the pairs. The *G* algorithm represents the different perspective mentioned when discussing the Jacobi algorithm. This broadly describes the mechanism by which the

FG algorithm works. The implementation of the algorithm, coded in *R*, is based on code received from Le Roux (2012).

7.2.3 Foundations of DCPC

Before endeavouring to marry CPC and biplot methodology, the estimation procedure required for DCPC is briefly discussed. Recall that the fundamental difference between CPC and DCPC is the relaxation on the assumption that the k groups be independent in the latter case. To understand what this means mathematically, consider the matrix $\boldsymbol{\Psi}$, the matrix of covariance matrices represented as

$$\boldsymbol{\Psi} = [\boldsymbol{\Psi}_{ij}] = \begin{pmatrix} \boldsymbol{\Psi}_{11} & \dots & \boldsymbol{\Psi}_{1k} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Psi}_{k1} & \dots & \boldsymbol{\Psi}_{kk} \end{pmatrix}. \tag{7.24}$$

In the context of CPC, $\boldsymbol{\Psi}_{ij} = \mathbf{0}$ for $i \neq j$ which is as a result of the assumption that the groups are independent. DCPC allows for the off-diagonal matrices to be non-zero which would imply dependence between groups. Extending CPC to allow for dependence simply means that the orthogonal matrix $\boldsymbol{B}$ that originally diagonalised only the covariance matrices on the diagonal of $\boldsymbol{\Psi}$ must now ensure that all the matrices comprising $\boldsymbol{\Psi}$ are diagonalised. The DCPC hypothesis as stated by Neuenschwander and Flury (2000) is as follows: The matrix $\boldsymbol{\Psi}$ satisfies the common principal components model for dependent random vectors if there exists an orthogonal matrix $\boldsymbol{B}$ such that

$$\boldsymbol{B}'\boldsymbol{\Psi}_{ij}\boldsymbol{B} = \boldsymbol{\Lambda}_{ij} = diag(\lambda_{ij,1}, \dots, \lambda_{ij,p}). \tag{7.25}$$

DCPC is also based on Maximum Likelihood Estimation and as such it is assumed that the data come from a multivariate normal distribution. The process for developing the equations to estimate the matrix $\boldsymbol{B}$ is very similar to that described for CPC albeit slightly more complex. The development will not be discussed here but rather the final system of equations to be solved in order to estimate the parameters $\boldsymbol{B}$ and $\boldsymbol{\Lambda}_{ij}$ is provided. The interested reader is referred to Neuenschwander and Flury (2000) for the details. The Maximum Likelihood Estimators $\hat{\boldsymbol{B}}$ and $\hat{\boldsymbol{\Lambda}} := [\hat{\boldsymbol{\Lambda}}_{ij}]$ is given by the solution to the system of equations

$$\boldsymbol{b}_m'\Big[\sum_{i=1}^{k}\sum_{j=1}^{k}(\lambda_{ij,l} - \lambda_{ij,m})(\boldsymbol{S}_{ij} + \boldsymbol{S}_{ji})\Big]\boldsymbol{b}_l = 0 \tag{7.26}$$

$$\lambda_{ij,h} = \boldsymbol{b}_h'\boldsymbol{S}_{ij}\boldsymbol{b}_h, \tag{7.27}$$

for $l, m, h = 1, \ldots, p$, $(l \neq m)$, $i, j = 1, \ldots, k$, $\boldsymbol{B}$ is orthogonal and $\lambda_{ij,h}$ are the elements of $\boldsymbol{\Lambda}_{ij}$ in $\boldsymbol{\Lambda} = [\boldsymbol{\Lambda}_{ij}]_{i,j=1,\ldots,k}$. Neuenschwander and Flury developed an algorithm to solve this system rather aptly named the FG^{+} algorithm given that it is an extension of the FG algorithm. The algorithm works in a similar fashion to the FG algorithm described in the previous section with some adjustments to the G algorithm. The interested reader can find the technical detail in Neuenschwander and Flury (2000).

7.3 Constructing the biplot

The question to be addressed is how the matrix $\hat{\boldsymbol{B}}$, whether estimated in the context of CPC or DCPC, can be used in order to construct a biplot. Recall that in the construction of a PCA biplot, the first two columns of the matrix $\boldsymbol{V}$ resulting from the SVD of the centred data matrix was used to determine the scaffolding axes. It may seem curious that in this context the data matrix is considered where as CPC and DCPC is concerned with the covariance matrices. Using the fact that the covariance matrix for group k is given by $\boldsymbol{\Psi}_k = \boldsymbol{X}_k^{'}\boldsymbol{X}_k$ and that the SVD of the centered data matrix $\boldsymbol{X}_k = \boldsymbol{U}_k\boldsymbol{\Sigma}_k\boldsymbol{V}_k^{'}$, it is evident that the matrix which serves to diagonalise $\boldsymbol{\Psi}_k$ and the matrix $\boldsymbol{V}$ arising from the SVD of the centered data matrix is one in the same. The implication is that the columns of $\hat{\boldsymbol{B}}$ can be used to determine the scaffolding axes for the construction of the biplot. Ordinarily, the first two columns of $\boldsymbol{V}$ are used since they are associated with the directions of greatest variation. The choice is not as simple for the CPC biplot since the diagonal matrices $\boldsymbol{\Lambda}_i$ are not ordered as in the case of SVD. This means that it is not simply a matter of using the first two columns of the matrix $\hat{\boldsymbol{B}}$. The choice is further complicated by the fact that there are k separate diagonal matrices $\hat{\boldsymbol{\Lambda}}_k$ to consider. Determining which columns of $\hat{\boldsymbol{B}}$ to use requires a new definition for a measure of quality based on the variation in the data captured in the biplot display. In order to achieve this a new measure of quality needs to be defined based on that which is used for ordinary PCA biplots. This new measure of quality is defined as

$$quality(\boldsymbol{b}_i, \boldsymbol{b}_j) := \frac{\sum_{h=1}^{k} \frac{\lambda_h(i,i)+\lambda_h(j,j)}{\sum_{n=1}^{p}\lambda_h(n,n)}}{k}, \tag{7.28}$$

in the case of CPC where $\lambda_h(i,i)$ is the i^{th} element on the diagonal of $\boldsymbol{\Lambda}_h$. Division by k ensures that the average quality measure falls between 0 and 1 since the defined measure is based on the sum of k qualities. The quality measure is defined somewhat differently in the context of DCPC as

$$quality(\boldsymbol{b}_i, \boldsymbol{b}_j) := \frac{\sum_{h=1}^{k} \sum_{m=1}^{k} \frac{\lambda_{hm}(i,i)+\lambda_{hm}(j,j)}{\sum_{n=1}^{p} \lambda_{hm}(n,n)}}{k}, \tag{7.29}$$

where $\lambda_{hm}(i,i)$ is the i^{th} element on the diagonal of $\boldsymbol{\Lambda}_{hm}$. The two vectors $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$ that give the highest quality measure are used to construct the biplot. Once this has been established, the construction process is identical to that used in producing a PCA biplot. Assume that $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ represent the directions of the best fitting plane. The rows of the $p \times 2$ matrix $\hat{\boldsymbol{B}}_2 = [\boldsymbol{b}_1, \boldsymbol{b}_2]$ will be used to represent the variable axes. For each of the data matrices $\boldsymbol{X}_i$, the interpolated sample points are represented as $\boldsymbol{Z}_i = \boldsymbol{X}_i \hat{\boldsymbol{B}}_2$ for $i = 1, \ldots, k$. The rows of $\boldsymbol{Z}_i$ are plotted to represent the samples.

There is one more aspect of the construction process that should be considered and that is how the variables axes are to be calibrated. In Chapter 4, the calibration process was detailed and it was shown that the variable means are used in determining the values that appear on the markers. In the context of CPC, k different data matrices are being considered and that implies that there are k sets of p variable means to consider. This presents a problem, however there are two possible solutions. The first option is simply to have k sets of markers on each of the variable axes relating to each of the k groups. Although this is a viable option it can make reading values off of the biplot very cumbersome and more trivially it affects the aesthetics of the display. The second option relies on centering each of the k datasets, binding it into a single dataset and using the resulting columns to calibrate the axis. In this instance the values read off will represent deviations from a mean as opposed to the approximated data value. This means that if a value is read off of variable axis one for a sample point in the first dataset, that value represents the deviation of the observation from the mean for variable one in group one. To find the approximated value, the mean for variable one in group one must be added back to the value that was read off of the axis. Although this method is cumbersome in its own right it makes for a display that is easier to read. For illustration purposes consider $\hat{x}_{111}$ and call the value read off of the axis x. The mean vector $\overline{\boldsymbol{X}}_1 = (\overline{X}_{11} \ldots \overline{X}_{p1})$ represents the mean for each variable at occasion one. Since this observation is for subject one on variable one at occasion, the approximation $\hat{x}_{111} = x + \overline{X}_{11}$. Both of these will be illustrated graphically.

Although the construction of the biplot has been detailed it is important to give thought to how to interpret the resulting plot. More specifically, careful consideration must be given to how to interpret the angles between the variable axes. Since the constructed biplot is akin to that which optimally represents the Euclidean distances between sample points, the angles between the variable axes only provide a sense of the strength of association. The question is really what association is in fact being measured and in order to answer this it is necessary to appeal to the intuitive foundation of CPC and what the technique seeks to do. Essentially the technique posits that the principal component transformation is identical across the k groups under consideration and in order for that to be the case there needs to be similarity in the covariance structure of the k groups. In fact there must be a latent or underlying variation that is common across groups. This can be thought of as systemic variation. It is argued that the association read off of the biplot provides a sense of the latent relationship between variables. For example, a small angle between variable axes one and two would suggest that $V1$ and $V2$ have a latent relationship. Since the covariance structure is not assumed to be identical across groups, the relationship between variables does not remain constant. This is due to the effect of the unique variation inherent in each dataset comprising the complete data. Furthermore it is vital to be cognisant of the fact that the constructed biplot does not truly present the Euclidean distances between sample points optimally. In effect the CPC biplot is related to its PCA cousin in that both rely on projecting both sample points and variable axes onto a plane in $\Re^p$. In the former instance, the sample point co-ordinates that are plotted are not in principal co-ordinates as would be the case for an ordinary PCA biplot. It is for this reason that Euclidean distances are not necessarily preserved but the same interpretational tools used for the PCA biplot are used for the CPC biplot.

7.4 Application to Mansoor data

This section details the construction of both the CPC and DCPC biplots and seeks to ascertain whether the assertions made regarding the interpretation are evidenced by the data. The constructed biplots will be compared to the PCA biplots produced for each of the four occasions comprising the data. In this way it is possible to see how well the sample points are represented and surmise on what impacts the quality of the representation. Furthermore the strength of association between variables in the CPC and DCPC biplots will be compared with those in the single PCA biplots to see whether it is markedly different or not. Note that the data were seen not to follow a multivariate normal distribution. Flury (1988) suggests that even in the face of the distributional assumption being violated, it is still instructive to employ the CPC technique. Distributional considerations are beyond the scope of this dissertation. It is also assumed that the CPC hypothesis applies to the Mansoor data since the aim is to consider the biplot and interpret it.

After applying the FG algorithm to the estimated variance matrices $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_4$ and employing the quality measure defined in 7.28, it was found that the first two columns of the matrix $\hat{\boldsymbol{B}}$ was to be used in the biplot construction since these columns yielded a quality score of 81.8%. The estimated matrix $\hat{\boldsymbol{B}}_2$ is given by

$$\hat{\boldsymbol{B}}_2 = \begin{pmatrix} -0.167 & 0.352 \\ -0.922 & -0.357 \\ -0.152 & 0.144 \\ -0.302 & 0.717 \\ -0.044 & 0.374 \\ -0.071 & 0.101 \\ -0.0347 & 0.253 \end{pmatrix}. \tag{7.30}$$

The directions of the variable axes are based on plotting the rows of $\hat{\boldsymbol{B}}_2$. The sample points are represented by the rows of the matrices $\boldsymbol{X}_1\hat{\boldsymbol{B}}_2, \ldots, \boldsymbol{X}_4\hat{\boldsymbol{B}}_2$. Comment should be passed on the nature of the estimated diagonal matrices $\hat{\boldsymbol{\Lambda}}_1, \ldots, \hat{\boldsymbol{\Lambda}}_4$. Although these estimated matrices do not closely resemble diagonal matrices in the sense that off diagonal elements are large, Beaghen (1997) argues that if off-diagonal elements are small relative to the diagonal elements per column then the CPC hypothesis is appropriate. This was seen to be the case for the columns comprising $\hat{\boldsymbol{B}}_2$. Attention now falls to interpreting the resulting plot.

Figure 7.1 represents the collection of PCA biplots for each occasion, constructed so as to approximate Euclidean distances between sample points optimally. Figures 7.2 and 7.3 both represent the CPC biplot for the Mansoor data differing only in the calibration of the axes. Figure 7.2 makes use of multiple markers and Figure 7.3 calibrates the axes with deviations from the relevant mean and for the purposes of discussion Figure 7.2 will be referred to. The first aspect to compare is the display of the sample points. Figure 7.2 adequately captures the gradual reduction in variation from occasion 1 to occasion 4. Furthermore it is clear that the CPC biplot seems to display the sample points for occasion one in a similar fashion to that seen in the separate PCA biplot for that occasion. The display of the sample points for the remaining occasions, although sharing similarities with display in the separate biplots, is less similar in comparison. This is to be expected since

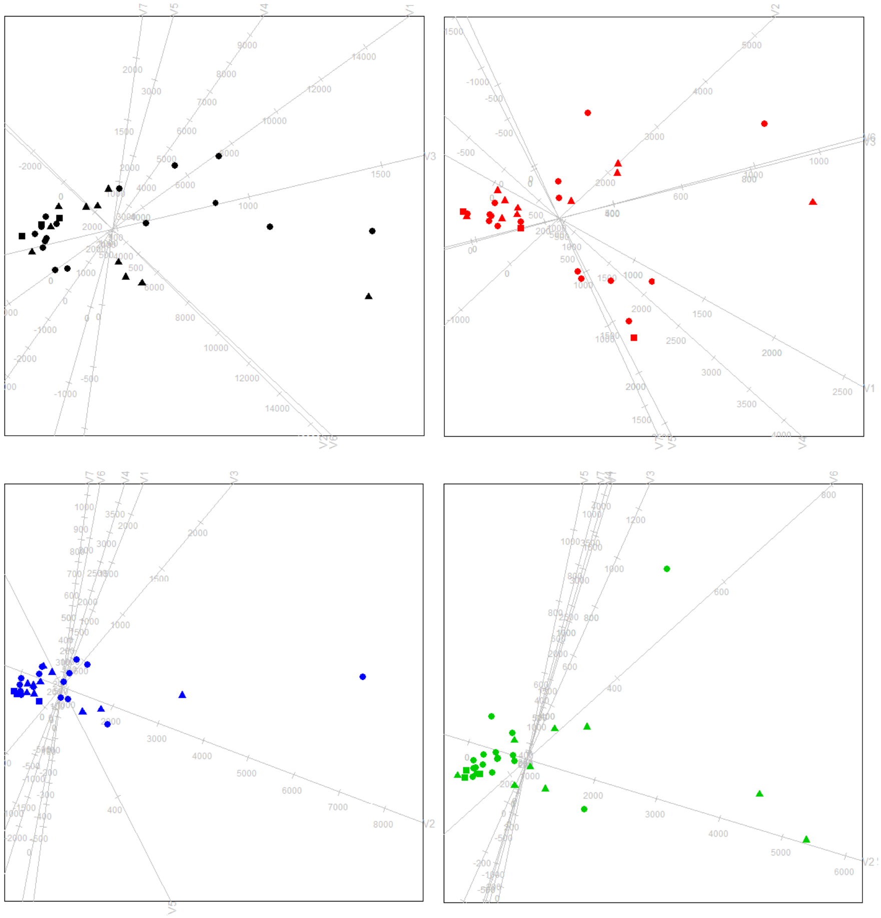


Figure 7.1: Separate PCA biplots for each occasion ordered in a clockwise fashion with the biplot corresponding to occasion 1 in the top-left.

the scaffolding produced by the CPC estimation represents a compromise between the best-fitting planes if each occasion were to be analysed separately. This does not render the display useless in conveying information about the sample points because although Euclidean distances between sample points is on well represented, the orientation of the points is very similar and it is possible to read off approximate values for these data points. In fact the displays look similar but for the fact that the Euclidean distances between the sample points in the CPC biplot are distorted when compared to the display in the separate PCA biplots. It is natural to wonder why occasion one seems relatively well represented when compared with the other occasions. The answer is not concrete and is rooted in conjecture however considering the estimating equations might shed some light on the matter.

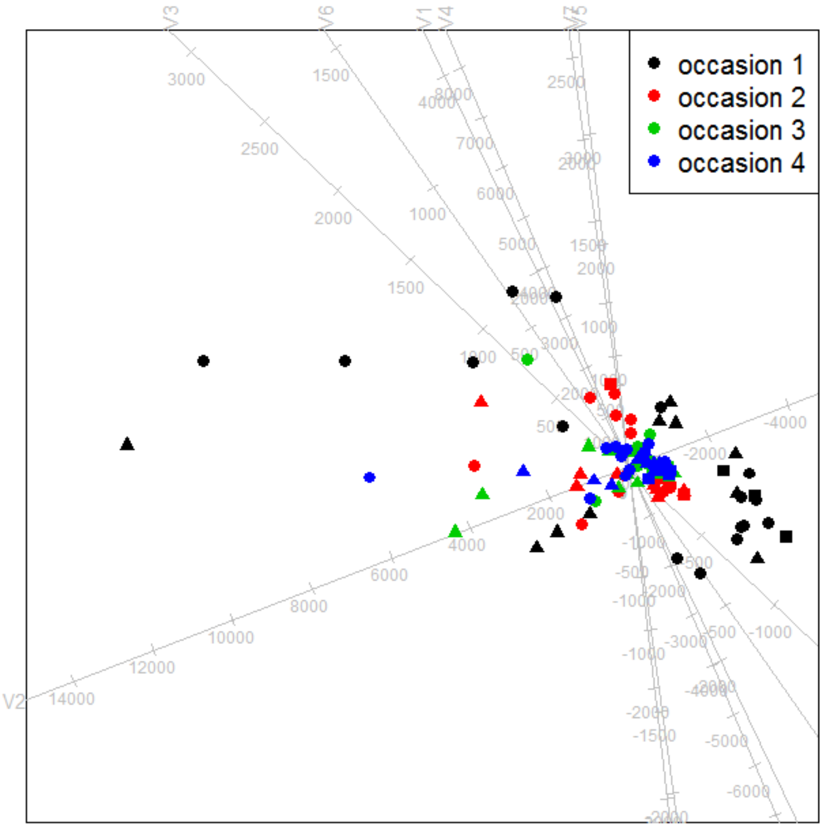


Figure 7.2: CPC biplot for the Mansoor data calibrated with deviations from relevant mean.

Equation (7.15) shows that the estimation procedure rests on a weighted sum of the variance matrices $\boldsymbol{S}_1 \ldots, \boldsymbol{S}_4$. It is not only the weighting factor but the actual variance matrices that have an impact. For this specific dataset it was seen that weighting factors did not differ vastly throughout the various iterations of the FG algorithm but the elements of the variance matrix $\boldsymbol{S}_1$ are relatively big when compared with the remaining variance matrices. It could thus be that $\boldsymbol{S}_1$ is influential in the estimation of the final matrix $\hat{\boldsymbol{B}}$ and as such the sample points at occasion 1 are better displayed.

The next aspect to consider is the interpretation of the angles between the variable axes as a measure of systemic association. Take $V5$ and $V7$ as an example and notice that in Figure 7.1 these variables seem strongly associated at occasions 1, 2 and 3 with a slightly less pronounced association conveyed at occasion 4. Figure 7.2 shows a strong association between these variables as evidenced by the magnitude of the angle between the variable axes. Considering $V2$ it is clear that with the exception of occasion 1, this variable shows no particular association with any of the variables. This is reflected in the CPC biplot in Figure 7.2. This lends credence to the notion that the CPC biplot illustrates the systemic relationship between variables.

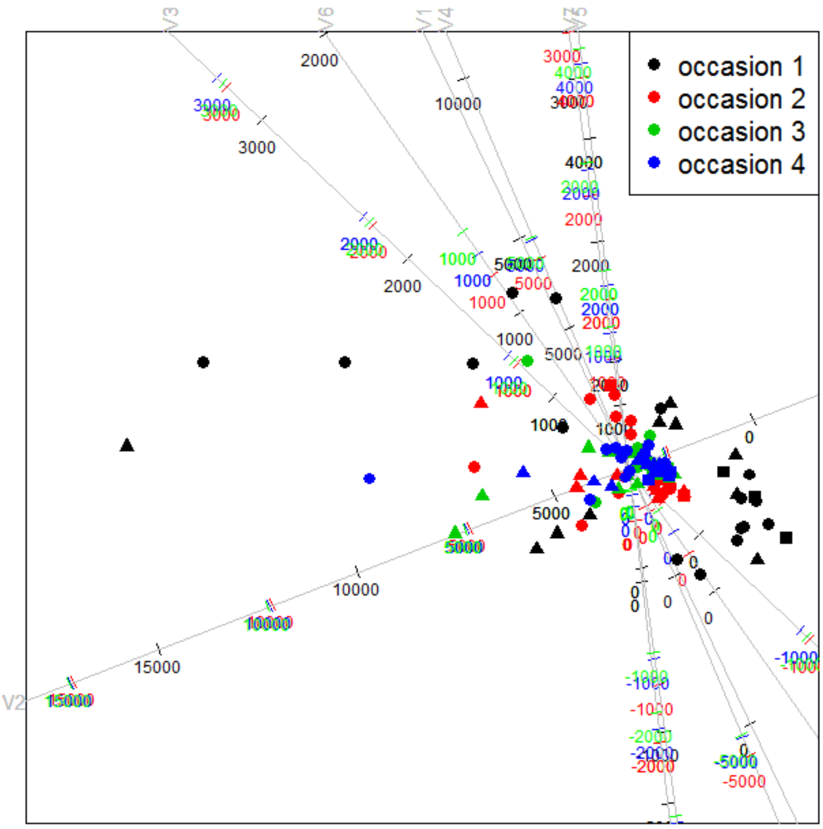


Figure 7.3: CPC biplot for the Mansoor data with multiple markers on the variable axes.

The last consideration is what information the biplot provides regarding the research questions posed by Mansoor *et al.* (2009). Recall that the authors were concerned with ascertaining whether the immune response after the administration of the drug *BCG* accorded with that which is though to be the

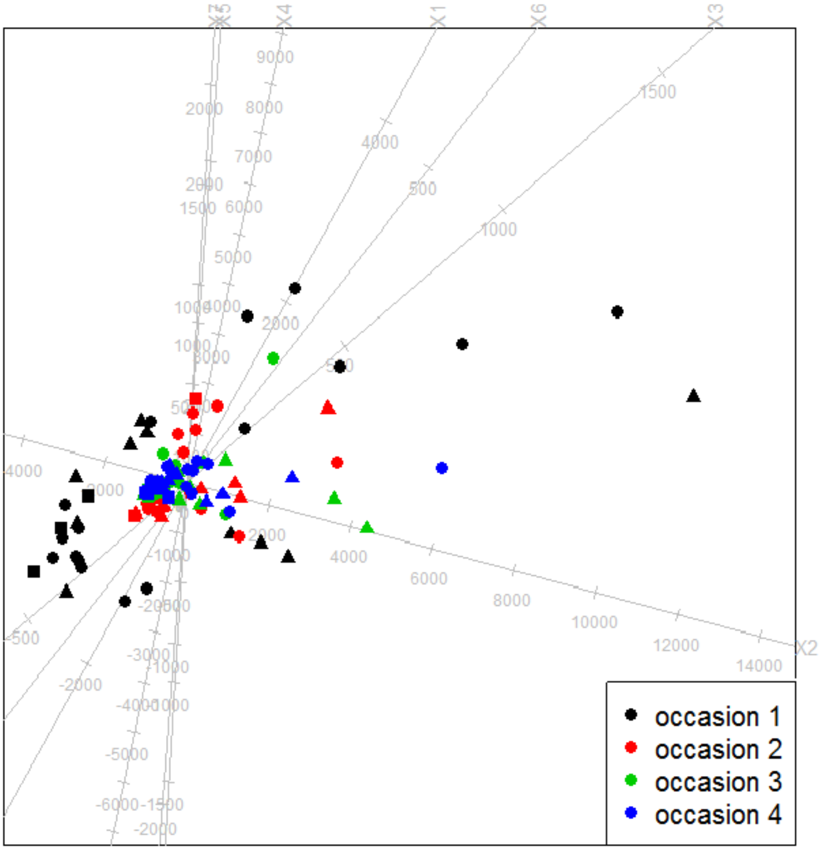


Figure 7.4: DCPC biplot for the Mansoor data calibrated with deviations from relevant mean.

required immune response in HIV^+ patients . This required that patients score highly on $V4$. Figure 7.2 shows that the patients comprising the HIV^+ group scored poorly on $V4$. Furthermore, although the variation reduced over time, it is evident that infants comprising groups 2 and 3 mustered appreciably better scores on all variables across time. The infants in group 3 generally seemed to achieve the best scores. The application of the DCPC technique was based on the fact that repeated measures data likely violates the independence assumption that is made in the context of CPC. Using the quality measure defined in (7.29) it was determined that columns two and four of the estimated matrix be used to construct the scaffolding since this yielded a quality score of 88.39%. Figure 7.4 is an illustration of the resulting DCPC biplot and it is interesting to note that the information conveyed by the plot is very similar to that seen in the CPC biplot. Both the orientation of the variable axes as well as the orientation of the sample points is very similar to that seen in the CPC biplot.

7.5 Conclusion

This chapter detailed the theory underpinning the techniques of CPC and DCPC with the intention of constructing biplots. The focus was not on testing whether the CPC hypothesis was feasible but rather assuming that a researcher is satisfied that this is indeed the case, using the estimates produced by this technique for graphical displays. New quality measures were defined in order to determine the scaffolding for the biplot construction. It was seen that the angles between variable axes represent systemic relationships between the variables as evidenced by the comparison of the CPC and DCPC biplots to the separate PCA biplots produced for each occasion. Furthermore, it was seen that whilst the Euclidean distances between sample points are not optimally represented, occasions with large variation relative to the rest may be better represented. The plot was also useful in showcasing changes in variation in the data over time and provided an approximation of how sample points relate to one another. Calibration presented a problem and the solution was either to show multiple markers on the variable axes relating to each occasion or to calibrate with deviations from the relevant mean. In effect, although the CPC and DCPC biplots are based on approximating a common scaffolding as opposed to using the optimal scaffolding at each occasion, it provided valuable information about systemic variable associations, changes in variation in the data as well visualising how sample points evolve over time or differ across condition.

Chapter 8

A Common CVA Biplot

8.1 Introduction

Chapter 6 detailed the use of GOPA in order to superimpose k PCA biplots and thus represent all biplots in a single plot. It might be tempting to use precisely the same technique with CVA biplots however this would not be correct. The problem with this method is that CVA requires that a non-singular linear transformation be employed to display the data in the canonical space. By virtue of the fact that the transformation is non-singular, the basis vectors for the canonical space are oblique and thus possibly completely different for each matrix $\boldsymbol{X}_k$. Ultimately this implies that each $\boldsymbol{X}_k$ is transformed to its unique canonical space in which the CVA biplots are constructed and as such it would not be valid to simply superimpose them.

This chapter is geared at alleviating this problem by introducing a means to effectively represent a Common CVA biplot so that each occasion can be represented on a single biplot. In order to achieve this, Common Principal Component Analysis (CPC) is used in order to find a solution to the problem of CVA as discussed in Chapter 5. Since the theoretical foundations of these techniques have been discussed previously, this chapter simply details the construction of the biplot and applies it to simulated data as well as to the Mansoor data.

8.1.1 Constructing a Common CVA Biplot

This section details how the two techniques can be combined to construct a common CVA biplot. It builds on the two-step approach to CVA in a simple manner. Recall that the first step is related to finding a matrix $\boldsymbol{L}$ such that $\boldsymbol{L}^{'}\boldsymbol{W}\boldsymbol{L} = \boldsymbol{I}_p$. Now there are k datasets $\boldsymbol{X}_i$ and consequently k

between and within group variation matrices $\boldsymbol{B}_i$ and $\boldsymbol{W}_i$ respectively. The first step in the process of constructing the Common CVA biplot is to apply the FG algorithm to find a matrix $\boldsymbol{L}^*$ that simultaneously diagonalises the matrices $\boldsymbol{W}_i$ so that $\boldsymbol{L}^{*'}\boldsymbol{W}_i\boldsymbol{L}^* = \boldsymbol{\Phi}_i$. Define $\boldsymbol{L}_i = \boldsymbol{L}^*\boldsymbol{\Phi}_i^{-0.5}$ to ensure that $\boldsymbol{L}_i^{'}\boldsymbol{W}_i\boldsymbol{L}_i = \boldsymbol{I}_p$ as required. The next step in the process is then to use the FG algorithm to find a matrix $\boldsymbol{V}^*$ that will simultaneously diagonalise the matrices $\boldsymbol{L}_i^{'}\boldsymbol{B}_i\boldsymbol{L}$ resulting in k near diagonal matrices $\boldsymbol{\Lambda}_i$. The second step can be represented mathematically as

$$\boldsymbol{\Phi}_i^{-0.5}\boldsymbol{L}^{*'}\boldsymbol{B}_i\boldsymbol{L}^*\boldsymbol{\Phi}_i^{-0.5} = \boldsymbol{V}^*\boldsymbol{\Lambda}_i\boldsymbol{V}^{*'}. \tag{8.1}$$

This implies that

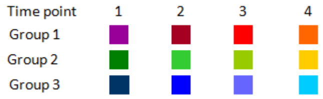
$$\boldsymbol{L}^{*'}\boldsymbol{B}_i\boldsymbol{L}^* = (\boldsymbol{\Phi}_i^{0.5}\boldsymbol{V}^*)\boldsymbol{\Lambda}_i(\boldsymbol{\Phi}_i^{0.5}\boldsymbol{V}^*)'. \tag{8.2}$$

Define a matrix $\boldsymbol{V}_i = \boldsymbol{\Phi}_i^{0.5}\boldsymbol{V}^*$. A slight alteration is made to the FG algorithm in the convergence criterion to accommodate the fact that the matrix $\boldsymbol{B}_i$ has rank, s equal to the minimum of $g-1$ and p. The alteration results in only the first s columns of $\boldsymbol{V}^{'}\boldsymbol{L}_i^{'}\boldsymbol{B}_i\boldsymbol{L}\boldsymbol{V}$ being used in the calculation of the convergence criterion. This ensures that the convergence criterion does not contain division by the product of eigenvalues of which some are zero. CPC has thus been used in both steps comprising the two-step approach to CVA.

Finding the two dimensional approximation of the data in traditional CVA requires that the first two columns of the matrix $\boldsymbol{XLV}$ be plotted. Define the matrix $\boldsymbol{M}$ to be $\boldsymbol{LV}$. The question at hand in the process of constructing the common CVA biplot is which matrices to use for $\boldsymbol{L}$ and $\boldsymbol{V}$ since in each case there are two options; $\boldsymbol{L}^*$, $\boldsymbol{L}_i$ in the former instance and $\boldsymbol{V}^*$, $\boldsymbol{V}_i$ in the latter instance. If this methodology is to reduce to ordinary CVA when applied to a single data set then $\boldsymbol{L}_i\boldsymbol{V}^*$ is to be used. This means that each data set $\boldsymbol{X}_i$ will be associated with a different $\boldsymbol{M}_i = \boldsymbol{L}^*\boldsymbol{\Phi}_i^{-0.5}\boldsymbol{V}^*$ and this immediately raises the question of how all the data can be displayed on one plot. The reason given for why it was not possible to simply apply GOPA to CVA biplots produced for each data set $\boldsymbol{X}_i$ was that the respective bases for the canonical spaces were not orthogonal and thus each space was unique. This is no longer of concern when using each of the $\boldsymbol{M}_i$ matrices because the basis vectors have the same direction in each instance and differ only in length. This means that all the axes can be projected onto a single plane resulting in a plot with kp axes together with the sample points.

8.2 Application to simulated and Mansoor data

The data set that was simulated in Chapter 5 is used in order to assess how well the extent of the separation and within-group variation is captured in the display of the data. Recall that $\boldsymbol{\mu}_3$ remained unchanged over time where as there is a pattern in the way that the mean vectors $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ change. It is also important to note the fact that the variation at timepoints 3 and 4 is greater relative to timepoints 1 and 2. These aspects of the simulated data would be expected to be represented in the biplot constructed if it is to be considered an adequate means of representing the data.



Time point	1	2	3	4
Group 1	🟪	🟥	🟥	🟧
Group 2	🟩	🟩	🟩	🟨
Group 3	🟦	🟦	🟦	🟦

Figure 8.1: Legend for biplots.

Figure 8.2 represents the biplot constructed by using the four different $\boldsymbol{M}$ matrices. The solid blocks represent the interpolated means and the colours correspond to groups and timepoints as indicated in Figure 8.1. From the position of the interpolated means for group 1 it is evident that the biplot captures the fact that the mean increased from timepoint 1 to timepoint 2, decreased back to its original value and increased once again at timepoint 4. Consider group 2 and notice that the biplot clearly shows greater variation for timepoint 4 relative to timepoint 2. The biplot does well to separate the groups and seems to provide a poor approximation of changes in group 3, particularly on variable 1. Notice that the plot contains 12 axes as opposed to 3. This is due to the fact that this plot contains kp axes where $k = 4$ and $p = 3$ in this example. This biplot however has the advantage that the predictions are very close to the specified vectors for the interpolated means which suggests that it fares well in providing a means of predicting data values. Biplots are often useful in providing a sense of the correlation between the variables comprising the data. $\boldsymbol{\Sigma}$ indicates that the variables are uncorrelated since it is a diagonal matrix. Visually this would be represented as orthogonal axes but it is clear that though variables 1 and 2 are near orthogonal, variable 3 seems closely related to variable 2 given that they are relatively close in Figure 8.2. There are two reasons for this. Given that this is a two dimensional approximation it is impossible to represent all axes orthogonal to one another. Secondly, since there was more variation in variables one and two, those are better represented. $V3$ is possibly orthogonal to the display space and as such projection onto the biplot could be in almost any

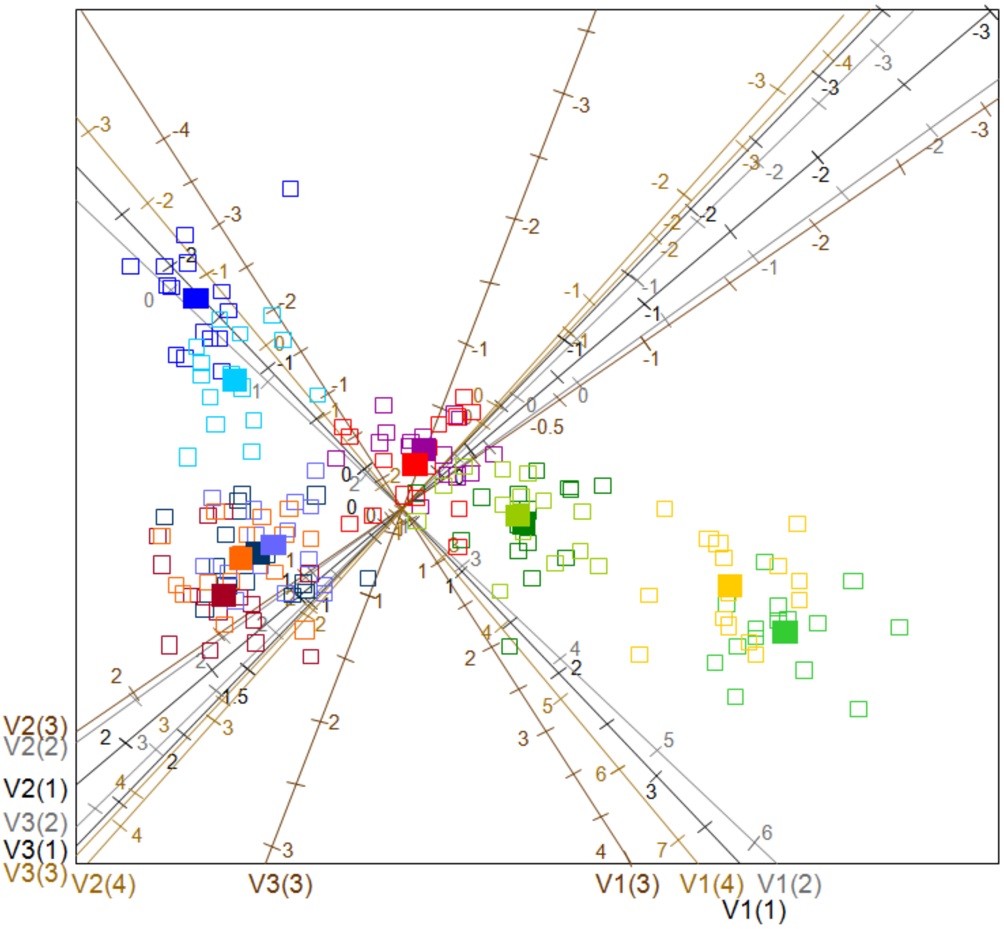


Figure 8.2: CVA biplot constructed using $\boldsymbol{M}_i$ where $i = 1, \dots, k$

direction. Figure 8.3 represents the biplot constructed for the Mansoor data with the legend in Figure 8.1 still applying for reference purposes. Bearing in mind that the CVA biplot is predominantly used to ascertain the separation between the groups in the data it is clear that there is not a great deal of separation between groups at any occasion. It also seems that the groups maintain very similar separation across time. One limitation results from the fact that four occasions are being represented simultaneously but each of the group means differ across occasion. The group means for occasions one to three respectively are as follows:

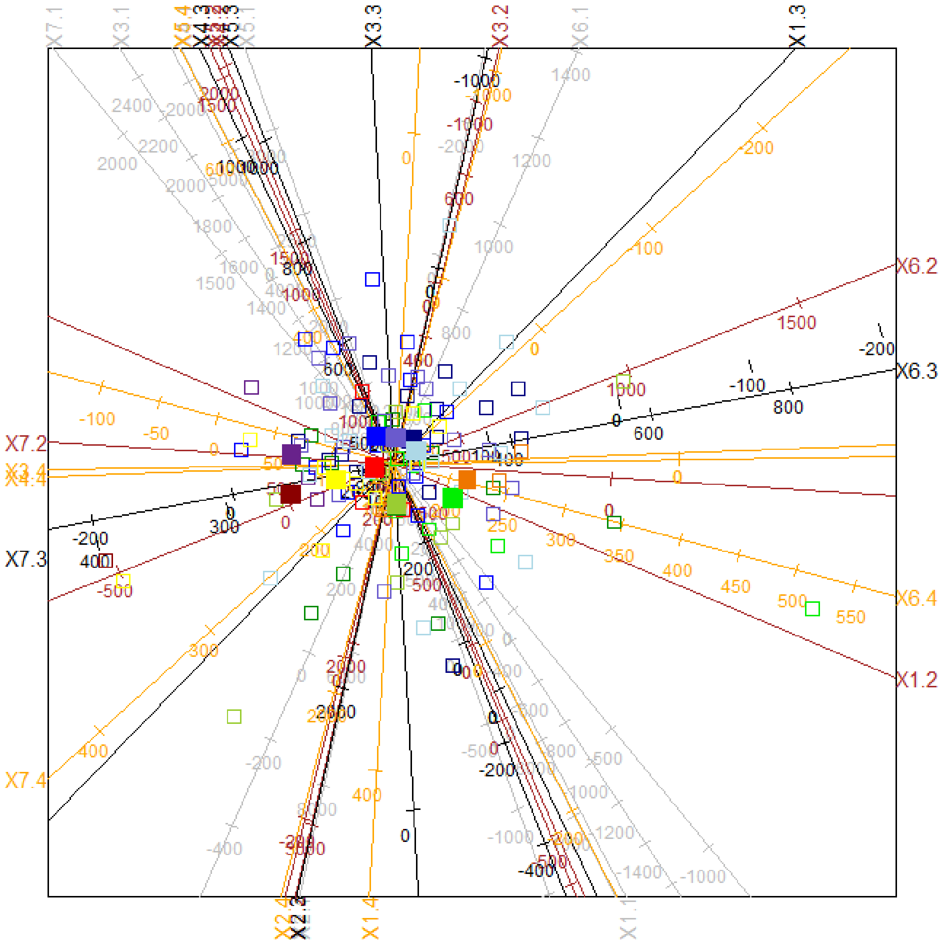


Figure 8.3: CVA biplot constructed using $\boldsymbol{M}_i$ where $i = 1, \ldots, 4$ for the Mansoor data.

$$\begin{array}{ccccccc} V1 & V2 & V3 & V4 & V5 & V6 & V7 \\ \left(\begin{array}{c} 600.67 \\ 2371.3 \\ 3190.19 \end{array}\right. & \begin{array}{c} 708 \\ 3785.7 \\ 3188.38 \end{array} & \begin{array}{c} 95.67 \\ 524 \\ 529.25 \end{array} & \begin{array}{c} 1077 \\ 2245.4 \\ 2348.56 \end{array} & \begin{array}{c} 799 \\ 853.2 \\ 1187.31 \end{array} & \begin{array}{c} 88 \\ 511.6 \\ 466.56 \end{array} & \left.\begin{array}{c} 543.33 \\ 689.5 \\ 582.63 \end{array}\right) \end{array}$$

$$\begin{array}{ccccccc} V1 & V2 & V3 & V4 & V5 & V6 & V7 \\ \left(\begin{array}{c} 395.33 \\ 498.7 \\ 451.38 \end{array}\right. & \begin{array}{c} 454 \\ 1214.5 \\ 972.8 \end{array} & \begin{array}{c} 86 \\ 231.5 \\ 277.07 \end{array} & \begin{array}{c} 854 \\ 619.4 \\ 877.38 \end{array} & \begin{array}{c} 724.67 \\ 349.4 \\ 513.5 \end{array} & \begin{array}{c} 109.67 \\ 441.3 \\ 261.88 \end{array} & \left.\begin{array}{c} 760.67 \\ 235.6 \\ 309.82 \end{array}\right) \end{array}$$

$$\begin{array}{ccccccc} V1 & V2 & V3 & V4 & V5 & V6 & V7 \\ \left(\begin{array}{c} 27 \\ 222.4 \\ 206.88 \end{array}\right. & \begin{array}{c} 181.67 \\ 1678.8 \\ 603.38 \end{array} & \begin{array}{c} 6.33 \\ 240 \\ 169.69 \end{array} & \begin{array}{c} 44 \\ 420.5 \\ 474.56 \end{array} & \begin{array}{c} 291.67 \\ 295 \\ 406.75 \end{array} & \begin{array}{c} 59.67 \\ 269.3 \\ 240.19 \end{array} & \left.\begin{array}{c} 63.67 \\ 144.8 \\ 210.88 \end{array}\right) \end{array}$$

Reading off the variables axes for the mean of group one at occasion one reveals a relatively good approximation but notice that the calibration on the axes are very different across variables. The point being made is that although reading off the axes can yield a relatively good approximation, the Euclidean distances between the samples will not be represented accurately. Nonetheless, the biplot remains useful in determining the extent to which the groups comprising the data are separated and how that separation changes over time. The plot also shows how the mean changes over time

8.3 Conclusion

In this chapter a novel method for displaying grouped longitudinal data on a single biplot was proposed. The methodology was based on marrying the ideas of Canonical Variate Analysis and Common Principal Components Analysis. The biplot constructed relies on the use of K different $\boldsymbol{M}$ matrices. This method reduces to classical CVA when applied to a single data matrix. It is this property that results in the Euclidean distance in the canonical space being equivalent to the Mahalanobis distance in the original space. Since any CVA biplot is concerned the separation of groups and the Mansoor data does not display a well separated group structure, the proposed biplot was tested using a simulated data set in order to vary certain aspects of the data and see how well the constructed biplot did to convey this information. The results indicated that the biplot produced a superior display of the data. The method was also applied to the Mansoor data and it was revealed that there was not a great deal of separation between the groups comprising the data.

Chapter 9

Three mode models, biplots and triplots

9.1 Introduction

Mathematicians at the beginning of the twentieth century, particularly those in the field of Linear Algebra, showed interest in finding ways to handle more than one matrix at a time and in understanding the properties and eigendecompositions of multiway arrays, also referred to as tensors. The earliest work in tensor decompositions and the problems related to the rank of multiway structures was done by Hitchcock (1927a, 1927b). This chapter takes on a distinctly different flavour in that consideration is given to the methods for higher order multiway array or tensor decomposition. In other words, methods that can be thought of as true three mode decomposition techniques are discussed. It is arguably the natural progression given the fact that biplot methodology depends on representing the best rank r approximation to a two mode dataset. The notion of best rank r needs to clarified in the three mode context before considering whether biplots can be produced and whether biplot methodology can in fact be extended to produce triplots, a graphical display that reflects information not only about samples and variables but also occasions or conditions. Although the ideas underpinning rank in the context of higher order tensors are very similar to those of matrix rank, there are nuances that introduce complexities. Here, the notion of tensor rank is thoroughly explored together with two classes of tensor decomposition methods, the Tucker three (Tucker3) model as well as the Parallel Factor Analysis (PARAFAC) model. Although statistical context will be given to these models with respect to their development as data analysis tools it is important to be cognisant of the inherent dual perspective in these methods. This can be likened to the different perspectives that resulted in the discovery of PCA. Where Pearson (1901) was concerned with a best fitting line or plane in p dimensional space, Hotelling (1933) was concerned with reducing the data to a few fundamental variables which he initially termed factors but later chose to call components. The former perspective is inherently geometric and sits at the heart of PCA biplot methodology. The latter perspective is somewhat more data analytic and seeks to give meaning to the components derived from a PCA. This paradigm extends to tensor decomposition techniques and this sentiment is echoed by Bro (1997) when he speaks of using the PARAFAC model in particular for parameter estimation or simply as a means to decompose multi-collinear data. The significance of this duality is that issues that are of a modelling nature are not considered here particularly dimensionality selection as well as component and core interpretation. The interested reader is referred to Kroonenberg (2008) which provides a comprehensive discussion on these issues. Instead a more exploratory approach is taken where methods are considered purely to decompose tensors and these simplified structures are used to produce biplots and attempt to extend biplot methodology to produce triplots. Given the fact that the SVD is fundamental to biplot construction, each tensor decomposition method will be considered in terms of the SVD properties that it preserves. The word tensor will refer to order three tensors or three mode arrays throughout this chapter.

9.2 Basic definitions

9.2.1 Matrix representations of order three tensors

As a result of the fact that tensor decomposition techniques are often represented in “unfolded” form and that multilinear rank is defined in this way it is necessary to consider what precisely is meant by unfolding a tensor in each of its modes. These definitions differ somewhat to those given in Chapter 2 with the exception of the first unfolding in Figure 9.1. There are various definitions available but the one given here is taken from Kiers (2000b). It is easiest to define this notion with the aid of Figure 9.1. Unfolding the tensor $\mathcal{X}$ in the first mode, represented as $\boldsymbol{X}_{(1)}$ is achieved by taking the frontal slices of the array and placing them next to one another as indicated in the first diagram in Figure 9.1 so that $\boldsymbol{X}_{(1)}$ has dimensions $N \times PK$. $\boldsymbol{X}_{(2)}$ and $\boldsymbol{X}_{(3)}$ are defined by placing the lateral and horizontal slices of the array $\mathcal{X}$ next to another with dimensions $P \times NK$ and $K \times NP$ respectively.

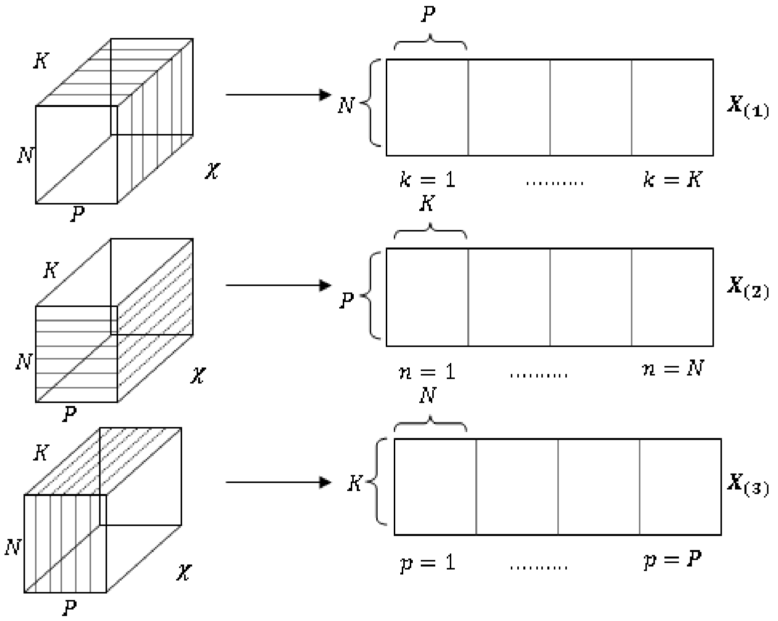


Figure 9.1: Visual representation of various matrix unfoldings of $\mathcal{X}$.

9.2.2 Scalar product, orthogonality and norm of tensors

It is necessary to define each of these concepts in the tensor context because they become important when placing tensor decomposition techniques in a SVD framework, particularly the Tucker3 decomposition. Each definition will simply be stated with the defined concepts being used later on. The first notion to define is that of the scalar product of tensors.

Definition 9.2.1. *The scalar product of tensors $\mathcal{X}$,$\mathcal{Y} \in \Re^{d_1 \times d_2 \times d_3}$, denoted by $< \mathcal{X}, \mathcal{Y} >$, is defined as*

$$< \mathcal{X}, \mathcal{Y} > := \sum_{d_1} \sum_{d_2} \sum_{d_3} x_{d_1 d_2 d_3} y_{d_1 d_2 d_3}. \tag{9.1}$$

In essence, the scalar product is calculated by multiplying corresponding tensor entries and summing these products. Tensors for which the scalar product is 0 are deemed orthogonal. Finally the notion of the Frobenius norm of a tensor is defined.

Definition 9.2.2. *The Frobenius norm of a tensor $\mathcal{X}$ is given by*

$$\|\mathcal{X}\| = \sqrt{< \mathcal{X}, \mathcal{X} >}. \tag{9.2}$$

9.2.3 Matrix Tensor multiplication

The motivation for defining the notion of tensor matrix multiplication will be given later. At this stage, it suffices to simply provide the definition. Naturally the definition will rely on the tensor unfolded in the n^{th} mode, $\boldsymbol{X}_{(n)}$, where $n = 1, 2, 3$ and the product referred to as the n-mode product.

Definition 9.2.3. *The* n-mode product *of a tensor* $\mathcal{X} \in \Re^{d_1 \times d_2 \times d_3}$ *with a matrix* $\boldsymbol{U} \in \Re^{J_n \times d_n}$ *is denoted by* $\mathcal{X}_{\times_n} \boldsymbol{U}$ *and is defined as*

$$\mathcal{X}_{\times_n} \boldsymbol{U} := \boldsymbol{U} \boldsymbol{X}_{(n)}. \tag{9.3}$$

This definition gives the n-mode product in matrix form though it is certainly possible to rearrange the resulting matrix into a tensor with the d_n^{th} dimension replaced by J_n. As an example consider the unfolding of the tensor $\mathcal{X}$ in the first mode

$$\boldsymbol{X}_{(1)} = \begin{pmatrix} x_{111} & \dots & x_{1p1} & x_{112} & \dots & x_{1pk} \\ \vdots & & \vdots & & \vdots & \\ x_{N11} & \dots & x_{Np1} & x_{N12} & \dots & x_{Npk} \end{pmatrix}, \tag{9.4}$$

and the 1-mode product of $\mathcal{X}$ with the matrix $\boldsymbol{U}_{m \times n}$ yields

$$\mathcal{X} \times_1 \boldsymbol{U} = \begin{pmatrix} \sum_{i=1}^{N} u_{1i} x_{i11} & \dots & \sum_{i=1}^{N} u_{1i} x_{ip1} & \sum_{i=1}^{N} u_{1i} x_{i12} & \dots & \sum_{i=1}^{N} u_{1i} x_{ipk} \\ \vdots & & \vdots & & \vdots & \\ \sum_{i=1}^{N} u_{mi} x_{i11} & \dots & \sum_{i=1}^{N} u_{mi} x_{ip1} & \sum_{i=1}^{N} u_{mi} x_{i12} & \dots & \sum_{i=1}^{N} u_{mi} x_{ipk} \end{pmatrix}. \tag{9.5}$$

This can be rearranged into a tensor $\mathcal{Y}_{m \times p \times k}$ with ijk^{th} element $\{\sum_{h=1}^{N} u_{ih} x_{hjk}\}$.

9.3 Higher Order Tensor Rank

The concept of the rank of a tensor is more complex than what is encountered in the context of matrices. Shedding light on tensor rank requires that the notion of a decomposable tensor be defined. A tensor $\mathcal{X} \in \Re^{d_1 \times \dots \times d_k}$ is said to be *decomposable* if it can be written in the form

$$\mathcal{X} = \mathbf{x_1} \circ \dots \circ \mathbf{x_k}, \tag{9.6}$$

where $\boldsymbol{x}_i \in \Re^{d_i}$ for $i = 1, \dots, k$ and $\circ$ is the outerproduct operator. The tensor $\mathcal{X}$ can thus be represented as the outerproduct of k vectors. Armed with this definition it is possible to formulate the definition of outer-product rank.

Definition 9.3.1. *A tensor $\mathcal{X}$ has* outer-product rank*, denoted by* $rank_{\circ}(\mathcal{X})$*, r if it can be written as the sum of r decomposable tensors but no fewer. Mathematically this is represented as*

$$rank_{\circ}(\mathbf{A}) := \mathbf{min}\{\mathbf{r}|\mathbf{A} = \sum_{\mathbf{i=1}}^{\mathbf{r}} \mathbf{u_i} \circ \mathbf{v_i} \circ \ldots \circ \mathbf{z_i}\}.$$

This definition was first introduced by Hitchcock (1927a) and then independently discovered by Kruskal (1977). There is currently no general means of determining the outer product rank of a tensor (Acar and Yener, 2009). The next conceptualisation of the rank of a tensor is referred to as *multilinear rank*. This definition generalizes the ideas of column and row rank of a matrix to higher order tensors. As a consequence of the fact that this dissertation is concerned with three mode data, the definition will only be given for tensors of order three. In order to define this concept some notation is required. Let $\mathcal{X} := [\![x_{ijk}]\!] \in \Re^{d_1 \times d_2 \times d_3}$. For fixed values of j and k consider the vector $\boldsymbol{x}_{\bullet jk} := [x_{ijk}]_{i=1}^{d_1} \in \Re^{d_1}$. In the same vein it is possible to define column vectors $\boldsymbol{x}_{i \bullet k}$ for fixed values of i and k and row vectors $\boldsymbol{x}_{ij\bullet}$ for fixed values of i and j. De Silva and Lim (2004) thus define multilinear rank as

$$\begin{aligned} r_1(\mathcal{X}) &:= dim(span_{\Re}\{\boldsymbol{x}_{\bullet jk} | 1 \leq j \leq d_2, 1 \leq k \leq d_3\}), \\ r_2(\mathcal{X}) &:= dim(span_{\Re}\{\boldsymbol{x}_{i\bullet k} | 1 \leq i \leq d_1, 1 \leq k \leq d_3\}), \\ r_3(\mathcal{X}) &:= dim(span_{\Re}\{\boldsymbol{x}_{ij\bullet} | 1 \leq i \leq d_1, 1 \leq j \leq d_2\}). \end{aligned} \tag{9.7}$$

Notice that the multilinear rank of the order three tensor,denoted by $rank_{\boxplus}(\mathcal{X})$ is defined by the 3-tuple $(r_1(\mathcal{X}), r_2(\mathcal{X}), r_3(\mathcal{X}))$. This concept was first introduced by Hitchcock (1927b) under the name *multiplex rank*. Despite the seemingly complex definition of multilinear rank there is a simple way to interpret it. The rank $r_1(\mathcal{X})$ is interpreted as the rank of the $d_1 \times d_2 d_3$ matrix $\boldsymbol{X}_{(1)}$ that is produced by unfolding the array $\mathcal{X}$ in the first mode. The $d_2 \times d_1 d_3$ and $d_3 \times d_1 d_2$ matrices $\boldsymbol{X}_{(2)}$ and $\boldsymbol{X}_{(3)}$ can be used to determine $r_2(\mathcal{X})$ and $r_3(\mathcal{X})$ respectively. The multilinear rank and outerproduct rank are in general all different, which is a complex departure from the case of matrices (Bader and Kolda, 2008).

Both multilinear rank as well as the outerproduct rank for a data matrix can be determined from the SVD of that matrix. The number of non-zero singular values is equivalent to the outerproduct rank of the matrix and this number corresponds to both the row and column rank of the matrix. This does not hold for tensors of order three and higher implying that the outerproduct rank and the multlinear rank need not be the same. Moreover, the multilinear rank can comprise different numbers so that the matrix concept of row and column rank being equivalent does not extend to higher order tensors.

9.4 Data preprocessing

The fact that PCA is not scale invariant and that the decomposition techniques to be discussed can be considered generalisations of PCA, (Kroonenberg, 1983) makes data preprocessing a necessary consideration in the context of tensor decomposition techniques. More specifically, the manner in which data is preprocessed can change the PCA biplot materially. This section offers a brief and simplistic outline of data preprocessing in a three mode context. The aim is really to introduce the reader to the complexities of preprocessing and to explore the effects on the graphical displays in later sections. Kroonenberg (2008) discusses two primary forms of preprocessing *viz.* centering and scaling. The former requires that a constant value be subtracted from every element in the data matrix and the latter refers to dividing every element in the data matrix by a constant, so that the *scale* of the resulting data values has a fixed value. In this context *scale* refers to some measure of variability, oftentimes the standard deviation. These operations are simple in the context of a two mode data matrix and their effects are well understood. Centering a $n \times p$ data matrix $\boldsymbol{X}$ requires that the relevant mean be subtracted from each element and scaling requires that each element be divided by the relevant standard deviation. The means and standard deviations are calculated from the columns of $\boldsymbol{X}$ and this is the only way to calculate these quantities. Preprocessing is considerably more complex in the context of three mode data because there are a number of different means and scaling factors to consider (Kroonenberg, 2008). In fact Kroonenberg (2008) states that three mode data allows three different sets of one-way means, three different sets of two mode means and an overall mean. Similarly, it is possible to define three different slice scaling factors, three different fiber scaling factors as well as an overall scaling factor. Table 9.1

	Mean	Scaling factor
one-way	$\overline{x}_{i..} = \frac{1}{PK}\sum_j \sum_k x_{ijk}$	$s_{i..} = \frac{1}{PK-1}\sum_j \sum_k (x_{ijk} - \overline{x}_{i..})^2$
two mode	$\overline{x}_{.jk} = \frac{1}{N}\sum_i x_{ijk}$	$s_{.jk} = \frac{1}{I-1}\sum_i (x_{ijk} - \overline{x}_{.jk})^2$
overall	$\overline{x}_{...} = \frac{1}{NPK}\sum_i \sum_j \sum_k x_{ijk}$	$s_{.jk} = \frac{1}{NPK}\sum_i \sum_j \sum_k (x_{ijk} - \overline{x}_{...})^2$

Table 9.1: Different possibilities for means and scaling factors.

illustrates how the various means and scaling factors can be calculated. The terms one-way and two mode can be interchanged with slice and fibre preprocessing. Not only are there considerably more scaling factors and means to consider but the choice of how to centre and scale is also not a simple one. The complexities of preprocessing can be untangled to an extent by understanding what informs the choice of centering and scaling as well as what the reasons for performing these operations are. The former aspect is considered first.

Kroonenberg (2008) suggests that the choice of preprocessing technique is informed by one of two types of arguments. The first of these arguments relies on choosing the preprocessing technique based on the measurement characteristics of the variables comprising the data as well as the research questions which may indicate a specific kind of preprocessing. This is referred to as *content-based* preprocessing. The second type of argument relies on choosing preprocessing methods that are allowed by the model to be used. Harshmann and Lundy (1984) provides an extensive theoretical framework for preprocessing techniques that are deemed appropriate and those that are not. This is referred to as *model-based* preprocessing. In spite of the fact that this latter argument exists, Harshmann and Lundy (1984) opine that preprocessing is largely informed by content-based arguments although model-based arguments do play a role. Inevitably, within the realm of what is deemed appropriate by model-based arguments, the choice of preprocessing is at the discretion of the researcher and hinges on content-based arguments. Content based considerations include aspects such as whether variances are comparable or should be made comparable and whether it is better to model relative to deviations from a base-line which is what centering would achieve.

Another important aspect of deciding on preprocessing is understanding why it needs to be done at all and what the effects of basic preprocessing techniques are. Kroonenberg (2008) suggests that preprocessing is paramount because the raw data may obscure the true relationships that are contained in the data. Without preprocessing the data may provide an inaccurate description of the relationships between the three modes in the data. The reason for normalisation is simple in that variables may have arbitrary or incommensurate measurement scales and scaling ensures that these differences do not influence the outcome of any analysis. A consequence of effectively creating this standard scale across variables is that “all parts of the data exercise equal influence on the outcomes of the analysis” (Kroonenberg, 2008). Centering can also be thought of as a means to remove undue influence of constants in the data. An example comes in the form of considering the effect of not removing the centroid from a two way data set when performing PCA; if the centroid is far from the origin in the space spanned by the variables then the first component is likely to run from the origin to the centroid (Kroonenberg, 2008). This influences the decision to centre the data.

It is important to note that there are several centerings and scalings that can be performed. Several slice or fiber centerings can be performed and this applies to scaling too. Once the data have been centered in a particular mode or across several modes, centering across a different mode cannot alter the centering of the data. In other words, several centerings can be performed without concern for undoing previous centering operations. Scaling is more complex in that several scalings can affect one another. Moreover, scaling perpendicular to the direction of centering will affect the centering operation. To understand this consider an $n \times p \times k$ tensor $\mathcal{X}$. Centering across the subject mode means subtracting the mean $\overline{x}_{i..}$. Slice normalisation in the variable mode (using $s_{.j.}$) would constitute scaling perpendicular to the centering operation and thus impact the centering (Kroonenberg, 2008). In this dissertation the focus will be on understanding the effects of the recommended form of preprocessing for profile data which is considered here. The preprocessing operation is defined as

$$x^*_{ijk} = \frac{x_{ijk} - \overline{x}_{.jk}}{s.j.}. \tag{9.8}$$

This combines fiber centering across the subject mode with slice centering in the variable mode. This type of centering is suggested because it ensures that scores on each variable represents the deviation from the average subject's profile. The slice normalisation ensures that all parts of the data have an equal influence on the analysis. This form of centering is favoured on a model-based argument basis because it ensures that if a three mode model is valid for the raw data then it still holds after this form of preprocessing has been applied (Kroonenberg, 2008). This type of preprocessing is also similar to that seen in the application of two mode PCA. It is important to reiterate that this section served to introduce the reader to the complexities of data preprocessing in the three mode context. Furthermore, although a vast body of literature exists on the subject it is not geared towards understanding the effects of preprocessing on an exploratory graphical display but rather speaks to the effects in a modelling context. For a comprehensive discussion on data preprocessing, the interested reader is directed to Harshmann and Lundy (1984).

9.5 Multilinear rank decomposition

The basic definitions together with the definition of rank in the tensor context affords the means to define a multilinear tensor decomposition, which is in fact the Tucker3 Decomposition with orthogonality constraints. The development in this section follows that of De Lathauwer *et al.* (2000). In order to see why this decomposition has been referred to as Higher Order Singular Value Decomposition (HOSVD) by De Lathauwer *et al.* (2000), the definition of SVD is restated in different notation in order to facilitate comparison to the HOSVD. Recall that SVD can be defined as follows:

Definition 9.5.1. *Every real matrix $\boldsymbol{X}$ with size $n \times p$ can be written as the product*

$$\boldsymbol{X} = \boldsymbol{U}^{(1)}\boldsymbol{D}(\boldsymbol{V}^{(2)})' = \boldsymbol{D} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{V}^{(2)} \tag{9.9}$$

where $\boldsymbol{U}^{(1)}$ is an $n \times n$ unitary matrix; $\boldsymbol{V}^{(2)}$ is an $p \times p$ unitary matrix and $\boldsymbol{D}$ is an $n \times p$ matrix with properties of

- *pseudo-diagonality so that $\boldsymbol{D} = diag(\sigma_1, \sigma_2, ..., \sigma_{min(n,p)})$,*
- *ordering so that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{min(n,p)} \geq 0$.*

Each of the σ_i are the singular values of $\boldsymbol{X}$ and the columns of $\boldsymbol{U}^{(1)}$ and $\boldsymbol{V}^{(2)}$ represent the left and right singular vectors respectively. These matrices are in fact orthonormal bases for the n^{th} mode of the matrix $\boldsymbol{X}$.

It is now possible to define the HOSVD as follows:

Definition 9.5.2. *(De Lathauwer et al., 2000) Any third order tensor $\mathcal{X} \in \Re^{d_1 \times d_2 \times d_3}$ can be represented as the product*

$$\mathcal{X} = \mathcal{D} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \boldsymbol{U}^{(3)} \tag{9.10}$$

where $\boldsymbol{U}^{(n)}$ is an $d_n \times d_n$ unitary matrix, $\mathcal{D}$ is a tensor $\in \Re^{d_1 \times d_2 \times d_3}$ of which the subtensors $\mathcal{D}_{d_n=\alpha}$ obtained by setting the n^{th} index to α has the following properties

- *all-orthogonality so that two subtensors $\mathcal{D}_{d_n=\alpha}$ and $\mathcal{D}_{d_n=\beta}$ are orthogonal for all possible values of n, α, β subject to $\alpha \neq \beta$. This means that $< \mathcal{D}_{d_n=\alpha}, \mathcal{D}_{d_n=\beta} >= 0$ when $\alpha \neq \beta$*
- *ordering so that $\| \mathcal{D}_{d_n=1} \| \geq \| \mathcal{D}_{d_n=2} \| \geq \ldots \geq \| \mathcal{D}_{d_n=D_n} \|$.*

The Frobenius norm of the subtensor $\| \mathcal{D}_{d_n=i} \|$ denoted by $\sigma_n^{(i)}$ are referred to as the n-mode singular values of $\mathcal{X}$ and $\boldsymbol{U}^{(n)}$ comprises the n-mode singular vectors.

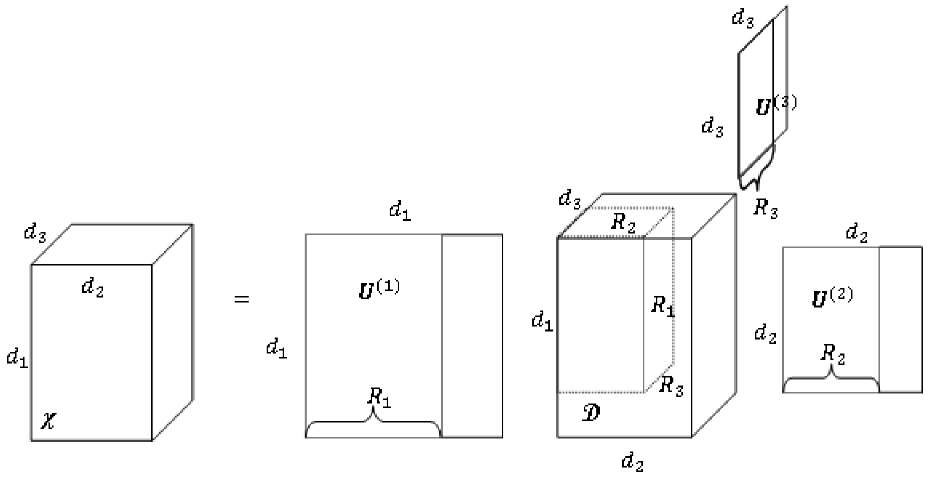


Figure 9.2: Visual representation of the Tucker3 decomposition of $\mathcal{X}$.

Figure 9.2 affords a visual illustration of the Tucker3 decomposition. It is interesting to compare the SVD definition to that of HOSVD definition in order to understand the similarities and differences between them. One key property of the SVD is that it defines an orthonormal basis for each of the row and column spaces of an $n \times p$ matrix $\boldsymbol{X}$. The columns of the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ represent the orthogonal bases for the row and column space of $\boldsymbol{X}$ respectively. This property is preserved in the HOSVD context where the matrices $\boldsymbol{U}^{(n)}$ represent orthonormal bases for each of the n-mode vector spaces represented. Furthermore the ordering property of the singular values is also preserved. The difference lies in the fact that the tensor $\mathcal{D}$ is a full tensor rather than being pseudo-diagonal which would imply that non-zero entries could only occur when indices $i_1 = \ldots = i_N$ as in the SVD instance. HOSVD requires that the tensor $\mathcal{D}$ adhere to the property of all-orthogonality which is also obeyed by the matrix $\boldsymbol{D}$ in the case of SVD. De Lathauwer *et al.* (2000) list a myriad other properties that strengthen the notion that HOSVD is one generalisation of SVD. The previous remark raises an important fact in that although HOSVD is considered a generalisation of SVD, it is not the only generalisation. SVD allows a matrix to be uniquely expressed as the sum of the outerproduct of two vectors ($\boldsymbol{X} = \sum_{i=1}^{R} \boldsymbol{u}_i \circ \boldsymbol{v}_i$ where R is the rank of the matrix $\boldsymbol{X}$). This property does not extend to HOSVD. A class of models that preserves this property will be discussed later. In essence, this decomposition is based on the mulitlinear rank of the tensor $\mathcal{X}$. Before considering whether the truncated HOSVD provides the best low multilinear rank approximation in the least squares sense it is instructive to put the HOSVD into a more familiar matrix representation akin to the SVD representation.

A matrix representation of the HOSVD is simply obtained by the process of unfolding the tensors $\mathcal{X}$ and $\mathcal{D}$ in equation (9.10). As an example consider the unfolding in the first mode which yields

$$\boldsymbol{X}_{(1)} = \boldsymbol{U}^{(1)}\boldsymbol{D}_{(1)}(\boldsymbol{U}^{(3)} \otimes \boldsymbol{U}^{(2)})'. \tag{9.11}$$

Similarly, expanding in the second and third modes yields $\boldsymbol{X}_{(2)} = \boldsymbol{U}^{(2)}\boldsymbol{D}_{(2)}(\boldsymbol{U}^{(3)} \otimes \boldsymbol{U}^{(1)})'$ and $\boldsymbol{X}_{(3)} = \boldsymbol{U}^{(3)}\boldsymbol{D}_{(3)}(\boldsymbol{U}^{(2)} \otimes \boldsymbol{U}^{(1)})'$. This still does not correspond to the SVD definition that is most familiar *viz.* $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{'}$. For the purpose of illustration, the unfolding in the first mode will be used. Define the $d_1 \times d_1$ diagonal matrix $\boldsymbol{\Sigma}^{(1)}$

$$\boldsymbol{\Sigma}^{(1)} := diag(\sigma_1^{(1)}, \sigma_2^{(1)}, \sigma_3^{(1)}, \ldots, \sigma_{d_1}^{(1)}), \tag{9.12}$$

as well as the column-wise orthonormal matrix $(\boldsymbol{V}^{(1)})'$ given by

$$(\boldsymbol{V}^{(1)})' = \tilde{\boldsymbol{D}}_{1}.(\boldsymbol{U}^{(3)} \otimes \boldsymbol{U}^{(2)})', \tag{9.13}$$

where

$$\tilde{\boldsymbol{D}}_{(1)} = (\boldsymbol{\Sigma}^{(1)})^{-1}\boldsymbol{D}_{(1)}, \tag{9.14}$$

is a normalized version of $\boldsymbol{D}_{(1)}$. De Lathauwer *et al.* (2000) state that the matrices $\boldsymbol{\Sigma}_{(1)}$ and $(\boldsymbol{V}^{(1)})'$ make it possible to represent the HOSVD as an SVD of the matrix unfolding $\boldsymbol{X}_{(1)}$ given by

$$\boldsymbol{X}_{(1)} = \boldsymbol{U}^{(1)}\boldsymbol{\Sigma}_{(1)}(\boldsymbol{V}^{(1)})'. \tag{9.15}$$

The only question left to answer is whether truncating the HOSVD by simply using the first two columns of each of the matrices comprising the SVD of the unfolded matrix in (9.15) yields the best low multilinear rank-(2,2,2) approximation of the tensor $\mathcal{X}$ in a least squares sense. In Chapter 4 it was shown that ordinary SVD yields the best low rank approximation of the matrix by simply considering the truncated form of the SVD. Kroonenberg and De Leeuw (1977) made an important contribution by devising a least squares algorithm to estimate $\mathcal{D}$ and $\boldsymbol{U}^{(1)}$,$\boldsymbol{U}^{(2)}$,$\boldsymbol{U}^{(3)}$. Some attention will be given to detailing the mechanism by which this algorithm operates. Formally, the problem at hand can be thought of as seeking the best rank-(R_1, R_2, R_3) approximation, $\hat{\mathcal{X}}$ to the $N \times P \times K$ tensor $\mathcal{X}$. Any least squares solution would seek to minimise the error, implying that the function to be minimsed can be defined as

$$\parallel \mathcal{X} - \hat{\mathcal{X}} \parallel^2 = \parallel \boldsymbol{X}_{(1)} - \boldsymbol{U}^{(1)}\boldsymbol{D}_1(\boldsymbol{U}^{(3)} \otimes \boldsymbol{U}^{(2)})' \parallel^2, \tag{9.16}$$

where $\boldsymbol{D}_{(1)}$ has dimensions $R_1 \times R_2R_3$ and $\boldsymbol{U}^{(1)}$,$\boldsymbol{U}^{(2)}$ and $\boldsymbol{U}^{(3)}$ are column-wise orthonormal matrices with dimensions $N \times R_1$, $P \times R_2$ and $K \times R_3$ respectively. The algorithm works on the basis of iteratively estimating one of the four parameters $\mathcal{D}, \boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}$ and $\boldsymbol{U}^{(3)}$ conditional on the remaining parameters being fixed. Notice that the unfolded core array, $\boldsymbol{D}_{(1)}$ can be expressed in terms of $\boldsymbol{X}_{(1)}$ together with the component matrices implying that

$$\boldsymbol{D}_{(1)} = (\boldsymbol{U}^{(1)})'\boldsymbol{X}_{(1)}(\boldsymbol{U}^{(3)} \otimes \boldsymbol{U}^{(2)}). \tag{9.17}$$

Substituting (9.17) into the unfolding of the approximation $\hat{\boldsymbol{X}}_{(1)}$ yields

$$\begin{aligned}\boldsymbol{U}^{(1)}\boldsymbol{D}_1(\boldsymbol{U}^{(3)} \otimes \boldsymbol{U}^{(2)})' &= \boldsymbol{U}^{(1)}(\boldsymbol{U}^{(1)})'\boldsymbol{X}^{(1)}(\boldsymbol{U}^{(3)} \otimes \boldsymbol{U}^{(2)})(\boldsymbol{U}^{(3)} \otimes \boldsymbol{U}^{(2)})' \\ &= \boldsymbol{U}^{(1)}(\boldsymbol{U}^{(1)})'\boldsymbol{X}^{(1)}(\boldsymbol{U}^{(3)}(\boldsymbol{U}^{(3)})' \otimes \boldsymbol{U}^{(2)}(\boldsymbol{U}^{(2)})'). \quad (9.18)\end{aligned}$$

Andersson and Bro (1998) define $\boldsymbol{M} = \boldsymbol{X}^{(1)}(\boldsymbol{U}^{(3)}(\boldsymbol{U}^{(3)})' \otimes \boldsymbol{U}^{(2)}(\boldsymbol{U}^{(2)})')$ which allows (9.18) to be expressed as $\boldsymbol{U}^{(1)}(\boldsymbol{U}^{(1)})'\boldsymbol{M}$. Expanding (9.16) with respect to the trace operator results in

$$\begin{aligned}&tr\big((\boldsymbol{X}_{(1)} - \boldsymbol{U}^{(1)}(\boldsymbol{U}^{(1)})'\boldsymbol{M})(\boldsymbol{X}_{(1)} - \boldsymbol{U}^{(1)}(\boldsymbol{U}^{(1)})'\boldsymbol{M})'\big) \\ &= tr(\boldsymbol{X}_{(1)}\boldsymbol{X}_{(1)}') - 2tr(\boldsymbol{U}^{(1)}(\boldsymbol{U}^{(1)})'\boldsymbol{M}\boldsymbol{X}_{(1)}') + tr(\boldsymbol{U}^{(1)}(\boldsymbol{U}^{(1)})'\boldsymbol{M}\boldsymbol{M}'\boldsymbol{U}^{(1)}(\boldsymbol{U}^{(1)})').\end{aligned} \tag{9.19}$$

Using the fact that $\boldsymbol{M}\boldsymbol{X}'$ is equivalent to $\boldsymbol{M}\boldsymbol{M}'$, $\boldsymbol{U}^{(1)}$ is column-wise orthonormal and that $tr(\boldsymbol{X}\boldsymbol{X}')$ is fixed, minimising (9.19) is equal to minimising

$$\begin{aligned}&-2tr(\boldsymbol{U}^{(1)}(\boldsymbol{U}^{(1)})'\boldsymbol{M}\boldsymbol{X}_{(1)}') + tr(\boldsymbol{U}^{(1)}(\boldsymbol{U}^{(1)})'\boldsymbol{M}\boldsymbol{M}'\boldsymbol{U}^{(1)}(\boldsymbol{U}^{(1)})') \\ &= -tr((\boldsymbol{U}^{(1)})'\boldsymbol{M}\boldsymbol{M}'\boldsymbol{U}^{(1)}).\end{aligned} \tag{9.20}$$

Given that the development assumes that all parameters but $\boldsymbol{U}^{(1)}$ are fixed, (9.19) is being minimized in $\boldsymbol{U}^{(1)}$. Minimising (9.19) is thus equivalent to maximising $tr((\boldsymbol{U}^{(1)})'\boldsymbol{M}\boldsymbol{M}'\boldsymbol{U}^{(1)})$ (Andersson and Bro, 1998). As a result of the fact that $\boldsymbol{M}\boldsymbol{M}'$ is symmetric and that the trace of a matrix is equal to the sum of its eigenvalues it is clear that the optimal matrix $\boldsymbol{U}^{(1)}$ is given by the first R_1 left singular vectors of the SVD of $\boldsymbol{M}$. The estimation of the optimal $\boldsymbol{U}^{(2)}$ and $\boldsymbol{U}^{(3)}$ follows a similar process as that described for $\boldsymbol{U}^{(1)}$

and this forms the basis for the least squares algorithm. Before detailing the skeleton of the alternating least squares (ALS) algorithm some thought must be given to what is to be used as the convergence criterion. Closer inspection of $tr((\boldsymbol{U}^{(1)})'\boldsymbol{M}\boldsymbol{M}'\boldsymbol{U}^{(1)})$ reveals that it is in fact equivalent to $\| \boldsymbol{D}_{(1)} \|^2$. Andersson and Bro (1998) contend that this "provides a robust and monotonically increasing parameter that may be used to detect convergence". A generic algorithm for estimation is constructed as follows:

1. Initialise $\boldsymbol{U}^{(2)}$ and $\boldsymbol{U}^{(3)}$.
2. Calculate $\boldsymbol{M}^{(1)}$ from $\boldsymbol{U}^{(2)}$, $\boldsymbol{U}^{(3)}$ and $\boldsymbol{X}_{(1)}$. Find $\boldsymbol{U}^{(1)}$ as described.
3. Calculate $\boldsymbol{M}^{(2)}$ from $\boldsymbol{U}^{(3)}$, $\boldsymbol{U}^{(1)}$ and $\boldsymbol{X}_{(2)}$. Find $\boldsymbol{U}^{(2)}$ as described.
4. Calculate $\boldsymbol{M}^{(3)}$ from $\boldsymbol{U}^{(1)}$, $\boldsymbol{U}^{(2)}$ and $\boldsymbol{X}_{(3)}$. Find $\boldsymbol{U}^{(3)}$ as described.
5. Check for convergence based on $\| \boldsymbol{D}_{(1)} \|^2$. If convergence occurs, terminate else return to step 1.

This algorithm is guaranteed to converge to a local minimum. In order to check the stability of the solution Kroonenberg (2008) suggests initialising $\boldsymbol{U}^{(2)}$ and $\boldsymbol{U}^{(3)}$ in different ways and comparing the resulting estimates. The most common way to initialise these matrices is by using the first R_2 and R_3 left singular vectors of $\boldsymbol{X}_{(2)}$ and $\boldsymbol{X}_{(3)}$ respectively and that is the convention adopted in this dissertation. There is a vast body of literature on various methods of introducing orthogonality constraints to the HOSVD, but given that the purpose here is to explore methods of representing a decomposition, these are not considered here. Andersson and Bro (1998) provides a comprehensive overview of the various methods used. Recall that the primary consideration was whether the truncated HOSVD yielded the best rank-(R_1, R_2, R_3) decomposition of the tensor $\mathcal{X}$ in a least squares sense and it turns out that the truncated HOSVD solution tends not to agree with the least squares solution which is deemed the best rank-(R_1, R_2, R_3) approximation in a least square sense. The least squares estimation procedure was implemented using the *N-Way Toolbox* in Matlab that was developed by Andersson and Bro (2000).

9.5.1 The Tucker3 biplot

With due consideration having been given to the development as well as the estimation of the parameters of the Tucker3 or HOSVD decomposition it is now possible to consider the construction of what Kroonenberg (2008) refers to as a nested-mode biplot. The construction will be based on the

HOSVD representation given in (9.15). It is known that $\boldsymbol{X}_{(1)}$ has dimensions $N \times PK$ so the constructed biplot will comprise N samples and PK variable axes. Two modes, the occasion and variable mode, are thus nested giving rise to the name used by Kroonenberg (2008). Chapter 4 detailed the construction of the biplot based on the SVD and following the same methodology discussed there it is simply a case of plotting the rows of $\boldsymbol{G} = \boldsymbol{U}^{(1)}\boldsymbol{\Sigma}_{(1)}$ to represent the observations and the rows of the $\boldsymbol{H} = \boldsymbol{V}^{(1)}$ to represent the variable axes with the former in principal-coordinates and the latter in normalised co-ordinates. Geometrically the data are being represented in the space $\Re^{PK}$ and the N sample points are being projected onto the plane, the basis for which comprises the columns of $\boldsymbol{U}^{(1)}$.

A rather obvious question is the choice of values for R_1,R_2 and R_3. $\boldsymbol{G}$ has dimensions $N \times R_1$ and $\boldsymbol{H}$ has dimensions $PK \times R_1$. It is thus imperative that $R_1 = 2$ and in general that the mode in which the matrix be unfolded is approximated with two components. The values for R_2 and R_3 can be arbitrary though Kroonenberg (1983) provides restrictions for these values. It is certainly possible to fix R_1 to be two and compare the fit of decompositions for various values of R_2 and R_3 based on the amount of explained variability, however this speaks to dimensionality selection which is beyond the scope of this dissertation. For the purpose of this dissertation it is convenient to find the best rank-$(2, 2, 2)$ approximation. The same reasoning can be applied to $\boldsymbol{X}'_{(2)} = \boldsymbol{V}^{(2)}\boldsymbol{\Sigma}_{(2)}(\boldsymbol{U}^{(2)})'$. Geometrically, the data are being represented in the space $\Re^{P}$ and the NK sample points are being projected onto the plane, the basis for which is the columns of $\boldsymbol{U}^{(2)}$. There is a key difference between the construction of the Tucker3 biplot and the PCA biplots considered earlier. Where as PCA biplot construction relies on the use of the right-singular vector matrix $\boldsymbol{V}$, the Tucker3 model estimates left-singular vector matrices $\boldsymbol{U}$ and these are used in the process of constructing the Tucker3 biplot. All the interpretational tools that are valid for the PCA biplot apply to the Tucker3 biplot (Kroonenberg, 2008).

9.5.2 Application to Mansoor data

With the construction process of the Tucker3 biplot having been explained, it is now possible to apply this method to the Mansoor data in order to see what these biplots reveal about the structure of the data. In the application, the centered data as well as the centered and scaled data were used to construct the biplots. The centering and scaling used is that which is shown in (9.8). A Tucker3 model was fitted with two components in each of the modes. It is important to remain cognisant of the fact that after estimating the Tucker3 model, the data were matricised to form a 29 × 28 data matrix akin to the wide combination matricisation seen in Chapter 4. This will be termed the wide Tucker3 biplot. Similarly, the data were matricised to form a 116 × 7 matrix akin to the tall combination matricisation seen in Chapter 4 and this will be termed the tall Tucker3 biplot. These biplots are assessed on how well they capture the structure inherent in the data.

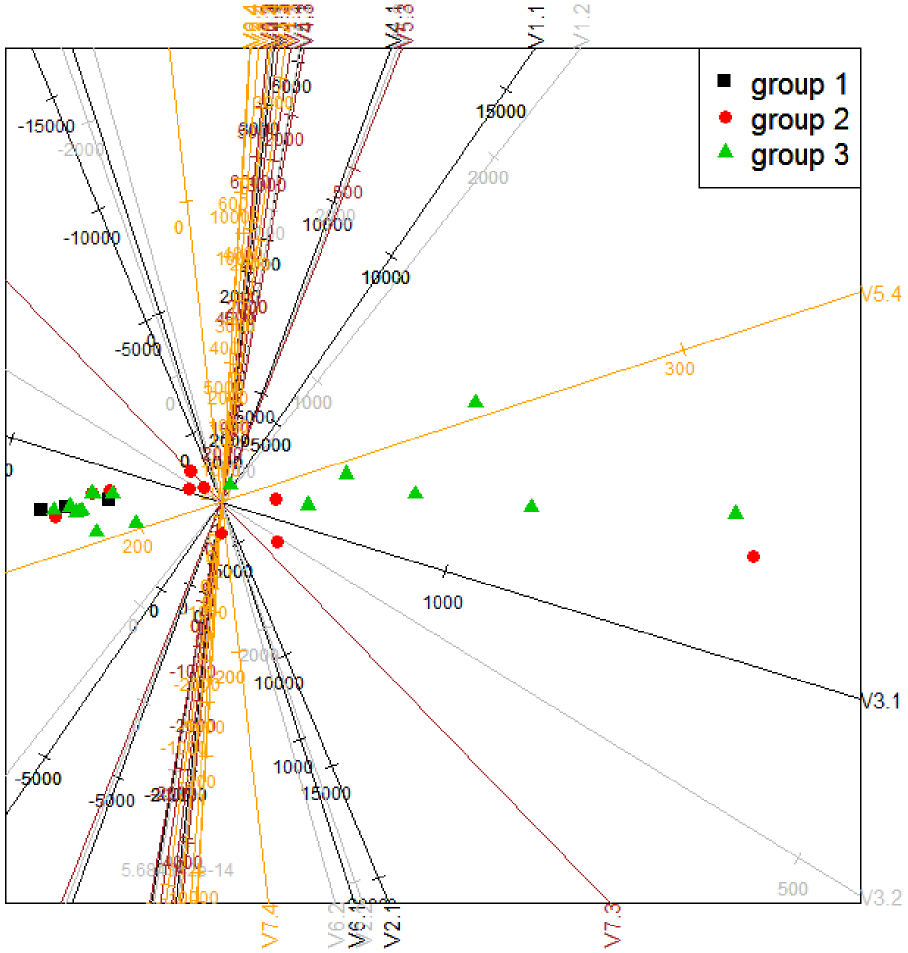


Figure 9.3: Wide combination Tucker biplot for centered Mansoor data.

Figure 9.3 illustrates the wide Tucker3 biplot for the centered Mansoor data. What is immediately obvious is that the distribution of the observations is similar to that seen in Figure 4.7. By projecting onto the variable axes it is evident that there is a reduction in variation over time. If occasion 4 is considered as an example it is clear that the observations projected onto the variable axes for this occasion display relatively small variation when compared to occasion 1. Moreover it generally seems to be the case that infants comprising the HIV^{+} group achieved relatively lower scores across occasion when compared with the groups of uninfected infants. Although the latter observation is also made in Figure 9.4, the wide Tucker3 biplot for the centered and scaled data, the reduction in variation is not immediately distinguishable in this plot. The technique is thus not scale invariant. This is further evidenced by the fact that the variable axes look markedly different in the respective plots. The fiber centering employed implies that the angles between axes provide an approximation to the correlation between the relevant variables. Consider $V5$ and $V7$ for example. The correlation between these two variables was seen to be 0.73, 0.86, 0.36 and 0.17 over the four occasions. This correlation profile can be seen in Figure 9.3 with the variable axes being near orthogonal at occasions 3 and 4 but much closer at occasions 1 and 2. It is also revealed that V3 at occasions 1 and 2 have an association and this is corroborated by the correlation coefficient of 0.55. Not all associations are well represented. The orientation of the variable axes for occasion 4 suggests that all variables are associated bar $V5$. $V6$ is not strongly correlated with any of the variables except $V7$ and this is clearly not well represented on the biplot. It is important to be aware of the fact that this small angles between variable axes suggest that relationships between the relevant variables should be investigated by considering the correlation matrix. The biplot will never represent two variables axes as nearly orthogonal if they do in fact have a relationship. Comparing Figure 9.4 to the correlation matrices for the data reveals that this biplot does relatively well in conveying the associations between variables accurately. Naturally, not all associations will be well represented because the sample points are in principal co-ordinates and this is based on a model approximation to the data. In spite of this, these biplots suggest similar answers to the questions put forward in the Mansoor *et al.* (2009) investigation as those methods that have been discussed previously. Notice the extensive negative values for the calibrations on each of the axes for the tall Tucker biplots. Although it is not common, some observations clearly score negative values in Figure 9.5. This is a product of the approximation. The model does not fit the data perfectly.

Figures 9.5 and 9.6 illustrate the tall Tucker3 biplot for the centered as well as the centered and scaled data respectively. One notable difference in these biplots when compared to Figure 4.8 is that the axes are calibrated with multiple markers. This is because of the fiber centering that was used in

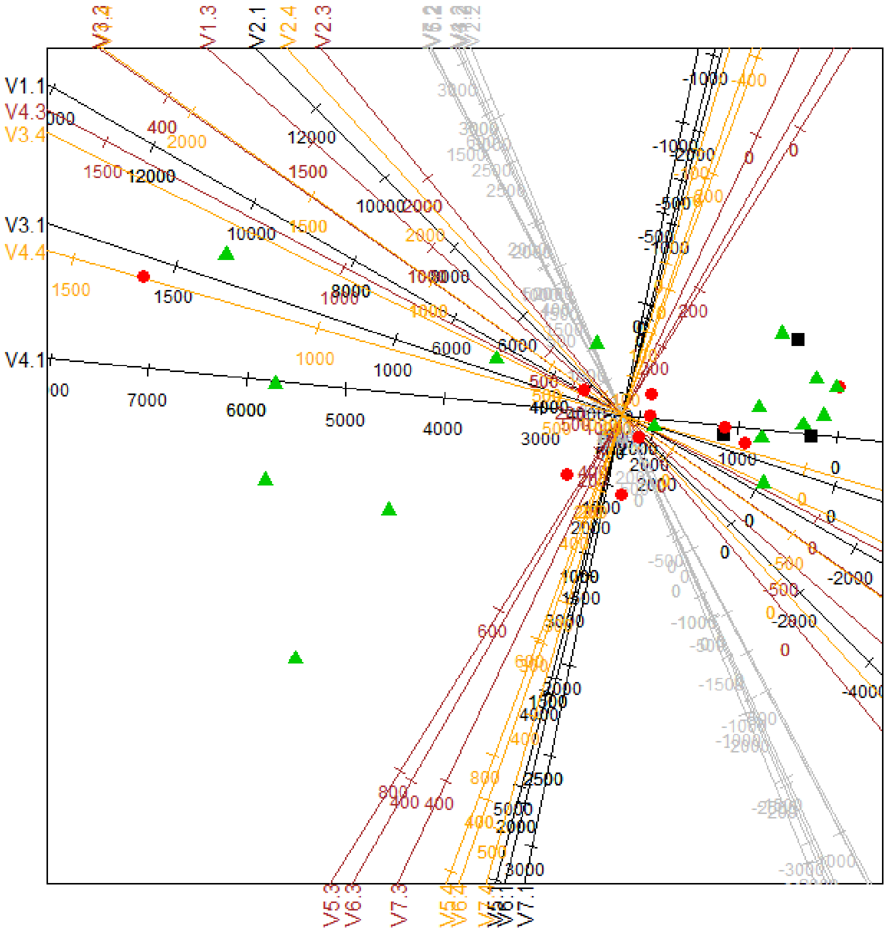


Figure 9.4: Wide combination Tucker biplot for centered and scaled Mansoor data.

preprocessing the data. In Chapter 4, the data was simply matricised to form a tall combination matrix and PCA applied to produce a biplot. The mean used in the calibration of each axes was an overall mean calculated from all the observations for each variable. In this context however the fiber centering is performed prior to the Tucker3 analysis and so by similar reasoning to that used in Chapter 7 on CPC biplots, multiple markers must be used to calibrate the axes. Alternatively a single set of markers can be used which would represent deviations from the overall mean for each of the variables. One of the key revelations of the tall combination PCA biplot was that the variation in the data changes over time. This can be seen in Figure 9.5 with occasion 1 displaying the most variation followed by a distinct reduction in variation

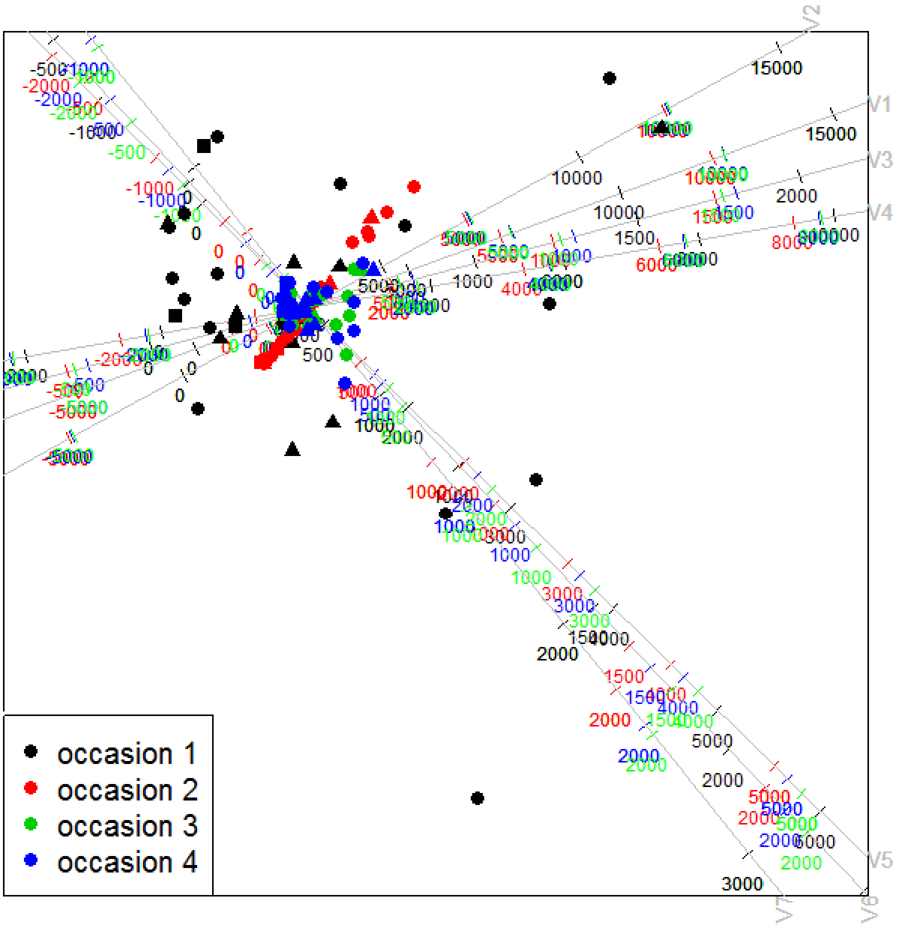


Figure 9.5: Tall combination Tucker biplot for centered Mansoor data.

over time. This phenomenon is not as obvious in Figure 9.6 however there is a clear reduction in variation from occasion 1 to occasion 2. Occasions 3 and 4 show similar variation. It is interesting to note that the orientation of the axes in Figure 9.6 is quite similar to that seen in Figure 4.7. The biplot in Figure 9.5 does not look quite as similar however it does reveal similar information about the associations between variables with $V5$ and $V7$, $V1$ and $V4$ as well as $V2$ and $V6$. These pairs of variables show similar relationships here and in Figure 4.7. The reason why the similarity between Figure 9.6 and Figure 4.8 is so interesting is because the former biplot is based on scaled data where as the latter biplot is not constructed from scaled data yet they reveal similar information about variable associations. This could be because slice normalisation ensures equal influence across occasion but does preserve differences in variation within an occasion. The fact that the variation in the data is studied from the perspective of the left-singular vectors as opposed to the right-singular vectors could also play a role. Ultimately these biplots do not indicate anything substantially different to the tall combination PCA biplots.

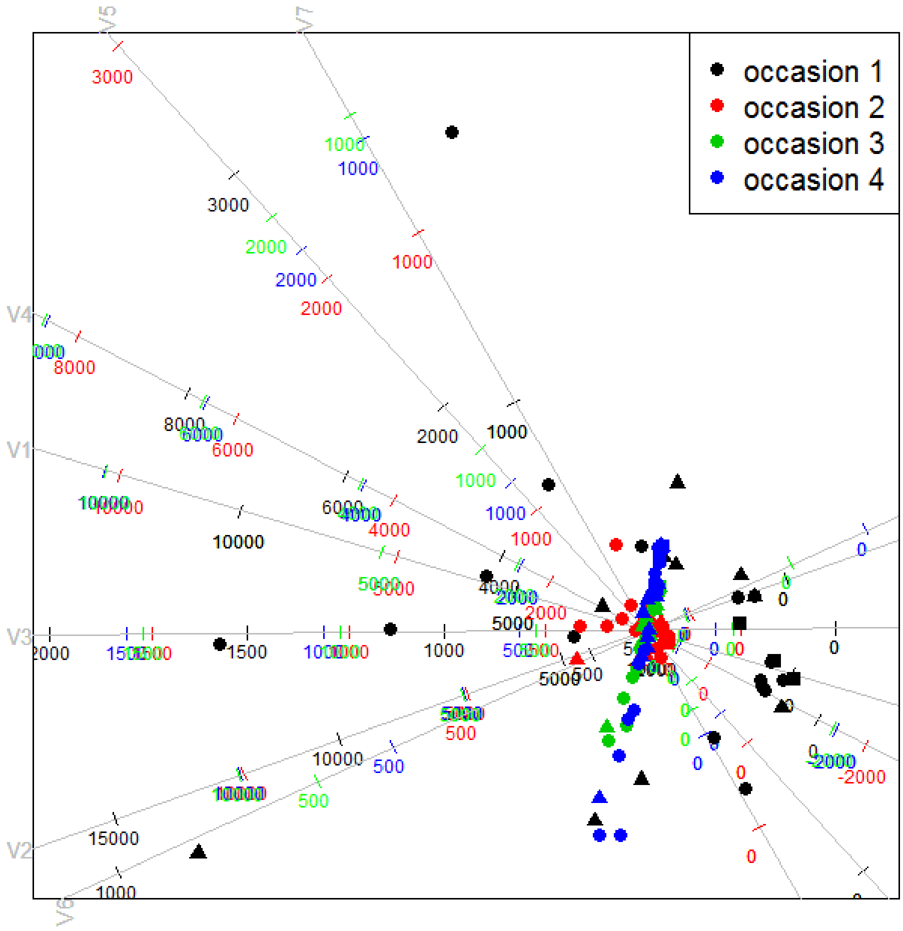


Figure 9.6: Tall combination Tucker biplot for centered and scaled Mansoor data.

9.6 Outerproduct rank decomposition

Although the multilinear decomposition of a tensor is amenable to producing biplots, it is important to note that this can only be done after the tensor

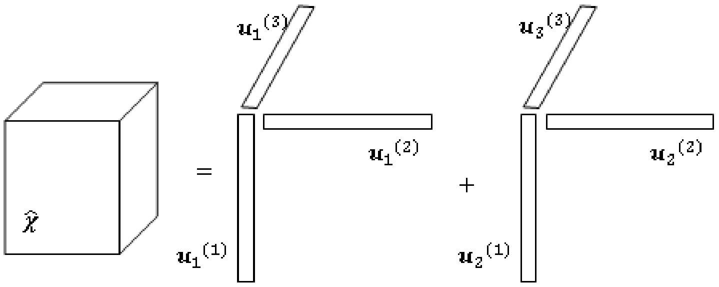


Figure 9.7: Visual representation of a triadic decomposition of $\hat{\mathcal{X}}$.

has been unfolded. Kiers (2000a) remarked that these plots “rely on two mode PCA models, obtained after rewriting the three mode model at hand”. The question is thus whether it is possible to construct a plot that does not rely on this unfolding process and captures information about all ways comprising the data, a triplot. The reason why the HOSVD was not used in considering the construction of such a plot was because it did not preserve the property of the unique link between components. To better understand this, consider Figure 9.7. In order to develop a triplot, the ideal would be to have a unique decomposition of the tensor that allows each mode to be displayed in a single plot. The HOSVD can be represented as a triadic decomposition, however components are not uniquely linked and as such choosing which set of components to represent is not a simple task. A unique triadic decomposition would remove this problem and make it a simple matter. In order to construct such a decomposition it is necessary to consider the outerproduct rank decomposition of the tensor which displays this uniqueness property that is sought. This is precisely the type of decomposition provided by the PARAFAC model. It can be thought of as another generalisation of SVD that preserves the uniqueness property of the components and seeks to determine the best rank R approximation where R is related to the concept of outerproduct rank. Define three component matrices $(\boldsymbol{U}^{(1)})_{N\times R}$, $(\boldsymbol{U}^{(2)})_{P\times R}$ and $\boldsymbol{U}^{(3)}_{K\times R}$ so that

$$\hat{\mathcal{X}} = \sum_{r=1}^{R} \boldsymbol{u}_r^{(1)} \circ \boldsymbol{u}_r^{(2)} \circ \boldsymbol{u}_r^{(3)}. \tag{9.21}$$

This is essentially the PARAFAC model. It should be noted that if R is chosen so that it corresponds with the outerproduct rank of a tensor $\mathcal{X}$ then the model will fit perfectly. Given the fact that the columns of each of the component matrices are uniquely linked to another so that the first columns in each of the matrices $\boldsymbol{U}^{(1)}$, $\boldsymbol{U}^{(2)}$ and $\boldsymbol{U}^{(3)}$ are exclusively linked for example it is only natural to assert the existence of a single set of components with different coefficients in each of the modes. For the sake of completeness, some statistical context is provided. The year 1970 saw the development of the PARAFAC model by two independent researchers, Harshmann (1970) as well as Carroll and Chang (1970). Harshmann (1970) conceived this model as an extension of component analysis and with Cattell's (1944) parallel proportional profiles principle he managed to show that PARAFAC solved the problem of rotational indeterminacy that plagued ordinary two-mode PCA as well as the Tucker3 model. This model was independently developed by Carroll and Chang (1970) and called *Canonical Decomposition* (CANDECOMP). The contribution of the latter authors came in the form of tackling algorithmic aspects of the model rather than developing it for the analysis of standard three mode data (Kroonenberg, 2008). It was Harshmann (1970) who developed the model for standard three mode data and in order to understand the modelling principle underpinning the PARAFAC model it is necessary to explore his motivations.

The foundation of two mode factor analysis is the notion that groups of highly correlated variables represent single underlying constructs or factors responsible for the observed correlations and the researcher wishes to uncover these factors and give meaning to them. Harshmann (1970) contends that what is fundamental to using factor analysis as a tool of scientific discovery is "the distinction between its 'descriptive' and its 'explanatory' application". The former application is concerned with finding a convenient and simplified representation of the relationships in the data where as the explanatory application seeks to provide sound estimates of the true underlying influences responsible for the structure of the observed data. This duality speaks directly to the use of methods like varimax rotation to simplify the factors and make them more interpretable. Harshmann (1970) refers to this as the factor analysis solution not being sufficiently constrained by the data leaving the problem of rotational indeterminacy. It is thus an arbitrary exercise to chose the solution which is most convenient and simplest to interpret.

This is the problem that Harshmann (1970) sought to address in the development of the PARAFAC model as a statistical data analysis tool. His solution rested on the parallel proportional profiles idea put forward by Cattel in 1944. Cattel (1944) suggested that in order for a factor to correspond to some real "organic unity" in the data, it would surely retain its pattern from one study to another changing only in its impact in the latter study. This means that whilst the pattern is retained, the loadings are simultaneously raised or lowered to account for the fact that the impact of the factor will differ from one study to another. The factors were thus the same in each study, accounting for the *parallel* aspect of the nomenclature but they would vary proportionately across studies. He further contended that no arbitrary "mathematical abstraction" would exhibit such behaviour. This argument lay at the heart of the PARAFAC model but it imposes rather stringent conditions on the nature of the variation in the data. The data must exhibit strong system variation and this concept is best illuminated by way of an example taken from Harshmann (1970). Consider an economic system in which a data set comprises a number of businesses measured on a number of variables over a number of months. Here the notion of system variation is reasonable since there are underlying factors that affect all businesses to varying degrees. One such factor could be inflationary pressure. Data of this nature lends itself to analysis with the PARAFAC model, however in the absence of this type of system variation, applying the PARAFAC model is problematic.

9.6.1 Degeneracy

This leads into the discussion on *degeneracy.* It is a well studied phenomenon that Harshmann (1970) suggests results from applying the PARAFAC model to data that contains more unique variation which he terms Tucker variation. In order to appreciate the problem of degeneracy it is necessary to have a basic understanding of what the PARAFAC model seeks to do on a more intuitive level. Consider the frontal slices of a three mode data array. The PARAFAC model is geared towards finding a common set of component axes across each of the frontal slices called factors. These factors have the same orientation across slices but can be stretched or shrunk to indicate the importance of a particular factor at each level of the third mode. In the absence of true systemic variation however the factors need not only be reweighted but their orientation, the angle between the axes, must also change. When Tucker (1966) originally introduced the Tucker3 model there was a form that was unconstrained. The HOSVD is a constrained form of the Tucker3 model. As a result of the fact that the unconstrained Tucker3 model would allow for axes orientation to change, Harshmann referred to data containing this kind of variation as Tucker variation. The PARAFAC model attempts to compromise when faced with data of this nature by introducing a shearing operator in one mode which forces the angles between the factors to change. To compensate for this an anti-shear which has the opposite affect of the shearing operator is applied in a second mode, the variable mode for example. This shearing effect is illustrated in Figure 9.8 where the original shape is a rectangle with orthogonal axes as seen on the left. The effect of shearing is to force the y-axis closer to the x-axis thereby changing the angle between the axes. This in fact causes the coefficient estimates to diverge but their contribution to the convergence criterion tends to cancel out which may lead to convergence at times (Harshmann, 2004). There are instances where this compromise that the PARAFAC model tries to make is not sufficient and no solution can be found. This provides an intuitive understanding of the notion of *degeneracy.* With respect to the estimated component matrices, the anti-shear tends to create highly negatively correlated columns in one of the component matrices which is evidence of degeneracy. Furthermore, if the estimating algorithm takes an extraordinary amount of time to converge this could point to a degenerate solution.

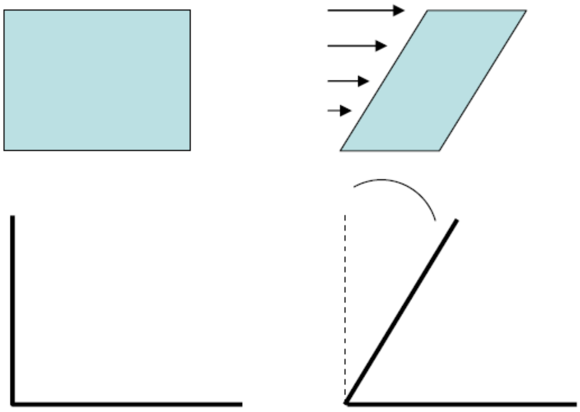


Figure 9.8: Illustration of the shear operator (Harshmann,2004).

The discussion on degeneracy from a modelling perspective makes the concept relatively simple to understand but the focus is on the PARAFAC model as a tensor decomposition technique. What remains is to understand what degeneracy implies for using the PARAFAC model as a decomposition rather than a modelling technique. Degeneracy represents a much more fundamental problem in this regard; it implies that the best rank R solution does not exist. De Silva and Lim (2004) produced seminal work on the topic of best rank R approximations for tensors. In fact Lim (2004) opined that rank is an algebraic concept where as approximate solutions is an analytic concept. The amicable relationship between these two concepts for matrices need not necessarily extend to third order tensors. De Silva and Lim (2004) showed that the best rank R problem has no general solution for higher order tensors. This implies that a best rank R approximation may not exist. For some time researchers clung to the hope that tensors that exhibited this pathological behaviour comprised a small number, but De Silva and Lim (2004) showed that the set of tensors that fail to have a best rank R approximation has positive volume, which means that there is a non-zero probability that a tensor chosen at random will fail to have a best rank R approximation. Furthermore, regardless of the choice of norm used in the approximation, Frobenius or otherwise, the problem still fails to have a solution in general (De Silva and Lim, 2004). A tensor $\mathcal{X}$ is deemed *degenerate* if it can be approximated arbitrarily well by a factorisation of lower rank (Kolda and Bader, 2008). Figure 9.9 illustrates this notion by showing the problem of approximating a rank three tensor $\mathcal{Y}$ with a tensor of rank two. A sequence of rank two tensors $\mathcal{X}^k$ are shown to give increasingly better approximations to the tensor $\mathcal{Y}$ with the best estimate necessarily on the border of the space of rank two and rank three tensors. The problem lies in the fact that the space of rank two tensors is not closed which implies that a sequence of rank two tensors could converge to a tensor of rank other than two. In Figure 9.9, the sequence converges to a tensor of rank three. This is precisely why the definition of degeneracy speaks to a tensor $\mathcal{X}$ being approximated arbitrarily well by a factorisation of lower rank. There simply is no best rank two tensor to approximate $\mathcal{Y}$ because the sequence of rank two tensors $\{\mathcal{X}^{(k)}\}$ converge to a rank three tensor.

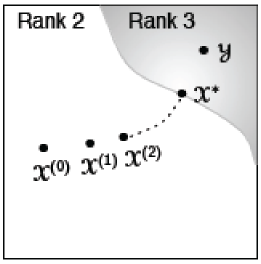


Figure 9.9: Sequence of tensors converging to one of higher rank (Kolda and Bader, 2008).

De Silva and Lim (2004) suggest that when degenerate solutions arise they are not truly providing a best rank R approximation of the given tensor $\mathcal{X}$, but rather providing a solution to a slightly perturbed version of the tensor $\mathcal{X}$. This thought aligns with the fact that the PARAFAC model tries to compromise when faced with data that cannot be modelled with this technique.

One may be tempted to suggest that if a degenerate solution arises then it is satisfactory since the tensor has been decomposed and it serves to give a good approximation to the rank R approximation to the tensor $\mathcal{X}$. The problem with this reasoning is that if degeneracy occurs it suggests that the best rank R approximation to a tensor $\mathcal{X}$ does not exist; it is not possible to approximate the best rank R tensor decomposition to the tensor $\mathcal{X}$ precisely because it does not exist. This is why these authors refer to the problem of determining the best rank R solution as being "ill posed". Recall that a decomposition technique such as the PARAFAC model was considered to be the best candidate for producing a triplot. In fact, Araújo (2009) used the PARAFAC model to produce such a plot that afforded the means to explore interactions between all three modes in a single display. The contention here is that the problem of degeneracy is a serious one and given that it is not a rare or pathological phenomenon, the PARAFAC model does not serve well as a tool for constructing an exploratory plot. The ideal would be a method that can be applied generally without having to be concerned with problems like degeneracy.

Interestingly the problem of degeneracy can be remedied by applying an orthogonality constraint to a component matrix in one of the modes. Harshmann and Lundy (1984) suggest that this will often suffice to block the degeneracy provided that the correct mode is selected and "there is no string internal characteristic of the data promoting highly correlated factors". Furthermore, the data should determine in which mode the orthogonality constraint is applied. This is because it makes most sense to apply the constraint which forces factors to be uncorrelated to the mode in which this assertion has some intuitive appeal. It should be fairly obvious why an orthogonality constraint prevents degeneracy from occurring since it implies that factors must be uncorrelated. It becomes clear that even the task of choosing where to apply the orthogonality constraint is not a simple matter. The contention here is that the PARAFAC decomposition is thus not ideal for constructing an exploratory plot. There is hope yet and it comes in the form of a decomposition technique that preserves the outer product decomposition property of the PARAFAC model together with the orthogonality property of the HOSVD. This technique is referred to as Tensor Singular Value Decomposition (TSVD).

9.7 Tensor SVD

It is simplest to begin the exploration of this decomposition technique with a definition.

Definition 9.7.1. *(Chen and Saad, 2009) A tensor $\mathcal{X} \in \Re^{d_1\times\ d_2\times d_3}$ admits a tensor SVD if it can be written in the form*

$$\mathcal{X} = \sum_{r=1}^{R} \sigma_r \boldsymbol{u}_r^{(1)} \circ \boldsymbol{u}_r^{(2)} \circ \boldsymbol{u}_r^{(3)}, \tag{9.22}$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_R > 0$ and $< \boldsymbol{u}_j^{(n)}, \boldsymbol{u}_k^{(n)} >= \delta_{ij}$ for $n = 1, 2, 3$. δ_{ij} is the Kronecker delta, σ_r's are the singular values and $\boldsymbol{u}_r^{(n)}$ for $r = 1, 2, \ldots, R$ are the n-mode singular vectors.

An equivalent representation, also taken from Chen and Saad (2009), is given by

$$\mathcal{X} = \mathcal{D} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \boldsymbol{U}^{(3)}, \tag{9.23}$$

where $\mathcal{D} \in \Re^{R\times R\times R}$ is the diagonal core tensor with $\mathcal{D}_{ii\ldots i} = \sigma_i$ and

$$\boldsymbol{U}^{(n)} = \left(\boldsymbol{u}_1^{(n)}, \boldsymbol{u}_2^{(n)}, \ldots, \boldsymbol{u}_R^{(n)}\right) \in \Re^{d_n\times R}, \tag{9.24}$$

are orthogonal matrices for $n = 1, 2, 3$. It is interesting to note the similarities between (9.23) and the definition of HOSVD given in (9.10). The fundamental difference between the two decompositions is that $\mathcal{D}$ is a diagonal tensor in the TSVD instance where as it is often a full tensor in the case of HOSVD. There is an intimate link between the HOSVD and the TSVD in that a tensor $\mathcal{X}$ will admit a TSVD if and only if the core tensor arising from a HOSVD is diagonalisable but in general this cannot be done (Bro, 2008). A tensor may thus fail to have a decomposition as defined in (9.23). This would seem to put paid to the notion that the TSVD can in fact provide the most general means to construct a triplot but exploring the implications of the TSVD definition will prove otherwise. The TSVD definition refers to a tensor of rank R being expressed as in (9.23). It can be shown that if a tensor $\mathcal{X}$ is decomposed as in (9.23) and the vectors comprising the outer-product terms are linearly independent, then the tensor $\mathcal{X}$ will have rank R. The importance of this assertion is that a TSVD might not exist to fully decompose a tensor into the sum of R outerproducts where R is the rank of the tensor but this does not make a statement regarding the ability to use TSVD to find a lower rank approximation to the tensor $\mathcal{X}$. Here the problem of interest is to minimise

$$\| \mathcal{X} - \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i^{(1)} \circ \boldsymbol{u}_i^{(2)} \circ \boldsymbol{u}_i^{(3)} \|^2, \tag{9.25}$$

subject to the constraint that $<\boldsymbol{u}_j^{(n)}, \boldsymbol{u}_k^{(n)}>= \delta_{ij}$ for $n = 1, 2, 3$. It has already been established that this problem might not have a solution if $r = R$, the rank of the tensor $\mathcal{X}$. It is thus necessary to determine which values of r result in a solution. Chen and Saad (2009) showed that the minimisation problem will always have a solution for any $\mathcal{X} \in \Re^{d_1 \times d_2 \times d_3}$ and any $r \leq min\{d_1, d_2, \ldots, d_n\}$. The proof will not be included here but some mathematical detail is required in order to explain the mechanics of the estimating algorithm. Chen and Saad (2009) showed that (9.25) can be written as

$$\parallel \mathcal{X} - \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i^{(1)} \circ \boldsymbol{u}_i^{(2)} \circ \boldsymbol{u}_i^{(3)} \parallel^2 = \parallel \mathcal{X} \parallel^2 - \sum_{i=1}^{r} \sigma_i^2. \tag{9.26}$$

The implication of this is that minimising (9.25) can be based on maximising the quantity $\sum_{i=1}^{r} \sigma_i^2$ subject to the same orthogonality constraints specified earlier. On this basis, Chen and Saad (2009) define the Langragian as

$$L = \sum_{i=1}^{r} \sigma_i^2 - \sum_{j,k=1}^{R} \sum_{n=1}^{3} \mu_{j,k}^n (< \boldsymbol{u}_j^{(n)}, \boldsymbol{u}_k^{(n)} > -\delta_{jk}), \tag{9.27}$$

where

$$\sigma_i = \mathcal{X} \times_1 (\boldsymbol{u}_i^{(1)})' \times_2 (\boldsymbol{u}_i^{(2)})' \times_3 (\boldsymbol{u}_i^{(3)})', \tag{9.28}$$

and $\mu_{j,k}^n$ terms represent the langrangian multipliers. Remaining ever cognisant of the fact that the maximisation is done relative to $\boldsymbol{u}_i^{(n)}$, the partial derivative of L with respect to $\boldsymbol{u}_i^{(n)}$ is given as

$$\frac{\partial L}{\partial \boldsymbol{u}_i^{(n)}} = 2\sigma_i \boldsymbol{v}_i^{(n)} - \sum_{j=1}^{r} \mu_{j,i}^n \boldsymbol{u}_i^{(n)} - \sum_{k=1}^{r} \mu_{k,i}^n \boldsymbol{u}_i^{(n)}, \tag{9.29}$$

for any n and i. The definition of $\boldsymbol{v}_i^{(n)}$ is best explained by way of example. The term $\boldsymbol{v}_i^{(1)}$ is equal to the unfolding of the tensor $\mathcal{X} \times_2 (\boldsymbol{u}_i^{(2)})' \times_3 (\boldsymbol{u}_i^{(3)})'$ in the first mode with dimensions $d_1 \times 1 \times 1$. Similarly $\boldsymbol{v}_i^{(2)}$ is equal to the unfolding of the tensor $\mathcal{X} \times_1 (\boldsymbol{u}_i^{(1)})' \times_3 (\boldsymbol{u}_i^{(3)})'$ in the second mode with $1 \times d_2 \times 1$. The unfolding $\boldsymbol{v}_i^n$ is also the partial derivative of σ_i with respect to $\boldsymbol{u}_i^{(n)}$ and a simple application of the product rule yields the first term in (9.29). Setting the partial derivative equal to zero and collecting all terms referring to the same n in matrix form yields

$$\left[\boldsymbol{v}_1^n \ldots \boldsymbol{v}_r^n\right] \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{pmatrix} = \left[\boldsymbol{u}_1^{(n)} \ldots \boldsymbol{u}_r^{(n)}\right] \begin{pmatrix} \frac{\mu_{1,1}^{(n)} + \mu_{1,1}^{(n)}}{2} & \cdots & \frac{\mu_{1,r}^n + \mu_{r,1}^n}{2} \\ \vdots & \ddots & \vdots \\ \frac{\mu_{r,1}^n + \mu_{1,r}^n}{2} & \cdots & \frac{\mu_{r,r}^n + \mu_{r,r}^n}{2} \end{pmatrix}, \tag{9.30}$$

for $n = 1, 2, 3$. This expression can be represented in a neater fashion as

$$\boldsymbol{V}^{(n)}\boldsymbol{\Sigma} = \boldsymbol{U}^{(n)}\boldsymbol{M}^{(n)}, n = 1, 2, 3 \tag{9.31}$$

where the matrices $\boldsymbol{V}^{(n)}, \boldsymbol{\Sigma}, \boldsymbol{U}^{(n)}$ and $\boldsymbol{M}^{(n)}$ correspond to those in (9.30) respectively and note that $\boldsymbol{M}^{(n)}$ is symmetric. An estimating algorithm is required to estimate matrices $\boldsymbol{U}^{(n)}$ and $\boldsymbol{M}^{(n)}$ that satisfy the system of equations in (9.31). Note that by virtue of the fact that $\boldsymbol{U}^{(n)}$ is orthogonal and $\boldsymbol{M}^{(n)}$ is a symmetric matrix, the right hand side of (9.31) is thus the polar decomposition of $\boldsymbol{V}^{(n)}\boldsymbol{\Sigma}$ and this is at the heart of the estimating algorithm. Suppose that a matrix $\boldsymbol{A}$ has SVD $\boldsymbol{WDY}^{'}$, then the polar decomposition of $\boldsymbol{A}$ can be defined as the product of the orthogonal matrix $\boldsymbol{WY}^{'}$ and positive semi-definite symmetric matrix $\boldsymbol{YDY}^{'}$. The significance of this is that it affords the means to calculate the polar decomposition from the SVD of $\boldsymbol{V}^{(n)}\boldsymbol{\Sigma}$. It is now possible to detail the estimating algorithm as presented in Chen and Saad (2009).

1. Initialise $\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}$ and $\boldsymbol{U}^{(3)}$ which is often based on truncated HOSVD.
2. For n=1,2,3 do the following:
3. Compute $\boldsymbol{V}^{(n)}$.
4. Compute $\boldsymbol{\Sigma}$.
5. Compute $[\boldsymbol{Q}^{(n)}, \boldsymbol{H}^{(n)}] \leftarrow$ polar decomposition($\boldsymbol{V}^{(n)}\boldsymbol{\Sigma}$).
6. Update $\boldsymbol{U}^{(n)} \leftarrow \boldsymbol{Q}^{(n)}$.
7. Check for convergence based on $\sum_{i=1}^{r} \sigma_i^2$. If convergence has not been obtained return to step 2.

The algorithm described is based on an alternating procedure where all but one parameter, $\boldsymbol{U}^{(n)}$, is fixed during each step. Furthermore, Chen and Saad (2009) could not prove that the algorithm converged globally so it may suffer from the same shortcoming as alternating least squares algorithms in general. With the estimating algorithm having been discussed attention turns to the triplot.

9.7.1 Triplot construction

Now that a suitable decomposition technique has been found, thought must be given to constructing a plot that simultaneously represents each of the three modes comprising the tensor $\mathcal{X}$. This construction is based on the work by Araújo (2009), the only modification being the choice of tensor decomposition technique employed. The first matter to attend to is the choice of value for r in (9.25). In order for the approximation to yield a result, $r \leq min\{d_1, d_2, d_3\}$, the dimensions in each of the modes. Choosing $r = 2$ allows the matrices $\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}$ and $\boldsymbol{U}^{(3)}$ to be represented in a two dimensional plot. Furthermore it is not likely that any three mode data array will have less than two entries in any one of the modes implying that the TSVD approximation can always be found. Is is thus assumed that $r = 2$ for the remainder of this chapter. The TSVD decomposition can be

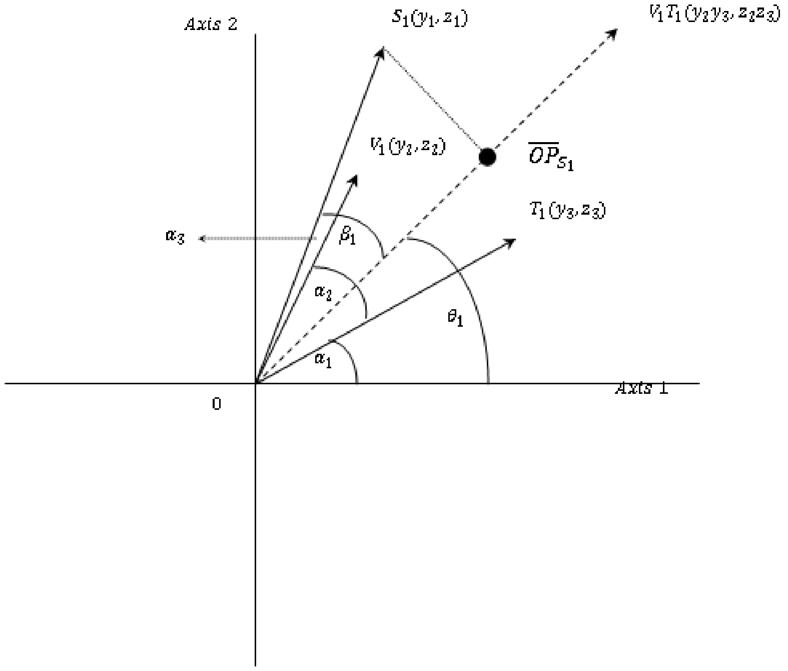


Figure 9.10: Representation of the triplot construction.

represented as

$$\hat{x}_{ijk} = \sum_{n=1}^{2} \sigma_n \boldsymbol{u}_{in}^{(1)} \boldsymbol{u}_{jn}^{(2)} \boldsymbol{u}_{kn}^{(3)}. \tag{9.32}$$

As an example, consider the expression for $\hat{x}_{111}$ which is given by

$$\begin{aligned} \hat{x}_{111} &= (\sigma_1 \boldsymbol{u}_{11}^{(1)}) \boldsymbol{u}_{11}^{(2)} \boldsymbol{u}_{11}^{(3)} + (\sigma_2 \boldsymbol{u}_{12}^{(1)}) \boldsymbol{u}_{12}^{(2)} \boldsymbol{u}_{12}^{(3)} \\ &= y_1 y_2 y_3 + z_1 z_2 z_3. \end{aligned} \tag{9.33}$$

Notice that for the purpose of explaining the construction process, the σ_i terms have been grouped with the elements of the first matrix $\boldsymbol{U}^{(1)}$. The effect of these values will be explored later but for the moment it suffices to treat them in an arbitrary fashion just for the purpose of explaining the construction of a triplot as suggested by Araújo (2009). With the aid of Figure 9.10 it is possible to interpret this representation in a graphical sense. S_1 represents the first row of the matrix $\boldsymbol{U}^{(1)}$ scaled by the singular values plotted relative to Cartesian axes, so labelled because the first mode often refers to subjects. V_1 and T_1 represent the rows of $\boldsymbol{U}^{(2)}$ and $\boldsymbol{U}^{(3)}$ plotted relative to Cartesian axes, labelled as they are due to the fact that variables and time often comprise the second and third modes respectively. V_1T_1 is the result of taking the product of corresponding elements of the vectors $\overline{OV}_1$ and $\overline{OT}_1$. Consider a somewhat different representation of each of the elements comprising the expansion in (9.33).

$$
\begin{aligned}
y_1 &= \overline{OS}_1 cos(\beta_1+\theta_1) & z_1 &= \overline{OS}_1 sin(\beta_1+\theta_1) && (9.34)\\
y_2 &= \overline{OV}_1 cos(\alpha_1+\alpha_2) & z_2 &= \overline{OV}_1 sin(\alpha_1+\alpha_2) && (9.35)\\
y_3 &= \overline{OT}_1 cos(\alpha_1) & z_3 &= \overline{OT}_1 cos(\alpha_1) && (9.36)\\
y_2y_3 = u_1 &= \overline{OV_1T_1} cos(\theta_1) & z_2z_3 = v_1 &= \overline{OV_1T_1} sin(\theta_1). && (9.37)\\
&&&&& (9.38)
\end{aligned}
$$

With this notation (9.33) can be written as

$$
\begin{aligned}
\hat{x}_{111} &= y_1u_1 + z_1v_1\\
&= \overline{OS}_1 cos(\beta_1+\theta_1)\overline{OV_1T_1} cos(\theta_1) + \overline{OS}_1 sin(\beta_1+\theta_1)\overline{OV_1T_1} sin(\theta_1))\\
&= \overline{OS}_1\overline{OV_1T_1} cos(\beta_1+\theta_1-\theta_1)\\
&= \overline{OS}_1\overline{OV_1T_1} cos(\beta_1)\\
&= \overline{OP}_{S_1}\overline{OV_1T_1},
\end{aligned}
\tag{9.39}
$$

where $\overline{OP}_{S_1} = \overline{OS}_1 cos(\beta_1)$ represents the projection of $\overline{OS}_1$ onto $\overline{OV_1T_1}$. This representation is based on projecting the vector representing subject one, $\overline{OS}_1$ onto the vector that essentially represents variable one at occasion one. This is not the only possibility however since it is also possible to project the vector representing variable one ($\overline{OV}_1$) onto the vector that represents subject one at occasion one ($\overline{OS_1T_1}$) or to project the vector representing occasion one ($\overline{OT}_1$) onto the vector that represents subject one on variable one ($\overline{OS_1V_1}$). Figure 9.11 is a representation of the first instance mentioned. In this way the plot affords the means to examine the interactions between all three modes. Generally, following the same process as in (9.39) it is a simple matter to show that

$$\begin{aligned} \hat{x}_{ijk} &= \overline{OS}_i cos(\beta_{S_i,j*k})\overline{OV_jT_k} = \overline{OP}_{S_i}\overline{OV_jT_k} \\ &= \overline{OV}_j cos(\beta_{V_j,i*k})\overline{OS_iT_k} = \overline{OP}_{V_i}\overline{OS_iT_k} \\ &= \overline{OT}_k cos(\beta_{T_k,i*j})\overline{OS_iV_j} = \overline{OP}_{T_i}\overline{OS_iV_j}, \end{aligned} \tag{9.40}$$

and $\beta_{S_i,j*k}$ refers to the angle between the vectors $\overline{OS_i}$ and $\overline{OV_jT_k}$. $\beta_{V_j,i*k}$ and $\beta_{T_k,i*j}$ are interpreted in a similar fashion. These angles carry information regarding the sign of the value $\hat{x}_{ijk}$ in that all the vector lengths are positive but the cosine term can be positive, negative or zero. If these angles are acute then $\hat{x}_{ijk}$ will be positive since the cosine term will be positive. An angle of 90 degrees yields $\hat{x}_{ijk}$ equal to zero and an angle between 90 degrees and 180 degrees yields a negative value for $\hat{x}_{ijk}$. Furthermore, the relative magnitude of the elements $\hat{x}_{ijk}$ can be compared by considering the projection vectors since

$$\frac{\hat{x}_{ijk}}{\overline{OV_jT_k}} = \overline{OP}_{S_i}. \tag{9.41}$$

It should be clear that this plot is primarily based on examining inner products. Beyond that, it cannot be thought of as an extension of biplot methodology since it is being plotted relative to Cartesian axes as opposed to making use of the best fitting plane construction. Kiers (2000a) warns against using Cartesian axes because it can result in the data being misrepresented in a graphical context. Specifically, Euclidian distance between sample points for example may not be accurately represented. An example taken from Kiers (2000a) will serve to clarify this argument. Consider two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ defined as

$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 \\ 1 & 0.5 \end{pmatrix} \text{ and } \boldsymbol{B} = \begin{pmatrix} 1 & 0.5 & 0.5 & 0 \\ 0 & 0.5 & -0.5 & 1 \end{pmatrix}', \tag{9.42}$$

so that

$$\hat{\boldsymbol{X}} = \boldsymbol{A}\boldsymbol{B}' = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0.75 & 0.25 & 0.5 \end{pmatrix}. \tag{9.43}$$

$\hat{\boldsymbol{X}}$ is a 2×4 matrix and can thus be represented in $\Re^2$ by plotting the columns or in $\Re^4$ by plotting the rows. Consider representing the data in $\Re^2$. The middle plot of Figure 9.11 illustrates this representation. The left most plot illustrates what results from plotting the columns of $\boldsymbol{B}$. Notice that although both these plots suggest that observations A, B, C and D are on a line they give different information regarding the proximity of C from the remaining observations. The middle plot which is the actual representation in $\Re^2$ suggests that C is farthest from A where as the left most plot is distorted in this regard. Plotting the rows of $\boldsymbol{B}$ implies that the columns of $\boldsymbol{A}$ serves as a basis so that the axes used when plotting the rows of $\boldsymbol{B}$ are in fact the columns of $\boldsymbol{A}$ and this is not column-wise orthonormal. It is possible to determine a transformation matrix $\boldsymbol{T}$ using an orthonormalisation procedure such as the Gram-Schmidt orthonormalisation to $\boldsymbol{A}$. Defining $\tilde{\boldsymbol{A}} = \boldsymbol{AT}$ and $\tilde{\boldsymbol{B}} = \boldsymbol{B}(\boldsymbol{T}')^{-1}$ then plotting the columns of $\tilde{\boldsymbol{B}}$ yields the right most plot in Figure 9.11. The orientation of the points in this plot is clearly a rotated version of that seen in the middle plot. It is thus undistorted. This illustrates the argument made by Kiers (2000a) that plotting relative to an orthonormal basis results in an undistorted plot.

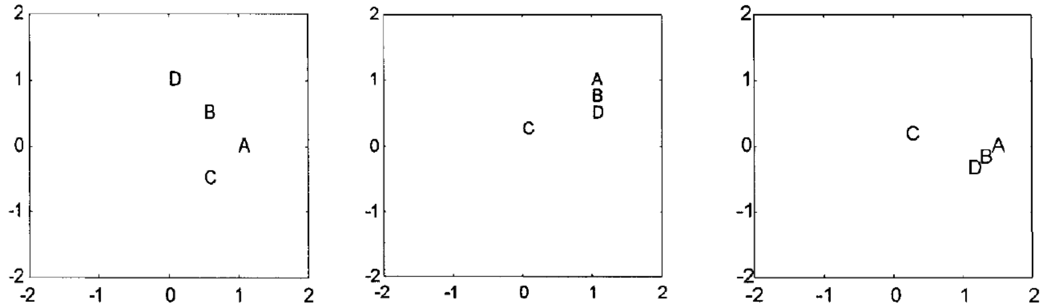


Figure 9.11: Plots (from left to right) of rows of $\boldsymbol{B}, \hat{\boldsymbol{X}}$ and $\tilde{\boldsymbol{B}}$ (Kiers, 2000a).

This is a fair point and a serious consideration, however the geometry of three mode data is very complex. Biplot methodology is so successful because visualising the construction process geometrically is not difficult. It can be argued that is precisely for this reason that Kiers (2000a) speaks to the fact that most graphical techniques for three mode data rely on the data being matricised after analysis by some three mode technique; it simplifies the geometry. Merely trying to visualise the tensor space is very difficult if not impossible. It makes trying to extend the concept of finding a best fitting plane exceedingly difficult. This is why the construction discussed relies on plotting relative to Cartesian axes, a method that Kiers (2000a) warns against. When modes are combined to create axes, these do not carry the same meaning as they would in a PCA biplot because they do not rep-

	Sample 12	Sample 21	Sample 23	Sample 27	Sample 28
Sample 16	12626.60	8580.71	8758.430	9425.859	8852.185

Table 9.2: Extract from the distance matrix for $\boldsymbol{X}_{(1)}$.

resent projection of the variable axes in the higher dimensional space. It is still informative to explore the triplot methodology put forward by Araújo (2009) even though Euclidean distances will likely not be well represented. Moreover, the objective is to graphically represent all three modes of the data and so none of the component matrices act as a basis relative to which the remaining component matrices are plotted. The triplot relies on simply plotting each of the component matrices relative to Cartesian axes. The contribution of this dissertation is to apply the triplot methodology using an altogether different tensor decomposition technique and not the PARAFAC decomposition that was originally used.

9.7.2 Application of Triplot Methodology

Due to the fact that the triplot is not well understood, simulated data is used together with the Mansoor data in order to see what the triplot conveys and how this relates to the data. In order to ascertain the effects of data preprocessing, triplots are produced for the unprocessed data, centered data as well as the centered and scaled data in each of the three instances considered.

Recall that the effect of the σ_r terms needs to be explored. The most logical effect that these terms would have is to ensure that the Euclidean distance between sample points, variables or occasions will be well displayed. This is because these terms play a role similar to that of the singular value in two-mode PCA. In order to see whether this is the case, the Mansoor data is used. Recall that the component matrices $\boldsymbol{U}_{(1)}$, $\boldsymbol{U}_{(2)}$ and $\boldsymbol{U}_{(3)}$ are associated with sample, variable and occasion modes respectively. To see whether the σ_r terms where $r = 1, 2$ affect how well the Euclidean distances between sample points are displayed, the rows of $\boldsymbol{U}_{(1)}\boldsymbol{\Sigma}$ are plotted and compared to the plot comprising the rows of $\boldsymbol{U}_{(1)}$. Since the initial estimate for $\boldsymbol{U}_{(1)}$ is based on the SVD of $\boldsymbol{X}_{(1)}$ it is reasonable to use the matrix of Euclidean distances calculated from $\boldsymbol{X}_{(1)}$ to see how well the plot represents the distances between observations. The resulting distance matrix is of dimension 29×29 and thus only select observations will be used to assess the plot. Figure 9.12 illustrates the resulting plot and the panel on the right is considered

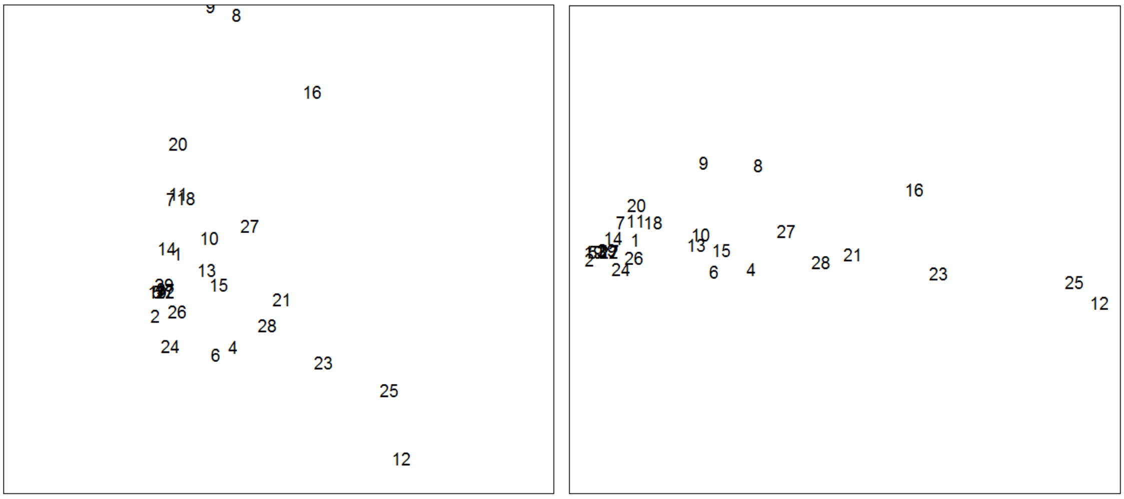


Figure 9.12: Plot of the rows of $\boldsymbol{U}_{(1)}$ and $\boldsymbol{U}_{(1)}\boldsymbol{\Sigma}$ respectively.

first. Observation 16 will be considered in relation to observations 27, 21, 28, 23 and 12 to ascertain whether Euclidean distances are accurately reflected. Table 9.2 shows the distances of observation 16 from observations 12, 21, 23 and 28. The relationships seen in Table 9.2 are accurately reflected in the panel on the right hand side of Figure 9.12. Of those observations considered, observation 21 is shown to be the closest to observation 16 in the plot and this agrees with what is seen in Table 9.2. This was seen to be the case when the distance matrix was studied at large. The panel on the left of Figure 9.12 shows that the Euclidean distances are indeed distorted. Observation 27 is closer in proximity to observation 16 than observation 21 but this is not the case when the distances are considered. The conclusion is thus that the effect of $\boldsymbol{\Sigma}$ is to ensure that the Euclidean distances are well represented regardless of whether it is applied to the sample, variable or occasion mode. It thus seems to follow that at least one mode can be well represented in the triplot display however there is a problem that shows itself in each of the data sets considered.

The triplot display requires that all component matrices be represented on a single plot. If for example $\boldsymbol{\Sigma}$ is applied to $\boldsymbol{U}_{(1)}$ then the triplot display be constructed by plotting the rows of $\boldsymbol{U}_{(1)}\boldsymbol{\Sigma}$, $\boldsymbol{U}_{(2)}$ and $\boldsymbol{U}_{(3)}$ relative to Cartesian axes. Figure 9.13 is an illustration of the resulting triplot and the problem should be immediately clear. The points representing the variable and occasion modes are all located around the origin whilst the sample points,

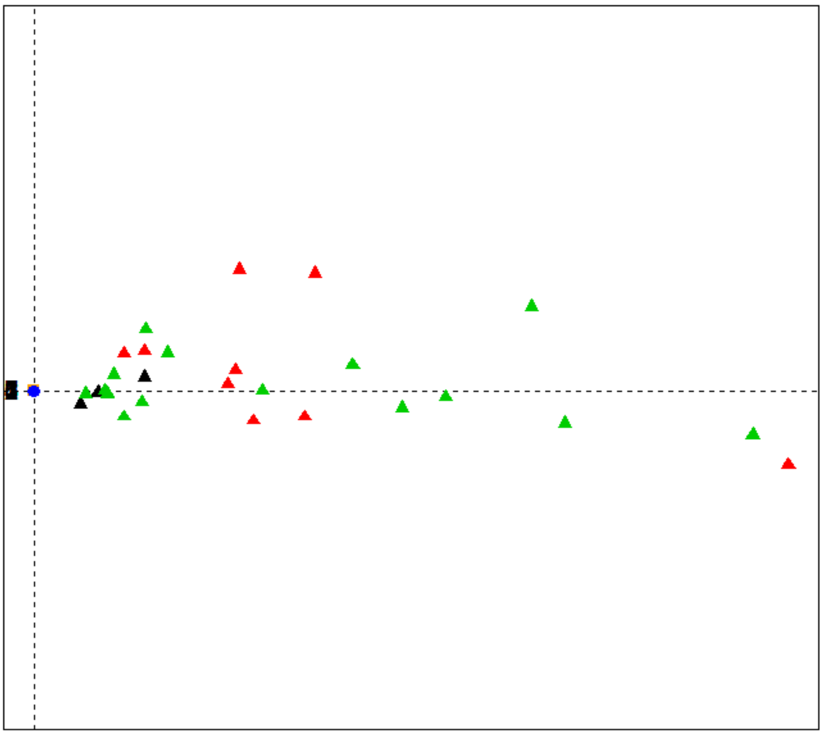


Figure 9.13: Plot of$\boldsymbol{U}_{(1)}\boldsymbol{\Sigma}$, $\boldsymbol{U}_{(2)}$ and $\boldsymbol{U}_{(3)}$ relative to Cartesian axes.

represented by the triangles, is well represented. The triplot thus conveys no useful information about the variable and occasion modes. The problem is less pronounced when the data are scaled but it is still prominent. Although this was illustrated by scaling the subject mode, it applies regardless of which mode is scaled. The implication is that having Euclidean distances well represented in any one mode renders the triplot impotent. The solution to this problem lies in simply scaling the component matrices in each of the modes equally. This means that each component matrix is post-multiplied by $\boldsymbol{\Sigma}^{\frac{1}{3}}$. This is termed *Sigma scaling* by Gower *et al.* (2011) in the two way context. This is how the triplot is constructed here and it is studied to determine whether it conveys any information regarding the structure of the data.

The first application of the triplot methodology is to the simulated data used in Chapter 4. Recall that this data comprised 45 observations classified into three groups with measurements on three variables taken over four occasions. Furthermore it was simulated to show marked separation between groups at occasions 2 and 4. Figures 9.14, 9.15 and 9.16 represent the triplots for the

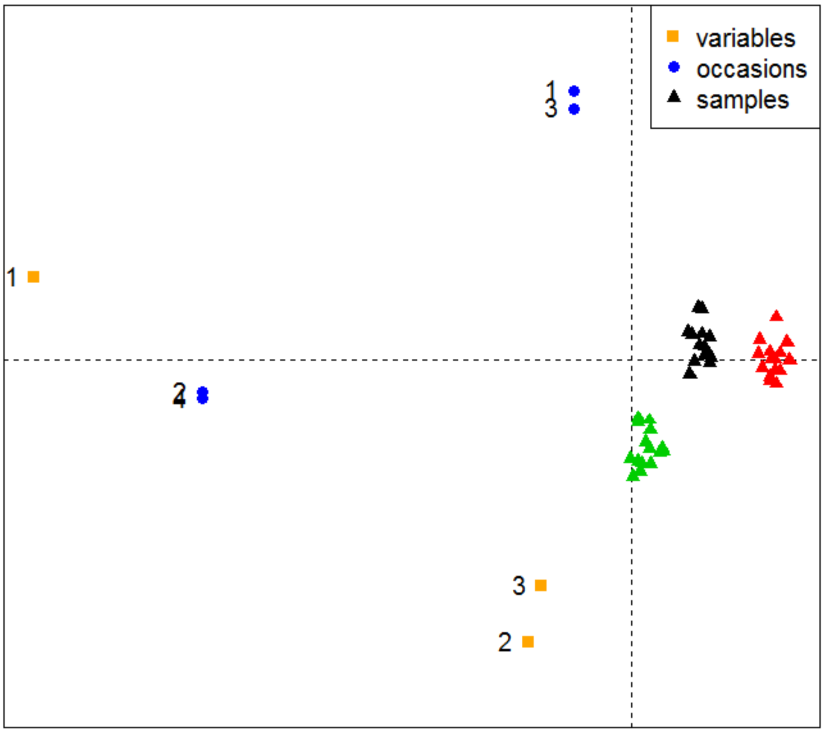


Figure 9.14: Triplot for the unprocessed simulated data set one.

unprocessed, centered as well as the centered and scaled data respectively. The legend in Figure 9.14 applies throughout and is thus only shown once. In order to explain the interpretation that Araújo (2009) puts on these triplots, Figure 9.14 will be used. The first consideration is the distance of the points from the origin which is represented by the intersection of the dashed lines. Closeness to the origin implies stability in scores. For example, if the symbol for occasion 1 were very close to the origin and the subject and variable modes were combined to produce axes as explained previously then the projection of the point representing occasion 1 onto any of these axes will result in similar inner products. The farther away a point is from the origin, the more dynamic the scores for that point are expected to be. The next aspect to consider is that of how points cluster. Occasions 1 and 3 form a group and the same can be said for Occasions 2 and 4. Araújo (2009) suggests that this implies that those occasions that are grouped share similar characteristics without providing clarity on what is meant by *characteristics*. It is argued that similarity with respect to characteristics can be measured by considering the similarity between variable means and variances at the occasions in

	Variable 1	Variable 2	Variable 3
Occasion 1	0.39	0.40	0.28
Occasion 2	2.39	0.99	0.95
Occasion 3	0.36	0.43	0.31
Occasion 4	2.36	0.99	0.92

Table 9.3: Means for Simulated data set one.

	Variable 1	Variable 2	Variable 3
Occasion 1	0.30	0.37	0.26
Occasion 2	4.16	0.67	0.82
Occasion 3	0.38	0.45	0.39
Occasion 4	4.55	0.86	1.05

Table 9.4: Variances for Simulated data set one.

question. Tables 9.3 and 9.4 contain the means and variances respectively for each variable at each occasion. Studying Table 9.3 reveals that the means at occasions 1 and 3 are very similar and those at occasions 2 and 4 are very similar. The same phenomenon can be seen when comparing the variances in Table 9.4. The triplot thus seems to capture this aspect of the data.

Araújo (2009) states that the angles between rows within a particular mode provides a sense of the strength of association between the elements comprising the mode in question. For example, the angles between the vectors drawn from the origin to the variable markers will provide an approximation to the correlation between variables. Araújo (2009) states this without making clear precisely which correlations are being approximated. Since the data are three mode in nature, the correlations between variables change over time but there are only three markers in Figure 9.14 to represent the variables. Given that the initial estimate for the variable mode component matrix $\boldsymbol{U}_{(2)}$

	Variable 1	Variable 2	Variable 3
Variable 1	1	-0.18	-0.17
Variable 2	-0.18	1	0.75
Variable 3	-0.17	0.75	1

Table 9.5: Correlations between the rows of $\boldsymbol{X}_{(2)}$.

is taken from the SVD of the $P \times NK$ matrix $\boldsymbol{X}_{(2)}$, it was thought that the correlations being approximated are those that exist between the rows of the matrix $\boldsymbol{X}_{(2)}$. Table 9.5 contains these correlations and it is clear that variables 2 and 3 show a strong correlation with each other where as variable 1 does not show this quality. Studying Figure 9.14 reveals that this is indeed represented on the triplot with variables 2 and 3 showing a strong relationship. The same observation was made when considering the correlations between the rows of the $K \times PN$ matrix $\boldsymbol{X}_{(3)}$ which represents the correlations between occasions. This lends credence to the notion that the correlations represented are in fact those calculated by considering the rows of the relevant unfolding.

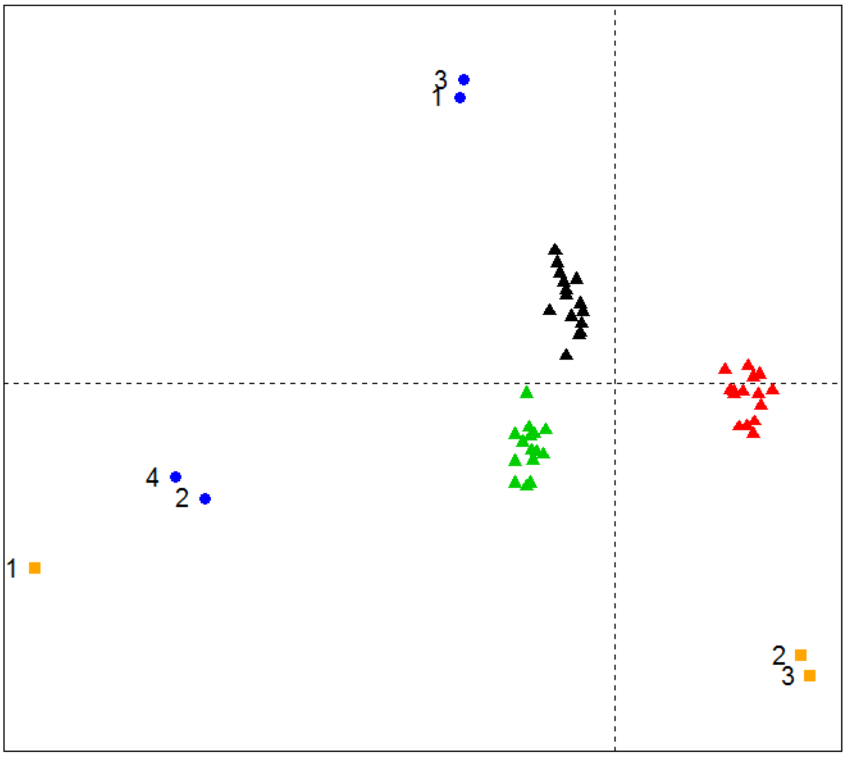


Figure 9.15: Triplot for the centered simulated data set one.

It is interesting to note that the separation between the groups comprising the subjects is in fact represented on the triplot. In Chapter 4 it was seen that group 2, represented by the red symbols, is more separated from the other two groups but Figure 9.14 suggests that group 3, represented by the green symbols, is more separated from the other two groups. In Figure 9.15, the triplot for the fibre centered simulated data, this is rectified and the grouping inherent in the data is displayed. Figures 9.14 and 9.15 are similar but for this change in the separation between the groups comprising the subjects. Figure 9.16, the triplot for the centered and scaled simulated data also looks similar to the previous triplots but this is because the variance matrix used in the simulation was diagonal with 0.1 on the diagonal. This implies similar variation across variables and so scaling would not have any significant effect. In effect the triplot was able to display the separation between the subjects, indicate which occasions and variables share similar characteristics as well as provide a sense of the correlations between elements within a mode. The simulated data suggests that the triplot can be a useful display. The second application of the triplot methodology is to a second simulated data set with characteristics similar to the previous simulated data set but the variance

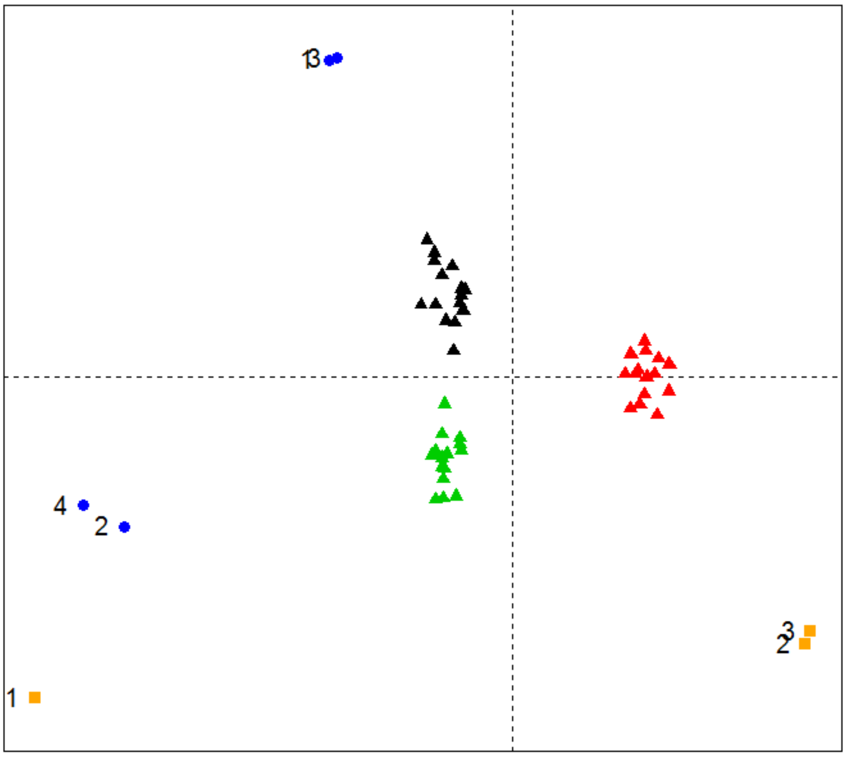


Figure 9.16: Triplot for the centered and scaled simulated data set one.

matrix defined as

$$\begin{pmatrix} 3.0 & 2.2 & 0.7 \\ 2.2 & 2.0 & 0 \\ 0.7 & 0 & 1 \end{pmatrix}. \tag{9.44}$$

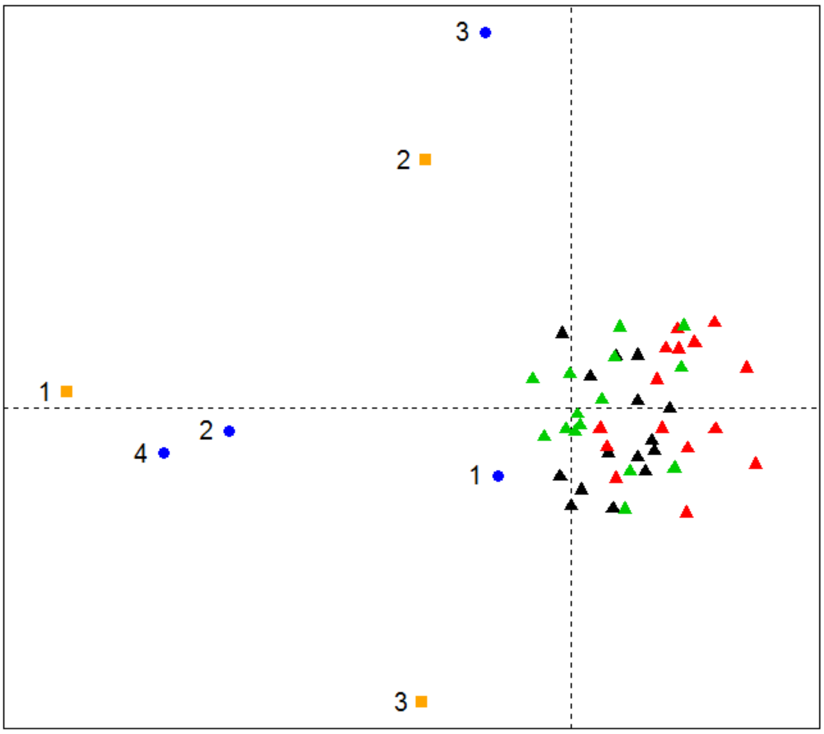


Figure 9.17: Triplot for the unprocessed simulated data set two.

Figure 9.17 represents the triplot for the unprocessed simulated data. Tables 9.6 and 9.7 provide the means and variances respectively for the second simulated data. The most striking feature in Figure 9.17 is the lack of separation between the subjects. The plot also suggests all four occasions are quite different with respect to characteristics but that occasions 2 and 4 share slight similarities. Table 9.6 reveals that the means at occasions 1 and 3 as well as occasions 2 and 4 are similar but the variances in Table 9.7 are quite different. The distance from the origin of the occasion markers suggests that subject scores were quite variable over time with occasion 1 showing the most stability when compared with the other occasions. This is evidenced by the fact that the variables at occasion 1 have the smallest variances relative to the other occasions. None of the variable markers are clustered and the plot suggests relatively weak associations between the variables. Figure

	Variable 1	Variable 2	Variable 3
Occasion 1	0.64	0.48	0.59
Occasion 2	2.57	1.23	1.03
Occasion 3	0.54	0.34	0.64
Occasion 4	2.25	1.00	0.98

Table 9.6: Means for Simulated data set two.

	Variable 1	Variable 2	Variable 3
Occasion 1	3.06	1.97	1.41
Occasion 2	6.67	3.34	1.18
Occasion 3	5.87	3.91	2.86
Occasion 4	11.44	3.53	2.67

Table 9.7: Variances for Simulated data set two

9.18, the triplot for the centered simulated data tells a somewhat different story. Firstly, the separation of group 2 from the other two groups is clearer and although the plot conveys similar information about the occasion mode it does suggest different correlations between variables. Variables 2 and 3 are revealed to have a strong relationship however the actual correlation between these variables is 0.18. Variables 1 and 2 have a correlation of 0.56 and variables 1 and 3 have a correlation of 0.14. This is not well represented in Figure 9.17 or Figure 9.18. Notice that Figure 9.19, the triplot for the centered and scaled simulated data, not only looks quite different to the previous triplots but represents the correlations between variables relatively well. This suggests that the triplot display is not scale invariant. Although Figure 9.19 represents correlations within modes relatively well, the way in which the occasion markers are represented is not completely accurate. The large variances on the variables at occasion 2 relative to occasion 1, with the exception of variable 3, suggests less stability in scores for this occasion than is suggested by the plot. Furthermore, the separation between subjects is lost. The triplot based on the centered data is the preferred one to get a general overview of the data with the triplot based on centered and scaled data being preferred to convey information about correlations.

The final application will be based on applying the triplot methodology to the Mansoor data. Tables 9.8 and 9.9 provide the mean and variance information for the Mansoor data. It is immediately clear that occasion 1 is considerably different to the other occasions given the significantly larger variances and means. Occasions 3 and 4 show the most similarity in terms of

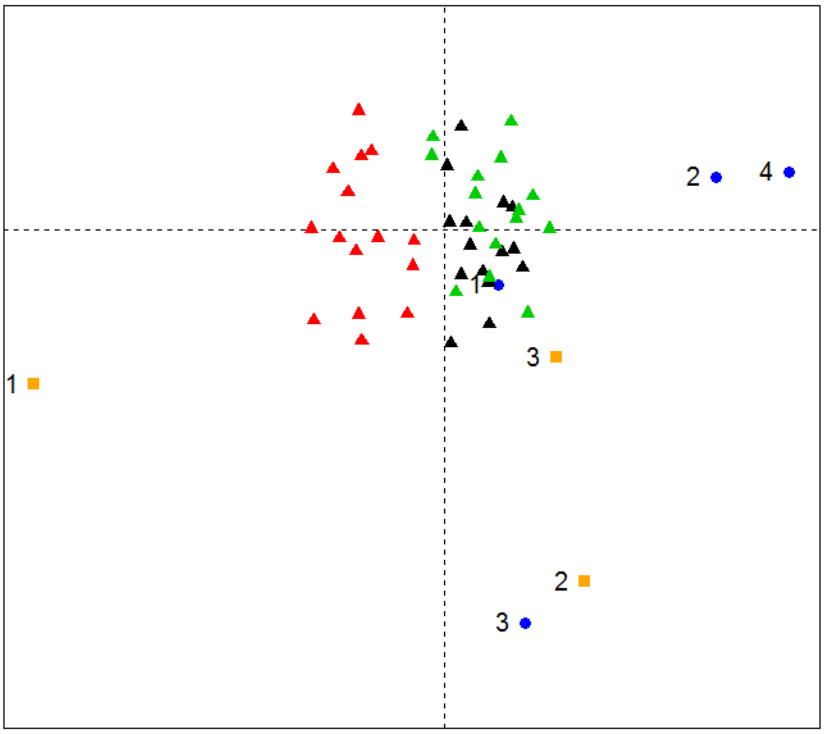


Figure 9.18: Triplot for the centered simulated data set two.

means and variances. It is thus expected that these occasions should appear relatively close to one another on the plot. Given the fact that the variation in the data is quite large, it is expected that occasion markers would not appear close to the origin. Given that the previous triplot applications revealed distributions of the sample values similar to what has been seen in previous Chapters, it is expected that that would be the case for the Mansoor data. Having established what is expected to be seen in a triplot of the Mansoor data attention now turns to Figures 9.20, 9.21 and 9.22. Each of these are the triplots for the unprocessed, centered as well as centered and scaled Mansoor data respectively. All the triplots display what was expected to be seen with respect to occasions and samples. The occasion markers do not appear very close to the origin and occasions 3 and 4 markers appear relatively close together. The marker for occasion 1 is relatively far from the rest. Figure 9.23, the triplot of the scaled Mansoor data, does well to represent the correlations between variables although Figure 9.22 also does relatively well.

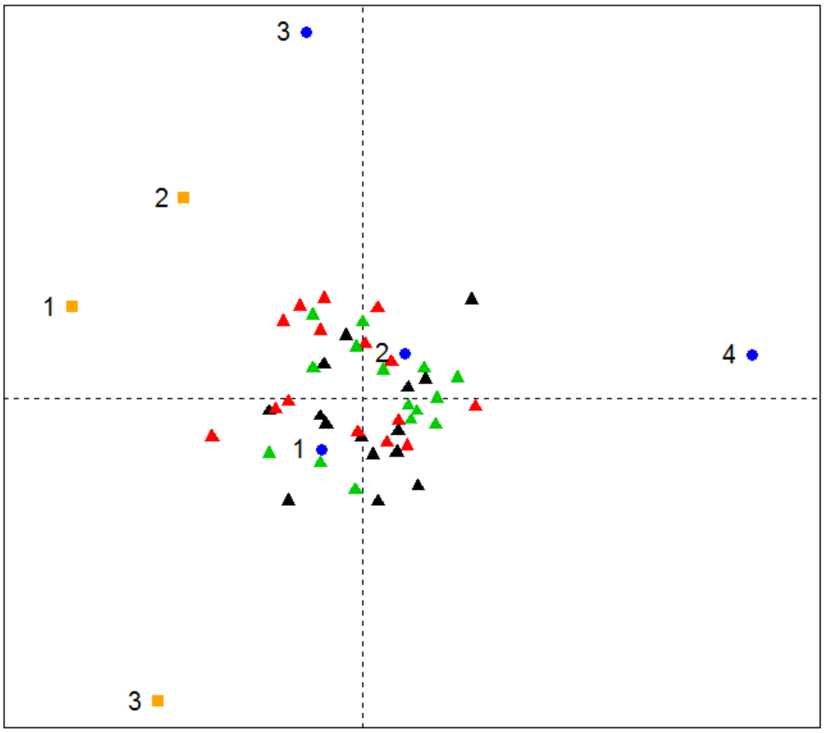


Figure 9.19: Triplot for the centered and scaled simulated data set two.

Figures 9.23 and 9.24 illustrate the centered and the centered and scaled data triplots together with the combination of variable and occasion mode axes. Although the variable and occasions modes are combined for illustration purposes, it is possible to combine other modes like the occasion and subject mode for example. Variable markers are then orthogonally projected onto these axes to get a sense of variable scores for the subjects at various occasions. This considers the interaction between all three modes simultaneously. Projecting subjects onto these axes provide a relative measure of the subject score on a particular variable at a particular occasion. It is akin to the wide combination PCA biplot discussed previously. The axes labels have been placed so that they appear on the side that corresponds with the direction of the combination vector. This means that it is possible to ascertain whether subjects scored positive or negative values by considering the angle between the subject marker and the axes of interest. Orthogonally projecting onto these axes in Figure 9.23 reveals that the variation in the data reduced over time, given that the projected values become more closely

	V1	V2	V3	V4	V5	V6	V7
Occasion 1	2639.93	3137.76	482.59	2181.44	1031.93	442.93	615.41
Occasion 2	461.89	1002.44	241.59	786	478.76	308	330.86
Occasion 3	193.62	930.59	177.03	411.38	356.31	231.56	176.86
Occasion 4	207	892.24	219.21	374.59	214.83	152.56	133.79

Table 9.8: Means for Mansoor data.

	V1	V2	V3	V4	V5	V6	V7
Occ. 1	8615268	11028624.3	171296	3308746	781366	150576	229052
Occ. 2	301346	1124500	82338	618915	295539	155757	156219
Occ. 3	80980	1 546370	51109	407141	85533	47792	40699
Occ. 4	83411	1740210	150462	162780	52049	14806	16642

Table 9.9: Variances for Mansoor data.

clustered when projected onto the axes for occasion 4 as opposed to those for occasion 1. This property cannot be seen in Figure 9.24 and that can only be attributable to the fact that the data used to construct the triplot was centered and scaled, the latter alteration being the pertinent one. Araújo (2009) does not place an interpretation on the angles between these combination mode axes but what is interesting is that the orientation of the axes suggest associations between variables that are indeed associated. In this case the scaled triplot performs better. For example variables 5 and 7 are strongly correlated at occasions 1 and 2. The same can be said for variables 1 and 3. The angles between the axes $V5.1$ and $V7.1$ as well as $V5.2$ and $V7.2$ is relatively small in Figure 9.24. The same observation is made for axes labelled $V1.1$ and $V3.1$ as well as $V1.2$ and $V3.2$. Consulting the correlation matrices in Chapter 3 and comparing to Figure 9.24 reveals numerous other examples. $V5$ and $V7$ show relatively weak association at occasions 3 and 4 but the plot suggests otherwise. It can thus be said that this is but one example though it provides evidence that the angles between the combination mode axes may be linked to correlations.

A final matter of interest is to determine whether the triplot provides answers to the investigative questions posed by Mansoor *et al.* (2009). By projecting observations onto the combination mode axes in Figure 9.23 it is evident that the relative scores of the HIV^+ infants (the black markers) was lower than

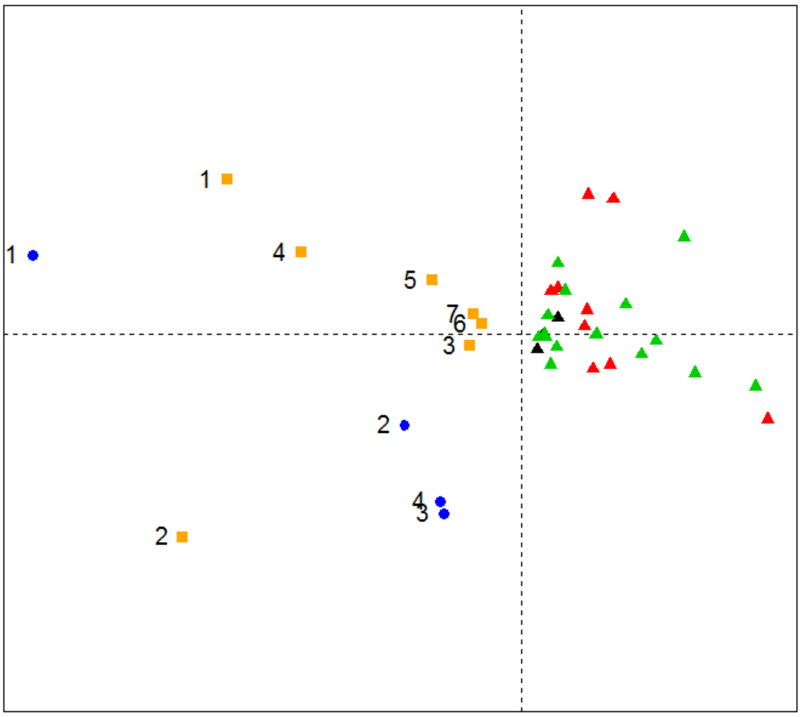


Figure 9.20: Triplot for the unprocessed Mansoor data.

the scores obtained in the other two groups. The implication is that *BCG* affects the uninfected groups differently to the infected group. Moreover, members of the uninfected groups tended to show higher scores on $V4$ which represented what is considered the required immune response. Not only did the triplot provide insight into the investigative questions but it also provided more information about relationships in the data.

9.8 Conclusion

This chapter focused on tensor decomposition techniques. It set out to determine how tensor decomposition techniques can be used to construct biplots and triplots. After a brief discussion on the basics of tensor algebra, the concept of tensor rank was discussed and a multilinear rank and outerproduct rank decomposition introduced. The Tucker3 decomposition was put in an SVD framework and used to construct biplots which were applied to the Mansoor data. These biplots revealed similar information about the data to that seen in previous chapters. Next, the PARAFAC decomposition and

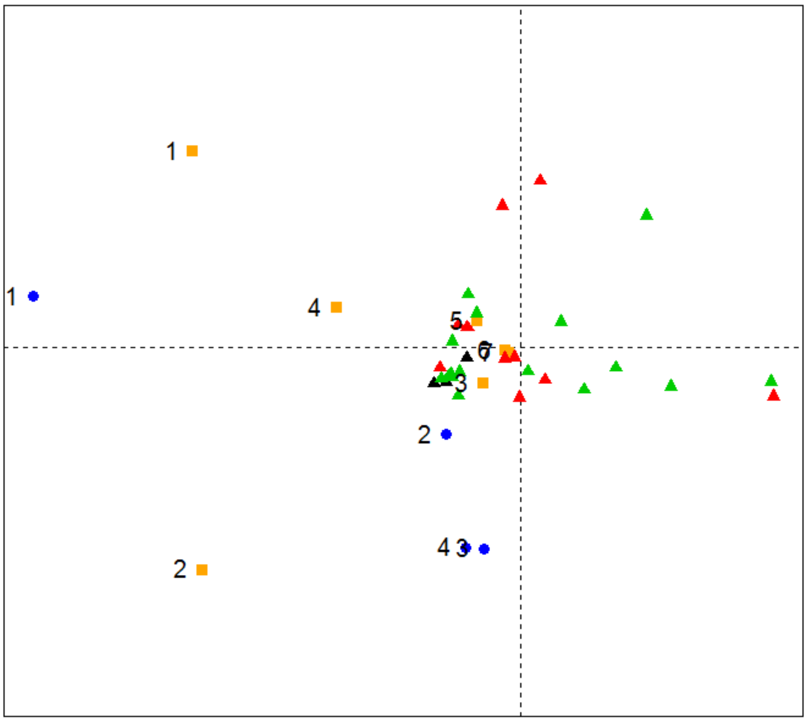


Figure 9.21: Triplot for the centered Mansoor data.

the problem of degeneracy were discussed. This was the reason why the PARAFAC decomposition was deemed inappropriate for the construction of an exploratory triplot. The tensor SVD decomposition was detailed and considered to be an alternative for triplot construction. After detailing the triplot construction, the method was applied to three data sets, two of which were simulated. The triplot interpretation was discussed and it was found to be a useful plot for revealing information about the data. The triplot was also seen to lack the property of scale invariance. Moreover, the scaled triplot tended to provide better information about the correlations within modes. The effects of scaling were detrimental to visualising separation in the data and so it was determined that the triplot for the centered data be used together with that for the scaled data, the latter triplot used only to ascertain correlations. There was also evidence to suggest that the angles between combination mode axes was in fact meaningful. The triplot served well as an exploratory tool in all instances reveals a great deal about the interactions between the various modes.

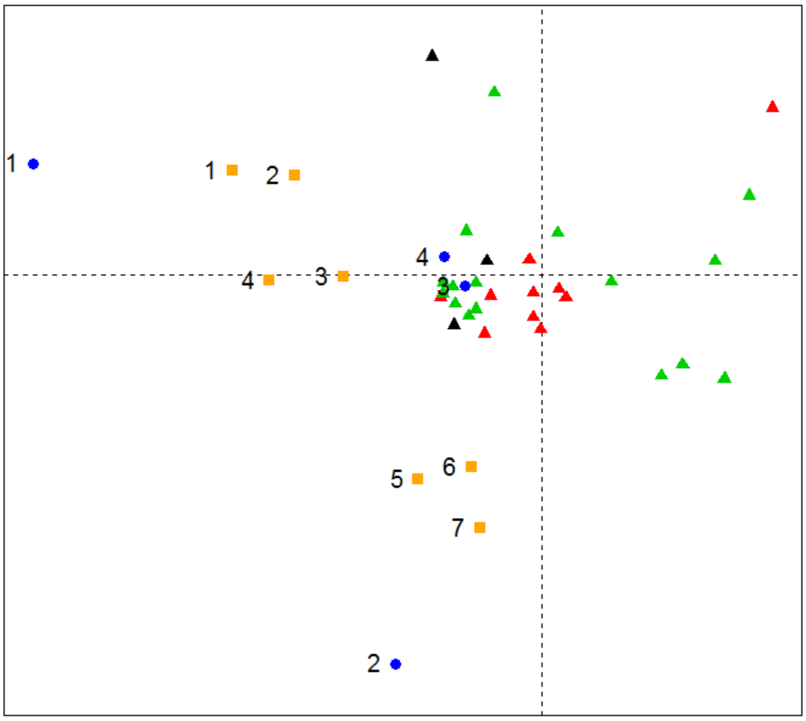


Figure 9.22: Triplot for the centered and scaled Mansoor data.

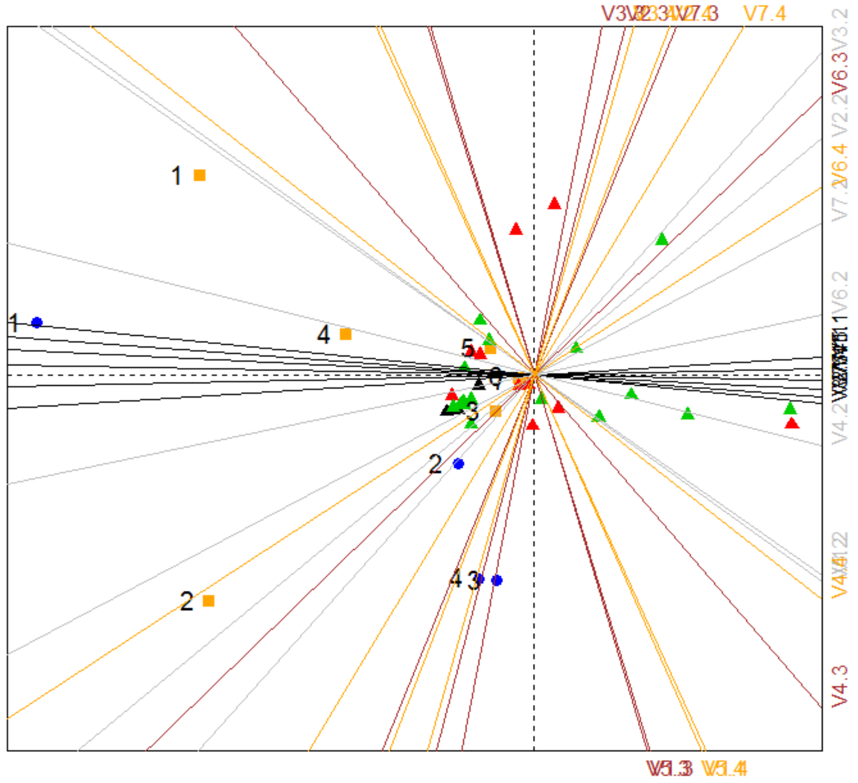


Figure 9.23: Triplot for the centered Mansoor data with variable and occasion mode combination axes shown.

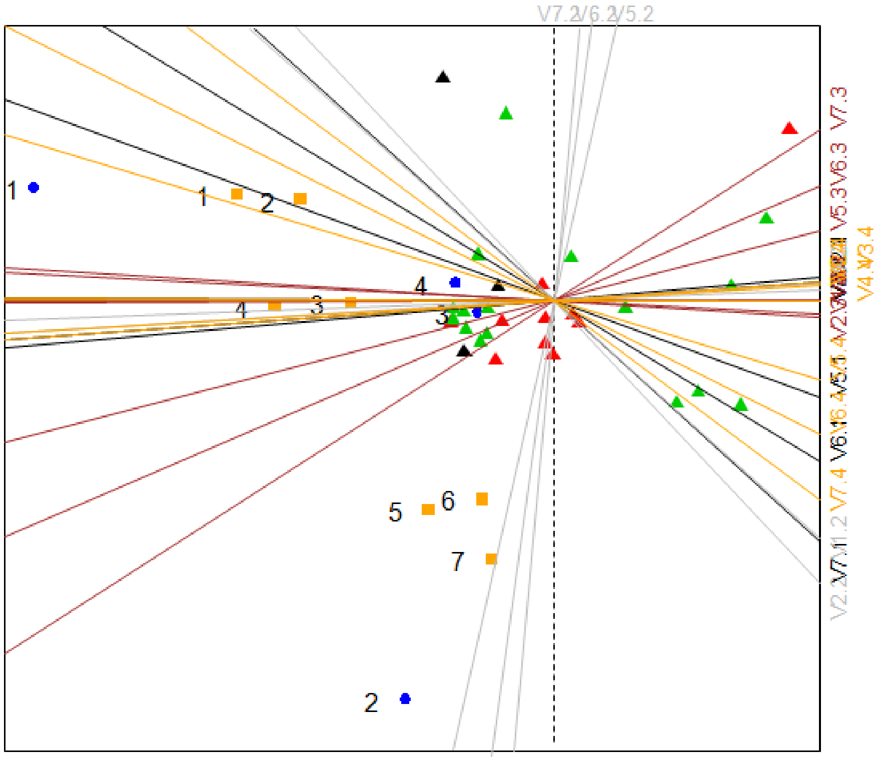


Figure 9.24: Triplot for the centered and scaled Mansoor data with variable and occasion mode combination axes shown.

Chapter 10

Conclusion

This dissertation concerned itself with investigating different methodologies for producing exploratory graphical plots for three mode data. More specifically, the aim was to investigate how biplots could be used in the exploratory analysis of three mode data. Furthermore, consideration was given to exploring the use of triplot methodology in the exploratory process. The research was born of the fact that the area of exploratory data analysis in a three mode context is not a well developed one. The objectives achieved in the course of this dissertation were as follows: It

1. Provided a comprehensive explanation of the theoretical foundations underpinning biplot construction, emphasising Principal Component Analysis as well as Canonical Variate Analysis Biplots;

2. Discussed the theoretical framework for two main classes of tensor decomposition techniques and explored whether these techniques could be used to produce biplots. Brief mention was made regarding triplot construction and interpreting some aspects of this plot;

3. Explored the use of other multivariate techniques for the construction of biplots in a three mode data context;

4. Applied all these methods to a data set taken from Mansoor *et al.* (2009) in order to determine whether the different techniques conveyed similar information about the structure of the data. Careful consideration was given to the interpretation as well as any similarities and differences that arose in the plots.

The primary research questions were thus how biplots could be used to explore three mode data and whether different methodologies yielded similar conclusions about the relationships hidden in the data. This chapter will show what was concluded regarding each of the objectives, provide a sense of the significance of the research and make recommendations on future work.

10.1 Conclusions regarding objectives

Each objective listed will be considered in turn to provide a sense of how it was satisfied in this dissertation. Chapters 4 and 5 served to meet the first objective in that they provided a comprehensive account of the theoretical foundations of biplot construction together with specific information on the PCA biplot as well as the CVA biplot. It was written not only to provide a rigorous mathematical understanding of the methodology, but also to leave the reader with a sound intuitive appreciation of the methodology from a geometric perspective. The techniques were used to construct biplots for the matricised forms of the data.

The second objective of this research was addressed in Chapter 9, where the reader was provided with rudimentary knowledge regarding the two main contenders in the realm of tensor decomposition, namely the Tucker3 and PARAFAC models. The focal point was the concept of rank and how this generalises to third order tensors. Since the SVD framework is fundamental in the biplot context, each of the tensor decomposition methods were discussed with respect to the SVD properties that they preserve. Complexities regarding PARAFAC solutions were discussed in order to substantiate the argument that it is not a viable model for constructing an exploratory plot. The Tucker3 decomposition was used to construct biplots. These plots were simply applied to the Mansoor data and the results discussed. It was noted that this construction depended on the left-singular vectors in each mode, which is different to the two-mode PCA counterpart. The triplot methodology that Araújo (2009) based on the PARAFAC framework was thus placed in the context of the third and final tensor decomposition technique discussed, the Tensor SVD. Araújo (2009) proposed the triplot but did little to discuss interpretational issues, even in an intuitive sense and placing the methodology in a different framework introduced new considerations, so the effects had to be explored. The effects of data preprocessing on these plots were also considered. It must be said that the conclusions regarding interpretation of the triplot in particular were based on empirical evidence and had no rigorous mathematical proof. The triplot showed itself to be a useful tool for exploratory analysis particularly

The third objective has the broadest reach in terms of the content of this dissertation. The concluding remarks will emphasise interpretational issues surrounding the constructed plots. Chapters 6, 7 and 8 served to satisfy this objective in that each of these chapters discussed specific techniques that could be used to construct exploratory plots. The application aspects of Chapters 4 and 5 also contributed to fulfilling this objective. Chapter 4 constructed PCA biplots for the matricised forms of the data and discussed interpretational matters surrounding the biplots. It was established that the tall combination biplot was preferred in order to get a sense of the Euclidean distances between subjects, due to the construction process as well as the benefit of immediate visual appraisal. To an extent, the angles between variable axes represented persistent relationships in the data, so that two highly correlated variables would tend to be thusly related across occasion or condition. The plot also provided information about changes in variation in the data. The wide combination biplot afforded the means to understand how a variable relates to itself across occasion or condition. The triplot was also seen to lack the property of scale invariance. Moreover, the scaled triplot tended to provide better information about the correlations within modes. The effects of scaling were detrimental to visualising separation in the data and so it was determined that the triplot for the centered data be used together with that for the scaled data. The latter triplot used only to ascertain correlations. There was also evidence to suggest that the angles between combination mode axes was in fact meaningful.

Chapter 6 detailed the use of Generalised Orthogonal Procrustes Analysis to construct a biplot based on combining the k separate PCA biplots constructed from the separate data matrices comprising the three-mode data set. Careful consideration had to be given to the process of superimposing the k separate biplots after rotation and it was determined that the optimal rotation could be based on the sample point representation, variable axes representation or a combination of the two. The chapter was focused on using biplots that optimally approximate the Euclidean distances between sample points given the nature of the Mansoor *et al.* (2009) investigation. It was argued that the biplot resulting from the application of this technique did not preserve the Euclidean distances between the observations primarily because of the construction process and issues of scale. Despite this, the plot served well to show changes in variation in the data as well as the extent of the separation between occasions as determined by the mean vectors at each occasion. The plot also afforded the means to visually appraise how the relationship between variables evolved over time or in a different context would allow the researcher to compare the correlations across condition. It also allowed the researcher to read off approximated variable scores for the subjects. Even with its faults, it served well as an exploratory tool.

Chapter 7 was concerned with the use of Common Principal Component Analysis (CPC) and how the researcher could use the estimated results from such an analysis in order to construct a biplot. After detailing the theoretical foundations of CPC and its counterpart for data that shows dependence across occasion, the biplot construction was described. Importantly, it was argued that the angles between the displayed variables represented systemic relationships that persisted across occasion or condition. Furthermore, it was argued that those occasions or conditions with comparatively larger variation would be better represented since these occasions or conditions would exert greater influence in the estimation procedure. The Euclidean distances between observations was not optimally approximated but the plot did afford the means to appreciate changes in variation in the data as well as separation between occasions or conditions. Axes calibration had to be carefully considered and two methods of calibration were provided. The content of this chapter represented a prelude to a novel development in Chapter 8.

This chapter is particularly significant in that it represents an original contribution to the field of exploratory data analysis. The methodology discussed here was inspired by the concepts described in Chapter 6 where separate PCA biplots could be superimposed to form a single comprehensive plot. Attempting to perform a similar task with CVA biplots was not as simple because the data are transformed to be represented in the canonical space, and this implied that separate CVA biplots, could not simply be superimposed. The contribution of this chapter was to develop a method that could be used to produce a consolidated CVA biplot and tests with simulated data showed that the technique was successful in capturing the separation between groups comprising the data.

The final objective related to the application of the various techniques to the Mansoor data in order to ascertain whether they conveyed similar information about the data. Mansoor *et al.* (2009) set out to investigate the effect of BCG, a Tuberculosis vaccine, on the immune response in HIV^+ infants. To do this , BCG was administered to infants classified into one of three groups and cytokine expression measurements taken over time. The groups comprised HIV^+ infants, uninfected and exposed infants as well as uninfected and unexposed infants. The primary investigative questions were as follows:

1. Does BCG induce the required immune response against Tuberculosis

in HIV^+ infants?

2. Is the immune response in the groups comprising the uninfected infants similar?

All of the techniques described yielded the same conclusions regarding these questions, and it was hypothesised that this would indeed be the case. $V4$ of the data was related to the required immune response, and it was seen that HIV^+ infants scored relatively low on this variable. It was also seen that the immune response induced in the uninfected infants was similar. Further observations included the fact that the variation in the data reduced across occasion and that scores tended to decrease over time. Occasion 1 showed itself to be quite different from the other occasions in terms of subjects scores, with occasions 3 and 4 displaying similar characteristics. Ultimately all the techniques agreed, some emphasising different characteristics described here.

10.2 Significance of research

The major contribution of this dissertation is two-fold. Firstly, it adds substance to a field for which there is not a vast body of literature. Secondly, it makes novel contributions in the form of placing triplot methodology in a different tensor decomposition framework, as well as discussing the development of a technique that allows CVA biplots to be displayed on a single plot. The first contribution means that researchers can now find cohesive research on matters of exploratory data analysis, in a three mode context. This will provide them with varying techniques for displaying the data that are based on PCA and biplot methodology. The triplot, a display technique that was found in a rather obscure work by Araújo (2009), was placed in a different framework to make it more generally applicable and due consideration was given to attempting to interpret it albeit on the basis of empirical evidence.

10.3 Recommendations

It is not a simple matter to recommend any one of the techniques as superior to the others. The techniques conveyed similar information about the data and worked well to visually represent the relationships in the data. The choice of technique is therefore informed by what the researcher's is investigating. For example, if the researcher sets out to do a CPC analysis then the discussion in Chapter 7 would serve well to better understand the data.

It was seen that the simplest of techniques, matricising the data, performed well and would thus serve as a good starting point in any exploratory analysis. It is this observation that makes a strong case for using this method as a means to better understand the data. The simplest of techniques often prove to be valuable and it is strongly recommended that this method be used before embarking on a more rigorous analysis. If interest lies in visualising changes in the mean of the data over occasion or condition then it is recommended that the Procrustes technique be used because it is most successful in displaying this separation. When faced with grouped data, it is recommended that the common CVA biplot be employed in order to understand the extent of the separation between the groups comprising the data. It is strongly recommended that the triplot be used regardless of the nature of the investigation because it is arguably the most revealing of the exploratory methods discussed, embodying all characteristics of the data where as other methods emphasised different characteristics of the data.

In terms of future research there is a great deal that can be done. While this dissertation investigated selected techniques, there are a myriad other three mode modelling means such as STATIS (Lavit *et al.*, 1994) for example that could lend themselves to constructing exploratory plots. Furthermore, the triplot methodology can be further investigated to provide rigorous proofs for that which is suggested by the empirical evidence. Thought can also be given to the calibration of combination mode axes and whether there is a way to explicitly incorporate the grouped nature of data into the triplot construction. Finally, the construction of quality measures for the triplot as well as the Tucker3 biplot is also an avenue that can be explored.

References

Acar, E., & Yener, B. (2009). Unsupervised multiway data analysis: A literature survey. *Knowledge and Data Engineering, IEEE Transactions on*, 21(1), 6-20.

Andersson, C. A. & Bro, R. (1998). Improving the speed of multi-way algorithms: Part I. Tucker3. *Chemometrics and Intelligent Laboratory Systems*, 42(1), 93-103.

Andersson, C. A. & Bro, R. (2000). The N-way toolbox for MATLAB. *Chemometrics and Intelligent Laboratory Systems*, 52(1), 1-4.

Araújo, L. B. (2009). *Seleção e análise dos modelos PARAFAC e Tucker e gráfico triplot com aplicação em interação tripla*. Doctoral Thesis, Escola Superior de Agricultura Luiz de Queiroz, University of Sao Paulo, Piracicaba.

Beaghen, M. (1997). *Canonical variate analysis and related methods with longitudinal data*. Doctoral dissertation, Virginia Polytechnic Institute and State University.

Bro, R. (1998).*The N-way on-line course on PARAFAC and PLS*. Retrieved from http://www.models.life.ku.dk/courses/

Carroll, J. D. & Chang, J. J. (1970). Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition. *Psychometrika*, 35(3), 283-319.

Cattell, R. B. (1944).“Parallel proportional profiles” and other principles for determining the choice of factors by rotation. *Psychometrika*, 9(4), 267-283.

Chen, J. & Saad, Y. (2009). On the tensor SVD and the optimal low rank orthogonal approximation of tensors. *SIAM Journal on Matrix Analysis and Applications*, 30(4), 1709-1734.

Cox, T. F., & Cox, M. A. (2000). *Multidimensional scaling.* London: Chapman & Hall.

De Lathauwer, L., De Moor, B. & Vandewalle, J. (2000). A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4), 1253-1278.

De Silva, V. & Lim, L. H. (2008). Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3), 1084-1127.

Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3), 211-218.

Efron, B. (2007). The future of statistics. *Amstat News*, 363, 47-50.

Flury, B. N. (1984). Common principal components in k groups. *Journal of the American Statistical Association*, 79(388), 892-898.

Flury, B. N. & Gautschi, W. (1986). An algorithm for simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly diagonal form. *SIAM Journal on Scientific and Statistical Computing*, 7(1), 169-184.

Flury, B. (1988). *Common principal components & related multivariate models.* New York: Wiley.

Flury, B. D. & Neuenschwander, B. E. (1994). Simultaneous diagonalization algorithms with applications in multivariate statistics. *In Proceedings of the conference on Approximation and computation: a fetschrift in honor of Walter Gautschi.* Boston: Birkhauser.

Gabriel, K. R. (1971). The biplot graphic display of matrices with application to principal component analysis. *Biometrika*, 58(3), 453-467.

Gabriel, K. R. (1972). Analysis of Meteorological Data by Means of Canonical Decomposition and Biplots. *Journal of Applied Meteorology*, 11, 1071-1077.

Gower, J. C. (1971). Statistical methods of comparing different multivariate analyses of the same data. In F. R. Hodson & D.G. Kendall (Eds.), *Mathematics in the Archaeological and Historical Sciences. Proceedings of the AngloRomanian Conference* (pp 138-149). Edinburgh: University Press.

Gower, J. C., & Hand, D. J. (1996). *Biplots.* London: Chapman & Hall/CRC.

Gower, J. C. & Dijksterhuis, G. B. (2004). *Procrustes problems.* Oxford: Oxford University Press.

Gower, J.C., Lubbe, S. & Le Roux, N.J. (2011). *Understanding biplots.* Chichester: Wiley.

Green, B. F. (1952). The orthogonal approximation of an oblique structure in factor analysis. *Psychometrika*, 17(4), 429-440.

Harshman, R. A. (1970). Foundations of the PARAFAC procedure: models and conditions for an "explanator" multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16, 1-84.

Harshman, R. A., & Lundy, M. E. (1984). Data preprocessing and the extended PARAFAC model. In H. G. Law, C. W. Snyder, Jr., J. Hattie, & R. P. McDonald (Eds.), *Research methods for multimode data analysis* (pp.216-284). New York: Praeger.

Harshman, R. A., & DeSarbo, W. S. (1984). An application of PARAFAC to a small sample problem, demonstrating preprocessing, orthogonality constraints, and split-half diagnostic techniques. In H. G. Law, C. W. Snyder, Jr., J. Hattie, & R. P. McDonald (Eds.), *Research methods for multimode data analysis* (pp.602-642). New York: Praeger.

Harshman, R.A. (2004). *The problem and nature of degenerate solutions or decompositions of 3-way arrays.* Talk at the Tensor Decompositions Workshop, July 1923. Palo Alto, CA: AIM.

Hitchcock, F. L. (1927a) The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics* ,6, 164-189.

Hitchcock, F. L. (1927b). Multiple invariants and generalized rank of a p-way matrix or tensor. *Journal of Mathematics and Physics*, 7(1), 39-79.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(7), 498-520.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 28(4), 321-377.

Hurley, J. R. & Cattell, R. B. (1962). The Procrustes program: Producing direct rotation to test a hypothesized factor structure. *Behavioral Science*, 7(2), 258-262.

Ibelgaufts, H. (2009). T-helper. In *Horst Ibelgaulfts' COPE: cytokines & cells online pathfinder encyclopedia*. Retrieved May 2011 from http://www.copewithcytokines.de/cope.cgi?key=T-helper

Jacobi, C. (1846). Uber die Kreisteilung. J. Reine Angew. *Math*, 254-274.

Jolliffe, I. T. (2002). *Principal component analysis*. New York: Springer-Verlag.

Karulin, A. Y., Hesse, M. D., Tary-Lehmann, M. & Lehmann, P. V. (2000). Single-cytokine-producing CD4 memory cells predominate in type 1 and type 2 immunity. *The Journal of Immunology*, 164(4), 1862-1872.

Kiers, H. A. (2000a). Some procedures for displaying results from three-mode methods. *Journal of Chemometrics*, 14(3), 151-170.

Kiers, H. A. (2000b). Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics*, 14(3), 105-122.

Kolda, T. G. & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, 51(3), 455-500.

Kroonenberg, P. M. & De Leeuw, J.(1977). TUCKALS2: Een hoofdassenanalyse voor drieweggegevens. *Methoden en Data Nieuwsbrief*, 30-53.

Kroonenberg, P. M. (1983). *Three-mode principal component analysis: Theory and applications*. Leiden: DSWO press.

Kroonenberg, P. M. (2008). *Applied multiway data analysis*. New York: Wiley.

Kruskal, J. B. (1977). Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics.*Linear Algebra and its Applications*, 18(2), 95-138.

Lavit, C., Escoufier, Y., Sabatier, R. & Traissac, P. (1994). The act (statis method). *Computational Statistics & Data Analysis*, 18(1), 97-119.

Lim L. H. (2004). *Whats possible and whats not possible in tensor decompositions a freshmans views.* Talk at the Tensor Decompositions Workshop, July 1923. Palo Alto, CA: AIM.

Le Roux, N.J. (2012). Personal communication

Mahalanobis, P. C. (1936). On the generalized distance in statistics. *In Proceedings of the National Institute of Sciences of India*, 2(1), 49-55.

Mansoor, N., Scriba, T. J., de Kock, M., Tameris, M., Abel, B., Keyser, A., ... & Hanekom, W. A. (2009). HIV-1 infection in infants severely impairs the immune response induced by Bacille Calmette-Guerin vaccine. *Journal of Infectious Diseases*, 199(7), 982-990.

Mosier, C. I. (1939). Determining a simple structure when loadings for certain tests are known. *Psychometrika*, 4(2), 149-162.

Neuenschwander, B. E. & Flury, B. D. (2000). Common principal components for dependent random vectors. *Journal of Multivariate Analysis*, 75(2), 163-183.

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11), 559-572.

Rao, C. (1965). *Linear statistical inference and its applications.* New York: Wiley.

R Development Core Team (2011). R: A language and environment for statistical computing. *R Foundation for Statistical Computing, Vienna, Austria.* Retrieved from http://www.R-project.org.

Appendix A

R Code

Due to the fact that the function PCAbiplot contained in the package accompanying Gower *et al.* (2011), it will be shown in its entireity and only the relevant modifications made to this function in each chapter shown.

```
function (X, G = NULL, scaled.mat = FALSE, dim.biplot = c(2,
   1, 3), e.vects = 1:ncol(X), correlation.biplot = FALSE, classes = 1:ncol(
       G),
     samples.new = NULL, predict.samples = NULL, predict.means = NULL,
       samples = list(...), ax.new = NULL, ax = list(...), new.samples =
           list            (...), class.means = list(...), alpha.bags = list
           (...), kappa.ellipse =          list(...), density.style = list
           (...), colour.scheme = NULL, Title =                NULL, exp.
           factor = 1.2, reflect = c(FALSE, "x", "y"), rotate = 0,
            select.origin = FALSE, adequacies.print = FALSE,...)
{
    dim.biplot <- dim.biplot[1]
    if (dim.biplot != 1 & dim.biplot != 2 & dim.biplot != 3)
        stop("Only 1D, 2D and 3D biplots")
    e.vects <- e.vects[1:dim.biplot]
    reflect <- reflect[1]
    X.info <- biplot.check.X(X, scaled.mat)
    X <- X.info$X
    unscaled.X <- X.info$unscaled.X
    means <- X.info$means
    sd <- X.info$sd
    G <- biplot.check.G(G, nrow(X))
    if (!is.null(samples.new))
        X.new <- scale(samples.new, center = means, scale = sd)
    else X.new <- NULL
    n <- nrow(X)
    p <- ncol(X)
    J <- ncol(G)
    if (!all(is.numeric(classes)))
        classes <- match(classes, dimnames(G)[[2]], nomatch = 0)
    classes <- classes[classes <= J]
    classes <- classes[classes > 0]
    svd.out <- svd(X)
    V.mat <- svd.out$v
    U.mat <- svd.out$u
    Sigma.mat <- diag(svd.out$d)
    Vr <- svd.out$v[, e.vects, drop = F]
    fit.out <- biplot.fit.measures(mat = X, mat.hat = X %*% Vr %*%
        t(Vr), eigenvals = svd.out$d^2, eigenvecs = svd.out$v,
        dims = e.vects)
    quality <- fit.out$quality
    adequacy <- fit.out$adequacy
    axis.predictivity <- fit.out$axis.predictivity
    sample.predictivity <- fit.out$item.predictivity
    if (adequacies.print & predictivity.print)
        stop("adequacies.print and predictivity.print cannot both be set to
                         True")
    if (adequacies.print)
        dimnames(X)[[2]] <- paste(dimnames(X)[[2]], " (", adequacy,
         ")", sep = "")
    if (predictivity.print)
        dimnames(X)[[2]] <- paste(dimnames(X)[[2]], " (", round
                               (axis.predictivity,
            digits = 2), ")", sep = "")
    reflect.mat <- diag(dim.biplot)
    if (reflect == "x" & dim.biplot < 3)
        reflect.mat[1, 1] <- -1
    if (reflect == "y" & dim.biplot == 2)
        reflect.mat[2, 2] <- -1
    if (reflect == "xy" & dim.biplot == 2)
        reflect.mat[1:2, 1:2] <- diag(-1, 2)
    rotate.mat <- diag(dim.biplot)
    if (dim.biplot == 2) {
        if (!is.null(ax$rotate)) {
            if (is.numeric(ax$rotate)) {
                radns <- pi * rotate/180
                rotate.mat <- matrix(c(cos(radns), -sin(radns),
                  sin(radns), cos(radns)), ncol = 2)
            }
            else {
                if (ax$rotate == "maxpred") {
                  ax$rotate <- (names(axis.predictivity))[axis.predictivity
                      ==
                    max(axis.predictivity)]
                  ax$rotate <- match(ax$rotate, dimnames(X)[[2]])
                }
                else ax$rotate <- match(ax$rotate, dimnames(X)[[2]])
                radns <- -atan2(V.mat[ax$rotate, e.vects[2]],
                  V.mat[ax$rotate, e.vects[1]])
                rotate.mat <- matrix(c(cos(radns), -sin(radns),
                  sin(radns), cos(radns)), ncol = 2)
            }
        }
    }
    class.means.mat <- as.matrix(solve(t(G) %*% G) %*% t(G) %*%
        unscaled.X, ncol = ncol(unscaled.X))
    Z.new <- NULL
    Z.means.mat <- NULL
    if (correlation.biplot) {
        lambda.r <- diag(svd(t(X) %*% X)$d[1:dim.biplot])
        Z <- sqrt(n - 1) * X %*% Vr %*% rotate.mat %*% reflect.mat %*%
          (sqrt(solve(lambda.r)))
        if (!is.null(class.means))
            Z.means.mat <- sqrt(n - 1) * scale(class.means.mat,
                means, sd) %*% Vr %*% rotate.mat %*% reflect.mat %*%
                (sqrt(solve(lambda.r)))
        if (!is.null(X.new))
            Z.new <- sqrt(n - 1) * X.new %*% Vr %*% rotate.mat %*%
                reflect.mat %*% (sqrt(solve(lambda.r)))
    }
    else {
        #Z REPRESENTS THE SAMPLE POINTS TO BE PLOTTED#
        Z <- X %*% Vr %*% rotate.mat %*% reflect.mat
        if (!is.null(class.means))
            Z.means.mat <- scale(class.means.mat, means, sd) %*%
                Vr %*% rotate.mat %*% reflect.mat
        if (!is.null(X.new))
            Z.new <- X.new %*% Vr %*% rotate.mat %*% reflect.mat
    }
    dimnames(Z) <- list(dimnames(X)[[1]], NULL)
    if (!is.null(X.new))
        if (is.null(dimnames(samples.new)[[1]]))
            dimnames(Z.new) <- list(paste("N", 1:nrow(Z.new),
                sep = ""), NULL)
        else dimnames(Z.new) <- list(dimnames(samples.new)[[1]],
            NULL)
    if (is.matrix(ax.new)) {
        NewVarsMeans <- apply(ax.new, 2, mean)
        if (scaled.mat)
            NewVarsSD <- sqrt(apply(ax.new, 2, var))
        else NewVarsSD <- rep(1, ncol(ax.new))
    }
    num.vars <- p
    var.names <- dimnames(X)[[2]]
    if (!is.null(ax.new)) {
        means <- c(means, NewVarsMeans)
        sd <- c(sd, NewVarsSD)
        unscaled.X <- cbind(unscaled.X, ax.new)
        num.vars <- ncol(unscaled.X)
        if (!is.null(dimnames(ax.new)[[2]]))
            var.names <- c(var.names, dimnames(ax.new)[[2]])
        else var.names <- c(var.names, paste("NV", 1:ncol(ax.new),
            sep = ""))
        SigmaMinOne <- ifelse(Sigma.mat < 1e-10, 0, 1/Sigma.mat)
        Vr.new <- t(scale(ax.new, center = TRUE, scale = NewVarsSD)) %*%
            U.mat %*% SigmaMinOne
        Vr.new <- Vr.new[, e.vects]
        Vr.all <- rbind(Vr, Vr.new)
    }
    else Vr.all <- Vr
    #DETERMINING AXES DIRECTION PREDICTIVE OR INTERPOLATIVE#
    ax <- do.call("biplot.ax.control", c(num.vars, list(var.names),ax))
    if (ax$type == "prediction") {
        if (correlation.biplot)
            axes.direction <- (sqrt(n - 1)/(diag(Vr.all %*% lambda.r %*%
                t(Vr.all)))) * Vr.all %*% sqrt(lambda.r) %*%
                rotate.mat %*% reflect.mat
        else axes.direction <- 1/(diag(Vr.all %*% t(Vr.all))) *
            Vr.all %*% rotate.mat %*% reflect.mat
    }
    else {
        if (correlation.biplot)
            axes.direction <- sqrt(lambda.r) %*% Vr.all %*% rotate.mat %*%
                reflect.mat
        else axes.direction <- Vr %*% rotate.mat %*% reflect.mat
    }
    if (length(ax$which) == 0)
        z.axes <- NULL
    #CALIBRATING THE AXES#
    else z.axes <- lapply(1:length(ax$which), calibrate.axis,
        unscaled.X, means, sd, axes.direction, ax$which, ax$ticks,
         ax$orthogx, ax$orthogy, ax$oblique)
          alpha.bags <- do.call("biplot.alpha.bag.control", c(J, list(
              dimnames            (G)[[2]]), alpha.bags))
    z.bags <- vector("list", length(alpha.bags$which))
    if (length(alpha.bags$which) > 0)
        for (j in 1:length(alpha.bags$which)) {
            class.num <- alpha.bags$which[j]
            mat <- Z[G[, class.num] == 1, ]
            flush.console()
            cat(paste("alpha bag for class ", dimnames(G)[[2]][class.num],
                " with ", nrow(mat), " samples", sep = ""), "\n")
            if (dim.biplot == 2)
                z.bags[[j]] <- calc.alpha.bags(mat, alpha.bags$alpha[j],
                  alpha.bags$max[j], alpha.bags$Tukey.median[j],
                  alpha.bags$min[j])
            if (dim.biplot == 1)
                z.bags[[j]] <- quantile(mat, c((100 - alpha.bags$alpha[j])/
                    200,
                  1 - (100 - alpha.bags$alpha[j])/200, 0.5))
        }
    kappa.ellipse <- do.call("biplot.kappa.ellipse.control",
        c(J, list(dimnames(G)[[2]]), dim.biplot, kappa.ellipse))
    z.ellipse <- vector("list", length(kappa.ellipse$which))
    if (length(kappa.ellipse$which) > 0)
        for (j in 1:length(kappa.ellipse$which)) {
            class.num <- kappa.ellipse$which[j]
            mat <- Z[G[, class.num] == 1, ]
            if (dim.biplot == 2)
                z.ellipse[[j]] <- calc.concentration.ellipse(mat,
                  kappa.ellipse$kappa[j])
            if (dim.biplot == 1)
                z.ellipse[[j]] <- qnorm(c(1 - pnorm(kappa.ellipse$kappa[j]),
                  pnorm(kappa.ellipse$kappa[j])), mean(mat),
                  sqrt(var(mat)))
            if (dim.biplot == 3) {
                require(rgl)
                z.ellipse[[j]] <- ellipse3d(x = var(mat), centre = apply(mat
                    ,
                  2, mean), t = kappa.ellipse$kappa[j])
            }
        }
    if (dim.biplot == 1) {
        density.style <- do.call("biplot.density.1D.control",
            c(J, list(dimnames(G)[[2]]), density.style))
        z.density <- vector("list", length(density.style$which))
        if (length(density.style$which) > 0)
            for (j in 1:length(density.style$which)) {
                class.num <- density.style$which[j]
                mat <- Z[G[, class.num] == 1, ]
                z.density[[j]] <- density(mat, bw = density.style$bw[j],
                  kernel = density.style$kernel[j])
            }
    }
    if (dim.biplot == 2) {
        density.style <- do.call("biplot.density.2D.control",
            c(J, list(dimnames(G)[[2]]), density.style))
        if (!is.null(density.style$which)) {
            if (density.style$which == 0)
                mat <- Z
            else mat <- Z[G[, density.style$which] == 1, ]
            x.range <- range(Z[, 1])
            y.range <- range(Z[, 2])
            width <- max(x.range[2] - x.range[1], y.range[2] -
                y.range[1])
            xlim <- mean(Z[, 1]) + c(-1, 1) * 0.75 * width
            ylim <- mean(Z[, 1]) + c(-1, 1) * 0.75 * width
            if (is.null(density.style$h))
                z.density <- kde2d(mat[, 1], mat[, 2], n = density.style$n,
                  lims = c(xlim, ylim))
            else z.density <- kde2d(mat[, 1], mat[, 2], h = density.style$h,
                n = density.style$n, lims = c(xlim, ylim))
        }
        else z.density <- NULL
    }
    if (dim.biplot == 3) {
        density.style <- do.call("biplot.density.2D.control",
            c(J, list(dimnames(G)[[2]]), density.style))
        if (!is.null(density.style$which))
            warning("No density plots in 3D")
    }
    if (!is.null(colour.scheme)) {
        my.sample.col <- colorRampPalette(colour.scheme)
        samples$col <- my.sample.col(samples$col)
    }
    samples <- do.call("biplot.sample.control", c(J, samples))
    new.samples <- do.call("biplot.new.sample.control", c(max(1,
        nrow(X.new)), new.samples))
    class.means <- do.call("biplot.mean.control", c(J, list(dimnames(G)
        [[2]]),
        class.means))
    legend.format <- do.call("biplot.legend.control", legend.format)
    legend.type <- do.call("biplot.legend.type.control", legend.type)
    if (dim.biplot == 2)
        draw.biplot(Z = Z, G = G, classes = classes, Z.means = Z.means.mat,
            z.axes = z.axes, z.bags = z.bags, z.ellipse = z.ellipse,
             Z.new = Z.new, Z.density = z.density, sample.style = samples,
              mean.style = class.means, ax.style = ax, bag.style = alpha.
                  bags,
               ellipse.style = kappa.ellipse, new.sample.style = new.samples
                   ,
                density.style = density.style, predict.samples =
                                                    predict.samples,
                    predict.means = predict.means, Title =
                                            Title, exp.factor = exp.factor
                    ,...)
    if (dim.biplot == 1)
        draw.biplot.1D(Z = Z, G = G, classes = classes, Z.means = Z.means.
            mat,
            z.axes = z.axes, z.bags = z.bags, z.ellipse = z.ellipse,
             Z.new = Z.new, Z.density = z.density, sample.style = samples,
               mean.style = class.means, ax.style = ax, bag.style = alpha.
                   bags,
                 ellipse.style = kappa.ellipse, new.sample.style = new.
                     samples,
                  density.style = density.style, predict.samples =
                                                      predict.samples,
                      predict.means = predict.means, Title =
                                          Title, exp.factor = exp.factor,
                      ...)
    if (dim.biplot == 3)
        draw.biplot.3D(Z = Z, G = G, classes = classes, Z.means = Z.means.
            mat,
           z.axes = z.axes, z.bags = z.bags, z.ellipse = z.ellipse,
             Z.new = Z.new, sample.style = samples, mean.style = class.means
```

```
                ,
              ax.style = ax, bag.style = alpha.bags, ellipse.style =
                                  kappa.ellipse, new.sample.style = new.
                  samples,
                predict.samples = predict.samples, predict.means =
                                      predict.means, Title = Title, exp.
                    factor = exp.factor, ...)
    if (!is.null(ax$oblique) & ax$type == "interpolation")
        points(0, 0, pch = "+", cex = 2)
    if (!is.null(predict.samples))
        predict.mat <- scale(Z[predict.samples, , drop = F] %*%
            t(reflect.mat) %*% t(rotate.mat) %*% t(Vr), center = F,
            scale = 1/sd)
    else predict.mat <- NULL
    if (!is.null(predict.mat))
        predict.mat <- scale(predict.mat, center = -means, scale = F)
    if (!is.null(predict.means))
        predict.means.mat <- scale(Z.means.mat[predict.means,
            , drop = F] %*% t(reflect.mat) %*% t(rotate.mat) %*%
            t(Vr), center = F, scale = 1/sd)
    else predict.means.mat <- NULL
    if (!is.null(predict.means.mat))
        predict.mat <- rbind(predict.mat, scale(predict.means.mat,
            center = -means, scale = F))
    if (!is.null(predict.mat))
        dimnames(predict.mat) <- list(c(dimnames(X)[[1]][predict.samples],
            dimnames(G)[[2]][predict.means]), dimnames(X)[[2]])
    if (any(unlist(legend.type))) {
        windows()
        sample.list <- list(pch = samples$pch, col = samples$col)
        mean.list = list(pch = rep(NA, J), col = rep(NA, J))
        mean.list$pch[class.means$which] <- class.means$pch
        mean.list$col[class.means$which] <- class.means$col
        bag.list = list(lty = rep(1, J), col = rep(NA, J), lwd = rep(NA,
            J))
        bag.list$lty[alpha.bags$which] <- alpha.bags$lty
        bag.list$col[alpha.bags$which] <- alpha.bags$col
        bag.list$lwd[alpha.bags$which] <- alpha.bags$lwd
        if (length(alpha.bags$which) == 0 & length(kappa.ellipse$which) >
            0) {
            bag.list$lty[kappa.ellipse$which] <- kappa.ellipse$lty
            bag.list$col[kappa.ellipse$which] <- kappa.ellipse$col
            bag.list$lwd[kappa.ellipse$which] <- kappa.ellipse$lwd
        }
        biplot.legend(legend.type, legend.format, mean.list = mean.list,
            sample.list = sample.list, bag.list = bag.list, class.names =
                              dimnames(G)[[2]],
               quality.print = quality.print, quality = quality)
    }
    list(predictions = predict.mat, quality = quality, adequacy = adequacy,
        axis.predictivity = axis.predictivity, sample.predictivity =
                          sample.predictivity)
}
```

PCAbiplot function

```
#FUNCTION THAT SIMULATES DATA#
##############################

simulation.CPC.CVA<-function (sigma=matrix(c(0.1,0,0,0,0.1,0,0,0,0.1),ncol
    =3),MU=rbind(c(0,0,0),c(1,0,0),c(0,1,1)))
```

```
{
  set.seed (4567)
  require(MASS)
  n1 <- 15
  n2 <- 15
  n3 <- 15
  #sigma=matrix(c(3,2.2,0.7,2.2,2,0,0.7,0,1),ncol=3)

  X11 <- mvrnorm(n1, MU[1,], sigma)
  X12 <- mvrnorm(n2, MU[2,], sigma)
  X13 <- mvrnorm(n3, MU[3,], sigma)

  X21 <- mvrnorm(n1, MU[1,]+2, sigma)
  X22 <- mvrnorm(n2, MU[2,]*5, sigma)
  X23 <- mvrnorm(n3, MU[3,], sigma)

  X31 <- mvrnorm(n1, MU[1,], sigma)
  X32 <- mvrnorm(n2, MU[2,], sigma)
  X33 <- mvrnorm(n3, MU[3,], sigma)

  X41 <- mvrnorm(n1, MU[1,]+2, sigma)
  X42 <- mvrnorm(n2, MU[2,]*5, sigma)
  X43 <- mvrnorm(n3, MU[3,],   sigma)

  X1 <- rbind (X11, X12, X13)
  X2 <- rbind (X21, X22, X23)
  X3 <- rbind (X31, X32, X33)
  X4 <- rbind (X41, X42, X43)
  X <- array(NA, dim=c(nrow(X1),ncol(X1),4))
  X[,,1] <- X1
  X[,,2] <- X2
  X[,,3] <- X3
  X[,,4] <- X4
  X.list <- list (X1, X2, X3, X4)
}
#MATRICIZED MANSOOR DATA#
wide<-do.call("cbind",Xdat)
tall<-do.call("rbind",Xdat)
agg<-0.25*(Xdat[[1]]+Xdat[[2]]+Xdat[[3]]+Xdat[[4]])

#CVA BIPLOT FUNCTION#
CVAbiplot<-function (X, G = NULL, dim.biplot = c(2, 1, 3), e.vects = 1:ncol(X), correlation.biplot = FALSE, classes = 1:ncol(G),
      samples = list(...), ax=list(...), ax.new = NULL, samples.new = NULL, new.samples= list(...), class.means=list(...), predict.means = NULL, predict.samples = NULL,
      alpha.bags=list(...), kappa.ellipse=list(...), density.style=list(...),
      colour.scheme = NULL, Title = NULL, exp.factor = 1.2,
      dim3.plane.col = "lightgrey", dim3.xdiameter = 2, dim3.ydiameter = 2,
      reflect = c(FALSE, "x", "y"), rotate = 0, select.origin = FALSE,
      legend.type=list(...), legend.format=list(...), weightedCVA = c("weighted", "unweightedI", "unweightedCent"),
          adequacies.print = FALSE, output = 1:10, predictivity.print = FALSE, quality.print = FALSE,
      adjust.3d = c(0.5, 0.5),  bag.alpha.3d = 0.7, aspect.3d = "iso", ax.list.3d = "black", cex.3d = 0.6, col.text.3d = "black",
          font.3d = 2, predictions.3D = TRUE, size.ax.3d = 0.5, size.means.3d = 10, size.points.3d = 5, xTitles.3d = c("", "", "Dim 1", "Dim 2", "Dim 3"),
      ID.labs = FALSE, ID.3d = 1:nrow(X), large.scale = FALSE, ort.lty = 1, ...)
```

```
{
# unique default arguments
    dim.biplot <- dim.biplot[1]
  if (dim.biplot != 1 & dim.biplot != 2) stop ("Only 1D and 2D biplots")
  e.vects <- e.vects[1:dim.biplot]
    reflect <- reflect[1]
    weightedCVA <- weightedCVA[1]

# data matrices
    X.info <- biplot.check.X (X, scaled.mat=F)
    X <- X.info$X
    unscaled.X <- X.info$unscaled.X
    means <- X.info$means
  G <- biplot.check.G (G, nrow(X))
  Nmat <- t(G)%*%G
  Xbar <- solve(Nmat) %*% t(G) %*% X
  if (!is.null(samples.new)) X.new <- scale(samples.new, center=means, scale=F) else X.new <- NULL

    n <- nrow(X)
    p <- ncol(X)
    J <- ncol(G)
    K <- min(p, J - 1)
  if (K == 1) {  dim.biplot <- 1
                 e.vects <- e.vects[1] }

    if(!all(is.numeric(classes)))  classes <- match(classes, dimnames(G)[[2]], nomatch = 0)
    classes <- classes[classes <= J]
    classes <- classes[classes > 0]

# MATRIX ALGEBRA
    SSP.T <- t(X) %*% X
    SSP.B <- t(Xbar) %*% Nmat %*% Xbar
    SSP.W <- SSP.T - SSP.B
    Wmat <- SSP.W
    svd.Wmat <- svd(Wmat)
    lambdamatI <- diag(svd.Wmat$d)
    Lmat <- svd.Wmat$u %*% solve(sqrt(lambdamatI))
    if (weightedCVA == "weighted") Cmat <- Nmat
    if (weightedCVA == "unweightedI") Cmat <- diag(J)
    if (weightedCVA == "unweightedCent") Cmat <- diag(J) - matrix(1/J, nrow = J, ncol = J)
    if (is.na(match(weightedCVA, c("weighted", "unweightedI", "unweightedCent")))) stop(" Argument 'weightedCVA' must be one of 'weighted','unweightedI','unweightedCent' ")
    svd.step2 <- svd(t(Lmat) %*% t(Xbar) %*% Cmat %*% Xbar %*% Lmat)
    Vmat <- svd.step2$v
    lambdamat <- diag(svd.step2$d)
    svd.2sided <- Eigen.twosided(t(Xbar) %*% Cmat %*% Xbar, Wmat)
    Mmat <- svd.2sided$W
    lambdamat.2sided <- svd.2sided$Lambda.mat
    vec.temp <- rep(0, p)
    vec.temp[e.vects] <- 1
    Jmat <- diag(vec.temp)
    XLVJ <- X %*% Mmat %*% Jmat
    XbarHat <- Xbar %*% Mmat %*% Jmat %*% solve(Mmat)
    XHat <- XLVJ %*% solve(Mmat)
  I.min.H <- (diag(n) - G %*% (solve(Nmat)) %*% t(G))

# Fit measures
# args(biplot.fit.measures) function (mat, mat.hat, weights = diag(nrow(mat)), orthog.metric = diag(ncol(mat)), eigenvals, eigenvecs, dims)

  fit.Canvar <- biplot.fit.measures (Xbar, XbarHat, weights=Cmat, orthog.metric = Wmat, eigenvals=diag(lambdamat.2sided), eigenvecs=Mmat, dims=e.vects)
    fit.Within <- biplot.fit.measures (I.min.H%*%X, I.min.H%*%XHat, orthog.metric = Wmat, eigenvals=diag(lambdamat.2sided), eigenvecs=Mmat, dims=e.vects)
  quality.Canvar <- fit.Canvar$quality
  quality.Origvar <- sum((diag(solve(t(Mmat)%*%Mmat)) * diag(lambdamat.2sided))[e.vects]) / sum(diag(solve(t(Mmat)%*%Mmat)) * diag(lambdamat.2sided))
  adequacy <- fit.Canvar$adequacy

  axis.predictivity <- fit.Canvar$axis.predictivity
  class.predictivity <- fit.Canvar$item.predictivity
  within.class.axis.predictivity <- fit.Within$axis.predictivity
  within.class.sample.predictivity <- fit.Within$item.predictivity

    if (adequacies.print & predictivity.print) stop("adequacies.print and predictivity.print cannot both be set to True")
    if (adequacies.print) dimnames(X)[[2]] <- paste(dimnames(X)[[2]], " (", adequacy, ")", sep = "")
    if (predictivity.print) dimnames(X)[[2]] <- paste(dimnames(X)[[2]], " (", round(axis.predictivity, digits = 2), ")", sep = "")

# reflect only for 1D & 2D biplots
    reflect.mat <- diag(dim.biplot)
  if (reflect == "x" & dim.biplot < 3) reflect.mat[1,1] <- -1
  if (reflect == "y" & dim.biplot == 2) reflect.mat[2,2] <- -1
  if (reflect == "xy" & dim.biplot == 2) reflect.mat[1:2,1:2] <- diag(-1,2)

# rotate only for 2D biplots
    rotate.mat <- diag(dim.biplot)
    if (dim.biplot == 2)
  {
    if (!is.null(ax$rotate))
      {
      if (is.numeric(ax$rotate))
        {
           radns <- pi * rotate/180
           rotate.mat <- matrix(c(cos(radns), -sin(radns), sin(radns), cos(radns)), ncol = 2)
      }
      else
            {
              if (ax$rotate == "maxpred")
           {
                 ax$rotate <- (names(axis.predictivity))[axis.predictivity == max(axis.predictivity)]
                 ax$rotate <- match(ax$rotate, dimnames(X)[[2]])
               }
              else ax$rotate <- match(ax$rotate, dimnames(X)[[2]])
              radns <- -atan2(V.mat[ax$rotate, e.vects[2]], V.mat[ax$rotate, e.vects[1]])
              rotate.mat <- matrix(c(cos(radns), -sin(radns), sin(radns), cos(radns)), ncol = 2)
            }
    }
  }
    Mr <- Mmat[,e.vects, drop=F]
```

```r

# samples and means
  Z.new <- NULL
 Z.means.mat <- NULL
  Z <- X %*% Mr %*% rotate.mat %*% reflect.mat
  Z.means.mat <- Xbar %*% Mr %*% rotate.mat %*% reflect.mat
  if (!is.null (X.new)) Z.new <- X.new %*% Mr %*% rotate.mat %*% reflect.mat
 dimnames(Z) <- list (dimnames(X)[[1]], NULL)
 if (!is.null(X.new)) if (is.null(dimnames(samples.new)[[1]])) dimnames(Z.new) <- list (paste("N",1:nrow(Z.new),sep=""),NULL) else dimnames(Z.new) <- list (dimnames(samples.new)[[1]], NULL)

  if (is.matrix(ax.new)) { NewVarsMeans <- apply(ax.new, 2, mean)
                           NewVars.cent <- scale(ax.new, center = TRUE, scale = FALSE)
                           NewVars.means <- solve(Nmat) %*% t(G) %*% NewVars.cent
                          }

# axes / variables
  num.vars <- p
 var.names <- dimnames(X)[[2]]
 Mrr <- solve(Mmat)[e.vects, ,drop=F]
  if (!is.null(ax.new)) {  means <- c(means, NewVarsMeans)
                           unscaled.X <- cbind(unscaled.X, ax.new)
                           num.vars <- ncol(unscaled.X)
        if (!is.null(dimnames(ax.new)[[2]])) var.names <- c(var.names, dimnames(ax.new)[[2]]) else var.names <- c (var.names, paste("NV",1:ncol(ax.new),sep=""))
                           LambdaMinOne <- ifelse(lambdamat < 1e-10, 0, 1/lambdamat)
                           Mr.new <- LambdaMinOne %*% t(Mmat) %*% t(Xbar) %*% Cmat %*% NewVars.means
                           Mr.new <- Mr.new[, e.vects, drop=F]
                           Mrr.all <- rbind(Mrr, Mr.new)
                         }
 else Mrr.all <- Mrr

  ax <- do.call("biplot.ax.control", c(num.vars,list(var.names),ax))
  if (ax$type == "prediction") axes.direction <- solve(diag(diag(t(Mrr.all) %*% Mrr.all))) %*% t(Mrr.all) %*% rotate.mat %*% reflect.mat
 else axes.direction <- Mr %*% rotate.mat %*% reflect.mat

 if (length(ax$which)==0) z.axes <- NULL
  else z.axes <- lapply(1:length(ax$which), calibrate.axis, unscaled.X, means, sd=rep(1,length(means)), axes.direction, ax$which, ax$ticks, ax$orthogx, ax$orthogy, ax$oblique)

# alpha-bags
  alpha.bags <- do.call("biplot.alpha.bag.control", c(J,list(dimnames(G)[[2]]),alpha.bags))
  z.bags <- vector ("list", length(alpha.bags$which))
  if (length(alpha.bags$which) > 0)
  for (j in 1:length(alpha.bags$which))
  {
     class.num <- alpha.bags$which[j]
     mat <- Z[G[,class.num]==1,]
     flush.console()
     cat (paste ("alpha bag for class ", dimnames(G)[[2]][class.num], " with ", nrow(mat), " samples", sep = ""), "\n")
```

```
       if (dim.biplot==2) z.bags[[j]] <- calc.alpha.bags (mat, alpha.bags$alpha[j], alpha.bags$max[j], alpha.bags$Tukey.median[j], alpha.bags$min[j])
       if (dim.biplot==1) z.bags[[j]] <- quantile(mat, c((100-alpha.bags$alpha[j])/200,1-(100-alpha.bags$alpha[j])/200,0.5))
    }

# kappa-ellipse
    kappa.ellipse <- do.call("biplot.kappa.ellipse.control", c(J,list(dimnames(G)[[2]]),dim.biplot,kappa.ellipse))
    z.ellipse <- vector ("list", length(kappa.ellipse$which))
    if (length(kappa.ellipse$which) > 0)
    for (j in 1:length(kappa.ellipse$which))
    {
       class.num <- kappa.ellipse$which[j]
       mat <- Z[G[,class.num]==1,]
       if (dim.biplot==2) z.ellipse[[j]] <- calc.concentration.ellipse (mat, kappa.ellipse$kappa[j])
       if (dim.biplot==1) z.ellipse[[j]] <- qnorm(c(1-pnorm(kappa.ellipse$kappa[j]),pnorm(kappa.ellipse$kappa[j])),mean(mat),sqrt(var(mat)))
    }

# density plots
    if (dim.biplot==1)
    {  density.style <- do.call("biplot.density.1D.control", c(J,list(dimnames(G)[[2]]),density.style))
       z.density <- vector ("list", length(density.style$which))
       if (length(density.style$which) > 0)
       for (j in 1:length(density.style$which))
        {
        class.num <- density.style$which[j]
        mat <- Z[G[,class.num]==1,]
        z.density[[j]] <- density (mat, bw=density.style$bw[j], kernel=density.style$kernel[j])
      }
    }
  if (dim.biplot==2)
    {  density.style <- do.call("biplot.density.2D.control", c(J,list(dimnames(G)[[2]]),density.style))
       if (!is.null(density.style$which))
       {
         if (density.style$which==0) mat <- Z
         else mat <- Z[G[,density.style$which]==1,]

             x.range <- range(Z[,1])
       y.range <- range(Z[,2])
       width <- max(x.range[2]-x.range[1],y.range[2]-y.range[1])
       xlim <- mean(Z[,1])+c(-1,1)*0.75*width
       ylim <- mean(Z[,1])+c(-1,1)*0.75*width
           if (is.null(density.style$h))
           z.density <- kde2d (mat[,1], mat[,2], n = density.style$n, lims = c(xlim, ylim))
         else
           z.density <- kde2d (mat[,1], mat[,2], h = density.style$h, n = density.style$n, lims = c(xlim, ylim))
      }
     else z.density <- NULL
    }

# allows for changing the colour palette
    if (!is.null(colour.scheme)) {
```

```
   my.sample.col <- colorRampPalette(colour.scheme)
   samples$col <- my.sample.col(samples$col)
 }

 samples <- do.call("biplot.sample.control", c(J,samples))
 new.samples <- do.call("biplot.new.sample.control", c(min(1,nrow(X.new)),new.samples))
 class.means <- do.call("biplot.mean.control", c(J,list(dimnames(G)[[2]]),class.means))
if (length(class.means$which)==0) {  class.means$which <- 1:J
                                   class.means$col <- samples$col
           class.means <- do.call("biplot.mean.control", c(J,list(dimnames(G)[[2]]),class.means))
                 }
 legend.format <- do.call("biplot.legend.control", legend.format)
 legend.type <- do.call("biplot.legend.type.control", legend.type)

 if (dim.biplot==2) draw.biplot (Z=Z, G=G, classes=classes, Z.means=Z.means.mat, z.axes=z.axes, z.bags=z.bags, z.ellipse=z.ellipse, Z.new=Z.new, Z.density=z.density,
                  sample.style=samples, mean.style=class.means, ax.style=ax, bag.style=alpha.bags, ellipse.style=kappa.ellipse, new.sample.style=new.samples, density.style=density.style,
          predict.samples=predict.samples, predict.means=predict.means, Title=Title, exp.factor=exp.factor, ...)
if (dim.biplot==1) draw.biplot.1D (Z=Z, G=G, classes=classes, Z.means=Z.means.mat, z.axes=z.axes, z.bags=z.bags, z.ellipse=z.ellipse, Z.new=Z.new, Z.density=z.density,
                  sample.style=samples, mean.style=class.means, ax.style=ax, bag.style=alpha.bags, ellipse.style=kappa.ellipse, new.sample.style=new.samples, density.style=density.style,
          predict.samples=predict.samples, predict.means=predict.means, Title=Title, exp.factor=exp.factor, ...)


 if (!is.null(ax$oblique) & ax$type == "interpolation") points(0, 0, pch = "+", cex = 2)

 if (!is.null(predict.samples)) predict.mat <- scale(Z[predict.samples,,drop=F] %*% t(reflect.mat) %*% t(rotate.mat) %*% Mrr, center=-means, scale=F) else predict.mat <- NULL
if (!is.null(predict.means)) predict.mat <- rbind (predict.mat, scale(Z.means.mat[predict.means,,drop=F] %*% t(reflect.mat) %*% t(rotate.mat) %*% Mrr, center=-means, scale=F))

if (!is.null(predict.mat)) dimnames(predict.mat) <- list(c(dimnames(X)[[1]][predict.samples],dimnames(G)[[2]][predict.means]), dimnames(X)[[2]])

if (any(unlist(legend.type)))
  {
     windows()
  sample.list <- list(pch = samples$pch, col = samples$col)
  mean.list=list(pch=rep(NA,J), col=rep(NA,J))
  mean.list$pch[class.means$which] <- class.means$pch
  mean.list$col[class.means$which] <- class.means$col
  bag.list=list(lty=rep(1,J), col=rep(NA,J), lwd=rep(NA,J))
  bag.list$lty[alpha.bags$which] <- alpha.bags$lty
```

```
      bag.list$col[alpha.bags$which] <- alpha.bags$col
      bag.list$lwd[alpha.bags$which] <- alpha.bags$lwd
      if (length(alpha.bags$which)==0 & length(kappa.ellipse$which)>0)
        {
          bag.list$lty[kappa.ellipse$which] <- kappa.ellipse$lty
          bag.list$col[kappa.ellipse$which] <- kappa.ellipse$col
          bag.list$lwd[kappa.ellipse$which] <- kappa.ellipse$lwd
        }
            biplot.legend (legend.type, legend.format, mean.list=mean.list,
                sample.list=sample.list, bag.list=bag.list, class.names=
                dimnames(G)[[2]], quality.print=quality.print, quality=
                quality)
        }
    list (predictions=predict.mat, quality.Canvar=quality.Canvar, quality.
        Origvar=quality.Origvar, adequacy=adequacy, axis.predictivity=axis.
        predictivity, class.predictivity=class.predictivity,
        within.class.axis.predictivity = within.class.axis.predictivity,
            within.class.sample.predictivity = within.class.sample.
            predictivity)

}
```

CVAbiplot function

```

#PROCRUSTES ALGORITHM#
function (X,iso,Xdat)
{
  sum<-0
  Q<-vector("list",4)
  s<-vector("list",4)
  k<-0

 tr<-function(X)
 {
  return(sum(diag(X)))
 }

  for(i in 1:4)
  {
    Q[[i]]<-matrix(c(1,0,0,1), nrow = 2, ncol = 2, byrow= T)
    G<-G+X[[i]]%*%Q[[i]]
    k<-k+tr(X[[i]])
  }


  for (i in 1:4)
  {
   if (iso)
   {
     s[[i]]<-((0.25*k)/tr(G%*%t(G)))*tr(t(G)%*%X[[i]]%*%Q[[i]])/tr((X[[i]]%*
         %Q[[i]])%*%t(X[[i]]%*%Q[[i]]))
     sum<-sum+tr((s[[i]]*X[[i]]%*%Q[[i]]-G)%*%t(s[[i]]*X[[i]]%*%Q[[i]]-G))
   }
  else sum<-sum+tr((X[[i]]%*%Q[[i]]-G)%*%t(X[[i]]%*%Q[[i]]-G))
  }
 if (iso)
  totcrit<-4*sum
 else totcrit<-sum
 newcrit<-0
 sum<-0
 while (totcrit!=newcrit)
 {
   newcrit<-totcrit
   if (iso)
   {
    G<-0.25*(s[[1]]*X[[1]]%*%Q[[1]]+s[[2]]*X[[2]]%*%Q[[2]]+s[[3]]*X[[3]]%*%Q[[3]]+s[[4]]*X[[4]]%*%Q[[4]])
   for (i in 1:4)
   {
     Q[[i]]<-svd(t(G)%*%X[[i]])$v%*%t(svd(t(G)%*%X[[i]])$u)
     s[[i]]<-((0.25*k)/tr(G%*%t(G)))*tr(t(G)%*%X[[i]]%*%Q[[i]])/tr((X[[i]]%*%Q[[i]])%*%t(X[[i]]%*%Q[[i]]))
     sum<-sum+tr((s[[i]]*X[[i]]%*%Q[[i]]-G)%*%t(s[[i]]*X[[i]]%*%Q[[i]]-G))
   }
  totcrit<-4*sum
  } else
  {
   G<-0.25*(X[[1]]%*%Q[[1]]+X[[2]]%*%Q[[2]]+X[[3]]%*%Q[[3]]+X[[4]]%*%Q[[4]])
   for (i in 1:4)
   {
     Q[[i]]<-svd(t(G)%*%X[[i]])$v%*%t(svd(t(G)%*%X[[i]])$u)
     sum<-sum+tr((X[[i]]%*%Q[[i]]-G)%*%t(X[[i]]%*%Q[[i]]-G))
   }
   totcrit<-sum
   sum<-0
  }
 }

 result<-list(s,Q)
 return(result)
}

#APPLYING GOPA METHOD#
function (Xdat,Proc.type="sample",iso=F)
{
####################
#DATA PREPROCESSING#
####################
n<-29
p<-7
k<-4
threeway<-array(0,dim=c(n,p,k))
for (i in 1:4)
 threeway[,,i]=as.matrix(Xdat[[i]])
meansk<-vector("list",4)

for (i in 1:k)
  meansk[[i]]<-apply(threeway[,,i],2,mean)
overall.mean <- rep(NA,p)
overall.sd <- rep(NA,p)
for (i in 1:p)
  {
   overall.mean[i] <- mean(threeway[,i,])
   overall.sd[i] <- sd(as.vector(threeway[,i,]))
   threeway[,i,]<-(threeway[,i,]-mean(threeway[,i,]))/sd(as.vector(threeway[,i,]))
  }

XDat<-vector("list",4)
for (i in 1:4)
 #XDat[[i]]<-as.matrix(Xdat[[i]])
 XDat[[i]]=as.matrix(threeway[,,i])

 X.new<-vector("list",4)
for (i in 1:k)
 X.new[[i]]<- matrix(as.matrix(0-meansk[[i]]),nrow=1)

#DEFINING GROUP INDICATOR MATRICES#
G1<-G2<-G3<-G4<-matrix(0,29,3)
p<-7
G1[1:3,1]<-G1[4:13,2]<-G1[14:29,3]<-1
G2[1:3,1]<-G2[4:13,2]<-G2[14:29,3]<-1
G3[1:3,1]<-G3[4:13,2]<-G3[14:29,3]<-1
G4[1:3,1]<-G4[4:13,2]<-G4[14:29,3]<-1
dimnames(XDat[[1]])<-dimnames(XDat[[2]])<-dimnames(XDat[[3]])<-dimnames(XDat[[4]]) <- list(paste(1:nrow(XDat[[1]])), paste("V", 1:p, sep = ""))

#DEFINING SAMPLE POINT AND VARIABLE POINT CONFIGURATIONS#
outt<-lapply(1:k,function(j) {PCAbiplot2(XDat[[j]], samples.new=matrix(X.new[[j]],nrow=1),Title="PCA of X1",new.samples=list(col="red",pch=16), samples=list(col=3))})
Zk<-lapply(outt,function(x) {x$Z})
Vk<-lapply(outt,function(x) {x$Vr})
Z.new<-lapply(outt,function(x) {x$Z.new})
both<-lapply(1:k,function(x) {rbind(Vk[[x]],Zk[[x]])})
lambda.r<-lapply(outt,function(x) {x$lambda.r})

#DETERMING ROTATION MATRICES FROM GOPA#
if (Proc.type=="both")
  {
     estimates<-proc(Zk,iso=F)
     Q<-estimates$Q
     Z <- lapply(1:k,function(x){rbind(Zk[[x]]%*%Q[[x]])})
     z.ax <-lapply(1:k,function(x){rbind(Vk[[x]]%*%Q[[x]])})
  }
if (Proc.type=="sample")
  {
     estimates<-proc(Zk,iso=F)
     Q<-estimates$Q
     s.vec<-estimates$s.vec
     Z<-lapply(1:k,function(x){rbind(Zk[[x]]%*%Q[[x]])})
     z.ax<-lapply(1:k,function(x){rbind(Vk[[x]]%*%Q[[x]])})
  }
if (Proc.type=="ax")
  {
    estimates<-proc(Zk,iso=F)
    Q<-estimates$Q
    s.vec<-estimates$s.vec
    Z<- lapply(1:k,function(x){rbind(Zk[[x]]%*%Q[[x]])})
    z.ax<-lapply(1:k,function(x){rbind(Vk[[x]]%*%Q[[x]])})
  }

  z.axes<-lapply(1:k, function(x) {outt[[x]]$z.axes})
  Z.new<-lapply(1:k,function(x) {Z.new[[x]]%*%Q[[x]]})

#ROTATING THE AXES#
   for(i in 1:4)
   {
    for (j in 1:7)
     outt[[i]]$z.axes[[j]][,1:2] <- outt[[i]]$z.axes[[j]][,1:2]%*%Q[[i]]
   }

#OPT SO THAT AXES PASS THROUGH INTERPOLATED POINT#
```

```
z.axes.1.OPT <- lapply(1:7, calibrate.axis, unscaled.X=do.call("rbind",Xdat), means=meanscaled, sd=overall.sd, axes.rows=(1/(diag(Vk[[1]] %*% t(Vk[[1]]))) * Vk[[1]])%*%Q[[1]], ax.which=1:7, ax.tickvec=outt[[1]]$ax.style$ticks, ax.orthogxvec=rep(Z.new[[1]][1],7), ax.orthogyvec=rep(Z.new[[1]][2],7),ax.oblique=out1$ax.style$oblique)
z.axes.2.OPT <- lapply(1:7, calibrate.axis, unscaled.X=do.call("rbind",Xdat), means=meanscaled, sd=overall.sd, axes.rows=(1/(diag(Vk[[2]] %*% t(Vk[[2]]))) * Vk[[2]])%*%Q[[2]], ax.which=1:7, ax.tickvec=outt[[2]]$ax.style$ticks, ax.orthogxvec=rep(Z.new[[2]][1],7), ax.orthogyvec=rep(Z.new[[2]][2],7),ax.oblique=out1$ax.style$oblique)
z.axes.3.OPT <- lapply(1:7, calibrate.axis, unscaled.X=do.call("rbind",Xdat), means=meanscaled, sd=overall.sd, axes.rows=(1/(diag(Vk[[3]] %*% t(Vk[[3]]))) * Vk[[3]])%*%Q[[3]], ax.which=1:7, ax.tickvec=outt[[3]]$ax.style$ticks, ax.orthogxvec=rep(Z.new[[3]][1],7), ax.orthogyvec=rep(Z.new[[3]][2],7), ax.oblique=out1$ax.style$oblique)
z.axes.4.OPT <- lapply(1:7, calibrate.axis, unscaled.X=do.call("rbind",Xdat), means=meanscaled, sd=overall.sd, axes.rows=(1/(diag(Vk[[4]] %*% t(Vk[[4]]))) * Vk[[4]])%*%Q[[4]], ax.which=1:7, ax.tickvec=outt[[4]]$ax.style$ticks, ax.orthogxvec=rep(Z.new[[4]][1],7), ax.orthogyvec=rep(Z.new[[4]][2],7),ax.oblique=out1$ax.style$oblique)

out1<-outt[[1]]
out2<-outt[[2]]
out3<-outt[[3]]
out4<-outt[[4]]

#BINDING ROTATED AND TRANSLATED SAMPLES FOR K OCC INTO ONE#
p <- 7
Z <- rbind (out1$Z%*%Q[[1]]-matrix(1,nrow=n1,ncol=1)%*%Z.new[[1]], out2$Z%*%Q[[2]]-matrix(1,nrow=n2,ncol=1)%*%Z.new[[2]],out3$Z%*%Q[[3]]-matrix(1,nrow=n2,ncol=1)%*%Z.new[[3]],out4$Z%*%Q[[4]]-matrix(1,nrow=n2,ncol=1)%*%Z.new[[4]])
print(Z)
z.axes <- vector("list",4*p)

#MOVING AXES TO NEW ORIGIN AT INTERPOLATED POINT#
for (j in 1:p)
   {  z.axes[[j]] <- z.axes.1.OPT[[j]]
      z.axes[[j]][,1:2] <- z.axes[[j]][,1:2] - matrix(1,nrow=nrow(z.axes.1.OPT[[j]]),ncol=1)%*%Z.new[[1]]
      z.axes[[j+p]] <- z.axes.2.OPT[[j]]
      z.axes[[j+p]][,1:2] <- z.axes[[j+p]][,1:2] - matrix(1,nrow=nrow(z.axes.2.OPT[[j]]),ncol=1)%*%Z.new[[2]]
      z.axes[[j+2*p]] <- z.axes.3.OPT[[j]]
      z.axes[[j+2*p]][,1:2] <- z.axes[[j+2*p]][,1:2] - matrix(1,nrow=nrow(z.axes.3.OPT[[j]]),ncol=1)%*%Z.new[[3]]
      z.axes[[j+3*p]] <- z.axes.4.OPT[[j]]
      z.axes[[j+3*p]][,1:2] <- z.axes[[j+3*p]][,1:2] - matrix(1,nrow=nrow(z.axes.4.OPT[[j]]),ncol=1)%*%Z.new[[4]]
  }

#DEFINING COLLECTIVE GROUP INDICATOR MATRIX#
G<-matrix(0,29*4,3*4)
for (i in 1:4)
{
 a<-1+29*(i-1)
 b<- 29*i
 c<-1+3*(i-1)
 d<-3*i
 G[a:b,c:d]<-G1
}
   out$G <- G
   out$classes <- 1:12
   out$Z.new <- NULL


#DEFINING PLOT SPECIFICATIONS#
for (i in 1:4)
{

 a<-1+3*(i-1)
 b<-3*i
 c<-2+3*(i-1)
 out$sample.style$col[a:b]<-i
 out$sample.style$pch[a]<-15
 out$sample.style$pch[b]<-16
 out$sample.style$pch[c]<-17
}
   J<-12
   out$sample.style$cex <- rep(1,12)
   out$sample.style$label <- rep(F,12)
   out$ax.style$which <- 1:28
   out$ax.style$lwd <- rep(1,28)
   out$ax.style$lty <- rep(1,28)
   out$ax.style$label.cex <- rep(0.75,28)
   out$ax.style$label.dist <- rep(0,28)
   out$ax.style$tick.size <- rep(1,28)
   out$ax.style$tick.label <- rep(T,28)
   out$ax.style$tick.label.cex <- rep(0.6,28)
   out$ax.style$tick.label.side <- rep("left",28)
   out$ax.style$tick.label.offset <- rep(0.5,28)
   out$ax.style$tick.label.pos <- rep(1,28)
   lab<-labs<-c(rep(0,28))
  i<-0
for (k in 1:4)
 for (j in 1:7)
 {
  i<-i+1
  lab[i]<-paste(j,k,sep=".")
  }

for (i in 1:28)
 labs[i]<-paste("X",lab[i],sep="")
out$ax.style$names <- labs

out$ax.style$col[8:14]<-out$ax.style$tick.col[8:14]<-out$ax.style$tick.label.col[8:14]<-out$ax.style$label.col[8:14]<-"brown"
out$ax.style$col[15:21]<-out$ax.style$tick.col[15:21]<-out$ax.style$tick.label.col[15:21]<-out$ax.style$label.col[15:21]<-"black"
out$ax.style$col[22:28]<-out$ax.style$tick.col[22:28]<-out$ax.style$tick.label.col[22:28]<-out$ax.style$label.col[22:28]<-"orange"

draw.biplot(Z = Z, G = out$G, classes = out$classes, Z.means = out$Z.means, z.axes = z.axes, z.bags = out$z.bags, z.ellipse = out$z.ellipse, Z.new = out$Z.new, Z.density = out$Z.density,sample.style = out$sample.style, mean.style = out$mean.style, ax.style = out$ax.style, bag.style = out$bag.style, ellipse.style = out$ellipse.style, new.sample.style = out$new.sample.style,density.style = out$density.style, predict.samples = out$predict.samples, predict.means = out$predict.means, Title = "Combined biplot", exp.factor = out1$exp.factor)
```

Chapter 6 code

```
#FG ALGORITHM FOR CPC#
######################

#F ALGORITHM#
function(A.mat, n.vec, B)
{
PHI <- function(F.mat, B, n.vec)
{
phi <- function(mat)
{
det1 <- prod(eigen(diag(diag(mat)))$values)
det2 <- prod(eigen(mat)$values)
return(det1/det2)
}
k <- length(F.mat)
prod(sapply(1:k, function(x, F.mat, B, n.vec, phi)
phi(t(B) %*% F.mat[[x]] %*% B)^n.vec[x], F.mat = F.mat,B = B, n.vec = n.vec,
     phi = phi))
}
p <- nrow(A.mat[[1]])
k <- length(A.mat)

T.mat <- lapply(1:k, function(x)
matrix(NA, ncol = 2, nrow = 2))
f.herh <- 0
klaar <- F
max.herh <- 100
while(!klaar) {
# --  Step F1
Bf <- B
f.herh <- f.herh + 1#
# --  Step F2
for(m in 1:(p - 1))
for(j in (m + 1):p) {
#  --  Step F21
for(i in 1:k)
  T.mat[[i]] <- t(B[, c(m, j)])%*% A.mat[[i]] %*% B[, c(m, j)]
#  --  Step F22
J22 <- G.al2(T.mat, n.vec)
#  --  Step F23
J <- diag(p)
J[m, m] <- J22[1, 1]
J[j, j] <- J22[2, 2]
J[m, j] <- J22[1, 2]
J[j, m] <- J22[2, 1]
B <- B %*% J
}
#  --  Step F3
if(abs(PHI(A.mat, Bf, n.vec) - PHI(A.mat, B, n.vec)) < 1e-005)
klaar <- T
if(f.herh > max.herh)
klaar <- T
}
list(Bf=B, ff=f.herh)
}

#G ALGORITHM#
function(T.mat, n.vec)
{
k <- length(T.mat)
#  --  Step G0
Q <- diag(2)
g.herh <- 0
max.herh <- 20
klaar <- F
alpha.mat <- matrix(0, ncol = 2, nrow = 2)
while(!klaar) {
#  --  Step G1
Qg <- Q
g.herh <- g.herh + 1
# --  Step G2
delta <- sapply(1:k, function(x, T.mat, Q)
diag(t(Q) %*% T.mat[[x]] %*% Q), T.mat = T.mat, Q = Q)
TT <- matrix(0, nrow = 2, ncol = 2)
for(i in 1:k)
TT <- TT + n.vec[i] * ((delta[1, i] - delta[2,i])/(delta[1, i] * delta[2, i
    ])) * T.mat[[i]]
#  --  Step G3
Q <- svd(TT)$u
if(acos(Q[1, 1]) > acos(cos(pi/4)))
Q[, 1] <- -1 * Q[, 1]
#  --  Step G4
if(max(abs(Qg - Q)) < 1e-005)
klaar <- T
if(g.herh > max.herh)
 klaar <- T
 }
 return(Q)
}

#FG ALGORITHM#
function(A.mat, n.vec)
{
p <- nrow(A.mat[[1]])
k<- length(A.mat)
p<-7
k<-4
B <- diag(p)
F.out <- F.al2(A.mat, n.vec, B)
B <- F.out$Bf
f.herh <- F.out$ff
if(f.herh == 1) {
for(m in 1:(p - 1))
for(j in 1:p) {
G1 <- 0.8 * B[m, j] + 0.6 * B[m + 1, j
  ]
G2 <- 0.8 * B[m + 1, j] - 0.6 * B[m, j
  ]
B[m, j] <- G1
B[j, m] <- G2
}
F.out <- F.al2(A.mat, n.vec, B)
B <- F.out$B
f.herh <- c(f.herh, F.out$f.herh)
}
print(t(B)%*%A.mat[[1]]%*%B)
list(V=B, f=f.herh)
}

#FG ALGORITHM FOR DCPC#
######################

#F ALGORITHM#

```

```
function(Smat,p,k)
{

PHI<-function(F.mat,B,p,k)
 {
  diagonal<-matrix(NA,k*p,k*p)
  F<-matrix(NA,k*p,k*p)
    for (i in 1:k)
      for (j in 1:k)
      {
        q<-(i-1)*p+1
        r<-i*p
        s<-(j-1)*p+1
        t<-j*p
        F[q:r,s:t]<-t(B)%*%F.mat[q:r,s:t]%*%B
        diagonal[q:r,s:t]<- diag(t(B)%*%F.mat[q:r,s:t]%*%B)
      }
  det1<-prod(eigen(diagonal)$values)
  det2<-prod(eigen(F)$values)
  return(as.real(det1/det2))
 }

#--Step F0
   f.iter<-0
        B<-diag(p)
        klaar<-FALSE
        while (!klaar)
        {
#--Step F1
         Bf<-B
         f.iter<-f.iter+1
#--Step F2
         for (m in 1:(p-1))
          for (l in (m+1):p)
          {
#--Step F21
            H<-B[,c(m,l)]
            Ik<-diag(k)

#--Step F22
            Tmat<-t(Ik%x%H)%*%Smat%*%(Ik%x%H)
#--Step F23
            J<-g.algd(Tmat,p,k)
#--Step F24
            H_new<-H%*%J
            B[,m]=H_new[,1]
            B[,l]=H_new[,2]
          }
#--Step F3
print(dim(B))
 print(as.real(abs(PHI(Smat,B,p,k)-PHI(Smat,Bf,p,k))))
 if (as.real(abs(PHI(Smat,B,p,k)-PHI(Smat,Bf,p,k)))<0.00001)
      klaar<-TRUE
 }
return(B)
}

#G ALGORITHM FR DCPC#
function(Tmat,p,k)
{
#--Step G0
 g.iter<-0
```

```
       Q<-diag(2)
       done1<-FALSE
while (!done1)
{
#--Step G1
       Qg<-Q
       g.iter<-g.iter+1
       T_new<-matrix(0,2,2)
#--Step G2
       Ik<-diag(k)
       M<-lapply(1:2,function(x) {t(Ik%x%as.matrix(Q[,x]))%*%Tmat%*%(Ik%x%as.matrix(Q[,x]))})
#--Step G3
       A<-solve(M[[1]])-solve(M[[2]])
       for (i in 1:k)
          for (j in 1:k)
          {
         n<-(i-1)*2+1
         o<-i*2
            f<-(j-1)*2+1
            d<-j*2
        T_new<-T_new+A[i,j]*Tmat[n:o,f:d]
          }
#--Step G4
if (T_new[1,2] != 0)
{
 ratio<-(T_new[2,2]-T_new[1,1])/T_new[1,2]
 discr<-sqrt(ratio^2+4)
 root1<-0.5* (ratio + discr)
 root2<-0.5* (ratio - discr)
 if (atan(root1)<= pi/4)
   d<-root1
 else d<-root2
 c<-1/sqrt(1+d^2)
 s<-d*c
} else
 {
   c=1
   s=0
  }

Q[1,1]<-c
Q[1,2]<--s
Q[2,1]<-s
Q[2,2]<-c
#--Step G5
 if(abs(max(Q-Qg))<0.00001)
         done1<-TRUE
}
return(Q)
}

#FG ALGORITHM FOR DCPC#
function(Smat,p,k)
{
  B<-f.algd(Smat,p,k)
  return(B)
}


#MODIFICATION TO PCABIPLOT FUNCTION#

```

```
#DO DCPC AND CHOOSE COMPONENTS#
Bd<-fg.algd(Smat,7,4)
p<-7
k<-4
larray<-array(NA,c(p,p,k*k))
l<-0
total<-0
for (i in 1:k)
 for (j in 1:k)
 {
   l<-l+1
   a<-1+p*(i-1)
   b<-i*p
   c<-1+p*(j-1)
   d<-p*j
   larray[,,l]<-t(Bd)%*%Smat[a:b,c:d]%*%Bd
   total<-total+sum(diag(larray[,,l]))
 }
combin<-combn(7,2)
u<-length(combin)/2
fit<-vector("integer",length=u)
for (i in 1:u)
fit[i]<-(sum(larray[combin[1,i],combin[1,i],1:16])+sum(larray[combin[2,i],combin[2,i],1:16]))/total

index_max<-which(fit==max(fit), arr.ind=T)

Br<-Bd[,c(combin[1,index_max],combin[2,index_max])]

#DO THE CPC AND CHOOSE COMPONENTS #

Bd<-FG.al2(Smat,c(1,1,1,1))$V
print(lapply(1:4,function(i) t(Bd)%*%Smat[[i]]%*%Bd))
p<-7
k<-4
larray<-array(NA,c(p,p,k))
l<-0
total<-0
for (i in 1:k)
{
   larray[,,i]<-t(Bd)%*%as.matrix(Smat[[i]])%*%Bd
   total<-total+sum(diag(larray[,,i]))
 }
combin<-combn(7,2)
u<-length(combin)/2
fit<-vector("integer",length=u)
for (i in 1:u)
fit[i]<-(sum(larray[combin[1,i],combin[1,i],1:4])+sum(larray[combin[2,i],combin[2,i],1:4]))/total

index_max<-which(fit==max(fit), arr.ind=T)
print(combin)
print(index_max)
print(max(fit))
Br<-Bd[,c(combin[1,index_max],combin[2,index_max])]
Xscaled<-lapply(1:4,function(i) scale(Xdat[[i]],center=T,scale=T))
X<-do.call("rbind",Xdat)
#IN PCAbipl, Br REPLACES Vr AND X IS DEFINED AS ABOVE#

#FOR MULTIPLE AXES#
#THIS IS INSERTED BEFORE THE CALIBRATION PROCEDURE#
num.vars <- 28
```

```
var.names <- rep(paste("V",1:7,sep=""),4)
axes.direction<-rbind(axes.direction,axes.direction,axes.direction,axes.
    direction)
unscaled.X2<-cbind(unscaled.X,unscaled.X,unscaled.X,unscaled.X)
means=as.vector(c(colMeans(Xdat[[1]]),colMeans(Xdat[[2]]),colMeans(Xdat
    [[3]]),colMeans(Xdat[[4]])))
ax$ticks<-rep(2,28)

#INSERT BEFORE DRAW BIPLOT AFTER CALIBRATION#
for (i in 1:4)
{

 a<-1+3*(i-1)
 b<-3*i
 c<-2+3*(i-1)
 samples$col[a:b]<-i
 samples$pch[a]<-15
 samples$pch[b]<-16
 samples$pch[c]<-17
}

ax$tick.label.col[1:7]<-ax$tick.col[1:7]<-"black"
ax$tick.label.col[8:14]<-ax$tick.col[8:14]<-"red"
ax$tick.label.col[15:21]<-ax$tick.col[15:21]<-"blue"
ax$tick.label.col[22:28]<-ax$tick.col[22:28]<-"green"
```

Chapter 7 code

```
#MODIFICATIONS TO CVAbiplot FUNCTION WHERE X IS DATA USED#


# MATRIX ALGEBRA
  Wmat <- vector("list",length(X))
  Lmat <- vector("list",length(X))
  LBL <- vector("list",length(X))
  Cmat <- vector("list",length(X))
  Vmat <- vector("list",length(X))
  Mmat <- vector("list",length(X))
  lambda <- vector("list",length(X))

  for (k in 1:length(X))
    {
        SSP.T <- t(X[[k]]) %*% X[[k]]
        SSP.B <- t(Xbar[[k]]) %*% Nmat %*% Xbar[[k]]
        SSP.W <- SSP.T - SSP.B
        Wmat[[k]] <- SSP.W
    }
#STEP 1 of COMMON CVA PROCESS#
  Lstar <- FG.al2(Wmat, c(rep(1,length(X))))[[1]]

  for (k in 1:length(X))
    {
       lambda[[k]] <- diag(diag(t(Lstar)%*%Wmat[[k]]%*%Lstar))
     Lmat[[k]] <- Lstar%*%diag(diag(lambda[[k]])^-0.5)
         if (weightedCVA == "weighted") Cmat[[k]] <- Nmat
     if (weightedCVA == "unweightedI") Cmat[[k]] <- diag(J)
     if (weightedCVA == "unweightedCent") Cmat[[k]] <- diag(J) - matrix(1/J,
         nrow = J, ncol = J)
    }
    if (is.na(match(weightedCVA, c("weighted", "unweightedI", "
        unweightedCent")))) stop(" Argument ’weightedCVA’ must be one of ’
        weighted','unweightedI','unweightedCent' ")

#STEP 2 OF COMMON CVA PROCESS#
    for (k in 1:length(X))
      LBL[[k]] <- t(Lmat[[k]]) %*% t(Xbar[[k]]) %*% Cmat[[k]] %*% Xbar[[k]] %*% Lmat[[k]]
    Vstar <- FG.al2 (LBL, c(rep(1,length(X))),r=K)[[1]]

  for (k in 1:length(X))
    {
      Vmat[[k]] <- (lambda[[k]])^0.5 %*% Vstar
    Mmat[[k]] <- Lmat[[k]] %*% Vstar
    }


  Zmat <- lapply(X, function(x) matrix (NA, nrow=nrow(x), ncol=dim.biplot, dimnames=list(dimnames(x)[[1]],NULL)))
  Z.means.mat <- vector("list",length(X))
  for (k in 1:length(X))
    {
       Zmat[[k]] <- X[[k]] %*% Mmat[[k]][,e.vects]
     Z.means.mat[[k]] <- Xbar[[k]] %*% Mmat[[k]][,e.vects]
    }
#DEFINE SAMPLE POINTS#
 Z <- Zmat[[1]]
  for (k in 2:length(X))
    Z <- rbind (Z, Zmat[[k]])

#DEFINING VARIABLE AXES#
z.axes.all <- vector("list",length(X)*p)
  i <- 0
  for (k in 1:length(X))
    {
       Mrr <- solve(Mmat[[k]])[e.vects, ,drop=F]
     ax <- NULL
         ax <- do.call("biplot.ax.control", c(num.vars,list(var.names),ax))
         if (ax$type == "prediction") axes.direction <- solve(diag(diag(t(Mrr) %*% Mrr))) %*% t(Mrr)
       else axes.direction <- Mrr[,e.vects, drop=F]

       if (length(ax$which)==0) z.axes <- NULL
         else z.axes <- lapply(1:length(ax$which), calibrate.axis, unscaled.X[[k]], means[[k]], sd=rep(1,length(means[[k]])), axes.direction, ax$which, ax$ticks, ax$orthogx, ax$orthogy, ax$oblique)
     for (j in 1:p)
       {
         i <- i + 1
       z.axes.all[[i]] <- z.axes[[j]]
       }
    }
#Z AND z.axes ARE USED IN THE CVAbiplot FUNCTION#
```

Chapter 8 code

```
#ONE EXAMPLE FOR CENTERED DATA#

matlab <- Matlab()
print(matlab)
isOpen <- open(matlab)

```

```

##USING MATLAB TO RUN THE TUCKER DECOMPOSITION#
################################################
str(parr)
setVariable(matlab,parr=parr)
evaluate(matlab,"[Factors,G,SSE]=tucker(parr,[2 2 2])")
core<-getVariable(matlab,"G")
evaluate(matlab,"[A B C]=fac2let(Factors)")
components<-getVariable(matlab,c("A","B","C"))


#PCAbipl TAKES TWO NEW ARGUMENTS components,core#


#REPRESENTING X1=AF'
###############################
threeway<-array(0,dim=c(n,p,k))

for (i in 1:4)
 threeway[,,i]=as.matrix(Xdat[[i]])
meansk<-vector("list",4)

for (i in 1:k)
  meansk[[i]]<-apply(threeway[,,i],2,mean)
overall.mean <- rep(NA,p)
overall.sd <- rep(NA,p)
for (i in 1:p)
  {
   overall.mean[i] <- mean(threeway[,i,])
   overall.sd[i] <- sd(as.vector(threeway[,i,]))
   threeway[,i,]<-(threeway[,i,]-mean(threeway[,i,]))/sd(as.vector(threeway[,i,]))
  }
XDat<-vector("list",4)
for (i in 1:4)
 XDat[[i]]=as.matrix(threeway[,,i])

#Unfolding the core in the subject mode
#######################################
P<-Q<-R<-2
G1<-matrix(0,P,Q*R)
for (k in 1:R)
{
 a<-1+R*(k-1)
 b<-R*k
 G1[,a:b]<-core$G[,,k]
}

#DEFINING F, Lambda AND SCALED MATRICES
#######################################
A<-components$A
B<-components$B
C<-components$C
F<-(C%x%B)%*%t(G1)
lambda<-diag(diag(G1%*%t(G1)))
Fnew<-F%*%svd(lambda)$u%*%diag(svd(lambda)$d^-1)%*%t(svd(lambda)$v)
Anew<-A%*%svd(lambda)$u%*%diag(svd(lambda)$d)%*%t(svd(lambda)$v)

#MODIFICATION TO PCAbipl FOR WIDE#
Vr <- Fnew
Z <- Anew

```

```
#MODIFICATION TO PCAbipl FOR TALL#
Vr<-Anew
Z<-t(Fnew)

#MULTIPLE MARKERS PROGRAMMED AS FOR CPC#

#TRIPLOT CONSTRUCTION
#########################################
#LROAT ALGORITHM FOR r=2 (Chen and Saad, 2009)
########################
function (threeway,r)
{
I<-dim(threeway)[1]
J<-dim(threeway)[2]
K<-dim(threeway)[3]
print(c(I,J,K))
print(dim(threeway[,,1]))

#SUBJECT MODE#

Unf_subj<-matrix(0,I,J)
for (i in 1:K)
 Unf_subj<-cbind(Unf_subj,threeway[,,i])

Unf_subj<-Unf_subj[,-c(1:J)]

#print(cor(t(Unf_subj)))
print(dist(Unf_subj))

U1<-svd(Unf_subj)$u[,1:r]
#VARIABLE MODE#

Unf_var<-matrix(0,J,K)
for (i in 1:I)
 Unf_var<-cbind(Unf_var,threeway[i,,])

Unf_var<-Unf_var[,-c(1:K)]

print(cor(t(Unf_var)))
U2<-svd(Unf_var)$u[,1:r]


#TIME MODE#

Unf_time<-matrix(0,K,I)
for (i in 1:J)
 Unf_time<-cbind(Unf_time,t(threeway[,i,]))

Unf_time<-Unf_time[,-c(1:I)]

print(cor(t(Unf_time)))

U3<-svd(Unf_time)$u[,1:r]

tr<-function(X)
 {
  return(sum(diag(X)))
 }
 oldcrit<-0
 first<-second<-matrix(NA,I,K)
 first2<-second2<-matrix(NA,K,J)
 done<-FALSE
 while (!done)
# for (l in 1:20)
 {
    #SUBJECT MODE V and Sigma#
  for (i in 1:I)
  {
     first[i,]<-t(U2[,1])%*%threeway[i,,]
   second[i,]<-t(U2[,2])%*%threeway[i,,]
  }
  V1<-matrix(c(t(t(U3[,1])%*%t(first)),t(t(U3[,2])%*%t(second))),ncol=r)
  Sigma<-diag(c(as.double(t(U1[,1])%*%V1[,1]),as.double(t(U1[,2])%*%V1[,2])))
    polart<-svd(V1%*%Sigma)
    U1<-polart$u%*%t(polart$v)
  #VARIABLE MODE V and Sigma#
  for (i in 1:K)
    {
     first2[i,]<-t(U1[,1])%*%threeway[,,i]
   second2[i,]<-t(U1[,2])%*%threeway[,,i]
  }
  V2<-matrix(c(t(t(U3[,1])%*%(first2)),t(t(U3[,2])%*%(second2))),ncol=r)
  Sigma<-diag(c(as.double(t(U2[,1])%*%V2[,1]),as.double(t(U2[,2])%*%V2[,2])))
    polart<-svd(V2%*%Sigma)
    U2<-polart$u%*%t(polart$v)
  #OCCASION MODE V and Sigma#
  V3<-matrix(c(t(t(U2[,1])%*%t(first2)),t(t(U2[,2])%*%t(second2))),ncol=r)
  Sigma<-diag(c(as.double(t(U3[,1])%*%V3[,1]),as.double(t(U3[,2])%*%V3[,2])))
    polart<-svd(V3%*%Sigma)
    U3<-polart$u%*%t(polart$v)
  newcrit<-tr(t(Sigma)%*%(Sigma))
  print(newcrit)
  if (abs(newcrit-oldcrit)<1e-3)
     done<-TRUE
  oldcrit<-newcrit
}
print(U2)
for (i in 1:r)
{
  if (Sigma[i,i] < 0)
  {
    U1[,i] = -1*U1[,i]
    Sigma[i,i] = -Sigma[i,i]
  }
}
 return(list(U1=U1,U2=U2,U3=U3,Sigma=Sigma))
}

#DRAWING THE TRIPLOT
####################
function (X,G,modes)
{
output<-LROAT(X,2)
U1<-output$U1
#plot(U1[,1],U1[,2],pch="",asp=1,xaxt="n",yaxt="n",xlab="",ylab="")
#text(U1[,1],U1[,2],labels=1:29,pos=3)
#stop()
U2<-output$U2
U3<-output$U3
D<-output$Sigma
Aout<-cbind(D[1,1]^(1/3)*U1[,1],D[2,2]^(1/3)*U1[,2])
```

```r
Bout<-cbind(D[1,1]^(1/3)*U2[,1],D[2,2]^(1/3)*U2[,2])
Cout<-cbind(D[1,1]^(1/3)*U3[,1],D[2,2]^(1/3)*U3[,2])
U2<-Bout
U3<-Cout
#plot(Aout[,1],Aout[,2],pch="",asp=1,xaxt="n",yaxt="n",xlab="",ylab="")
#text(Aout[,1],Aout[,2],labels=1:29,pos=3)
#stop()

#######################
#CREATES BASIC TRIPLOT#
#######################
new<-rbind(Aout,Bout,Cout)
#new<-rbind(Aout,U2,U3)
plot(new[,1],new[,2],pch="",asp=1,xaxt="n",yaxt="n",xlab="",ylab="")
abline(h=0,v=0,lty=2)
#points(U2[,1],U2[,2],asp=1,pch=15,col="orange")
#points(U3[,1],U3[,2],col="blue",pch=16)
points(Bout[,1],Bout[,2],asp=1,pch=15,col="orange")
points(Cout[,1],Cout[,2],col="blue",pch=16)
for (i in 1:ncol(G))
 {
 points(Aout[G[,i]==1,1],Aout[G[,i]==1,2],pch=17,col=i)
 }
#text(Aout[,1],Aout[,2],labels=1:nrow(Aout), pos=2)
#stop()
text(Cout[,1],Cout[,2],labels=1:nrow(Cout),pos=2)
text(Bout[,1],Bout[,2], labels=1:nrow(Bout), pos=2)
#legend ("topright",legend=c("variables","occasions","samples"), pch=15:17, col=c("orange","blue","black"))
######################################
#COMBINATION OF MODES
######################################
k<-dim(X)[3]
p<-dim(X)[2]
n<-dim(X)[1]

if (modes[1]==1 && modes[2]==2)
{
 A<-Aout
 B<-Bout
 C<-Cout
 m<-n*p
 g<-nrow(B)
 h<-nrow(A)
 G<-B
 H<-A
} else if (modes[1]==1 && modes[2]==3)
  {
    A<-Aout
  B<-Cout
  C<-Bout
  m<-n*k
  g<-nrow(A)
  h<-nrow(B)
  G<-A
  H<-B
  }
  else
    {
    A<-Bout
    B<-Cout
    C<-Aout
    m<-p*k
    g<-nrow(B)
    h<-nrow(A)
    G<-B
    H<-A
  }
l<-0
colours1<-vector("integer",m)
colours1<-c(rep("black",7),rep("grey",7),rep("brown",7),rep("orange",7))
vtime<-matrix(0,m,2)
for (k in 1:g)
 {
   for(j in 1:h)
   {
     l<-l+1
     b<-H[j,]*(G[k,])
     vtime[l,]=b
     abline(a=0,b=b[[2]]/b[[1]],col=colours1[l])
   #colours1[l]<-k

    }
   }
 #PROJECTS POINTS ONTO AXES#
#  for (i in 1:n)
 #    {
  ##   proj<-((a%*%t(t(b)))/t(b)%*%t(t(b)))*b
   #   points(proj[1],proj[2],col=1,pch=1)
   #  }
   # }
 #}
 l<-0
labels<-labs<-vector("integer",m)
for (i in 1:g)
 for (j in 1:h)
 {
   l=l+1
   labels[l]<-paste(j,i,sep=".")
 }

for (k in 1:m)
labs[k]<-paste("V",labels[k],sep="")
print(labs)
print(colours1)
usr<-par("usr")
FINDING CO-ORIDNATES FOR AXIS LABELS
for (i in 1:m)
{
b<-vtime[i,]
if (b[1]>0 && b[2]>0)
  if (b[2]/b[1] < usr[4]/usr[2])
  {
    m<-b[2]/b[1]
    ytext<-m*usr[2]
    mtext (text=labs[i], side=4, adj=0,at=ytext,cex=0.85,col=colours1[i])
  } else
    {
     m<-b[2]/b[1]
     ytext<-usr[4]/m
     mtext (text=labs[i], side=3, adj=0,at=ytext,cex=0.85,col=colours1[i])
    }

if (b[1]>0 && b[2]<0)
```

```
  if (b[2]/b[1] > usr[3]/usr[2])
  {
    m<-b[2]/b[1]
    ytext<-m*usr[2]
    mtext (text=labs[i], side=4, adj=0,at=ytext,cex=0.85,col=colours1[i])
  } else
    {
     m<-b[2]/b[1]
     ytext<-usr[3]/m
     mtext (text=labs[i], side=1, adj=0,at=ytext,cex=0.85,col=colours1[i])
    }

if (b[1]<0 && b[2]>0)
if (b[2]/b[1] > usr[4]/usr[1])
  {
    m<-b[2]/b[1]
    ytext<-m*usr[1]
    mtext (text=labs[i], side=2, adj=0,at=ytext,cex=0.85,col=colours1[i])
     m<-b[2]/b[1]
     ytext<-usr[4]/m
  } else
    {
     mtext (text=labs[i], side=3, adj=0,at=ytext,cex=0.85,col=colours1[i])
    }
if (b[1]<0 && b[2]<0)
if (b[2]/b[1] < usr[3]/usr[1])
  {
    m<-b[2]/b[1]
    ytext<-m*usr[1]
    mtext (text=labs[i], side=2, adj=0,at=ytext,cex=0.85,col=colours1[i])
  } else
    {
     m<-b[2]/b[1]
     ytext<-usr[3]/m
     mtext (text=labs[i], side=1, adj=0,at=ytext,cex=0.85,col=colours1[i])
    }
}
}
```

Chapter 9 code