The Construction of a Partial Least Squares Biplot

Opeoluwa Funmilayo Oyedele

Dissertation Presented for the Degree of

Doctor of Philosophy in Statistical Sciences

in the Department of Statistical Sciences at the

University of Cape Town

Advisor: Associate Professor Sugnet Lubbe

December 2014

I know the meaning of plagiarism and declare that all of the work in this dissertation, save for that which is properly acknowledged, is my own.

Name of candidate: Opeoluwa Funmilayo Oyedele

Signature:

Date:

I hereby:

(a) grant the University free license to reproduce this dissertation in whole or in part, for the purpose of research;

(b) declare that:

(i) the above dissertation is my own unaided work, both in conception and execution, and that apart from the normal guidance from my advisor, I have received no assistance;

(ii) neither the substance nor any part of the thesis has been submitted in the past, or is being, or is to be submitted for a degree at this University or at any other University.

I am now presenting this dissertation for examination for the Degree of PhD.

Name of candidate: Opeoluwa Funmilayo Oyedele

Signature:

Date:

ABSTRACT OF THE DISSERTATION

The Construction of a Partial Least Squares Biplot

by

Opeoluwa Funmilayo Oyedele

In multivariate analysis, data matrices are often very large, which sometimes makes it difficult to describe their structure and to make a visual inspection of the relationship between their respective rows (samples) and columns (variables). For this reason, biplots, the joint graphical display of the rows and columns of a data matrix, can be useful tools for analysis. Since they were first introduced, biplots have been employed in a number of multivariate methods, such as Correspondence Analysis (CA), Principal Component Analysis (PCA), Canonical Variate Analysis (CVA) and Discriminant Analysis (DA), as a form of graphical display of data. Another possible employment is in Partial Least Squares (PLS). First introduced as a regression method, PLS is more flexible than multivariate regression, but better suited than Principal Component Regression (PCR) for the prediction of a set of response variables from a large set of predictor variables. Employing the biplot in PLS gave rise to the *PLS biplot*, a new addition to the biplot family. In the current study, this biplot was successfully applied to the sensory data to investigate the relationships between the sensory panel characteristics and the chemical quality measurements of sixteen olive oils. It was also applied to a large set of mineral sorting production data to investigate the relationships between the output variables and the process factors used to produce a final product. Furthermore, the PLS biplot was applied to a Binomial-distributed data concerning the diabetes testing of Indian women and to a Poisson-distributed data showing the diversity of arboreal marsupials (possum) in the Montane ash forest. After these applications, it is proposed that the PLS biplot is a useful graphical tool for displaying results from the (univariate) Partial Least Squares-Generalized Linear Model (PLS-GLM) analysis of a data set. With Partial Least Squares Regression (PLSR) being a valuable method for modelling high-dimensional data, especially in chemometrics, the PLS biplot was successfully applied to a cereal evaluation containing one hundred and forty five infrared spectra and six chemical properties, and a gene expression data with two thousand genes.

Moreover, as the PLS biplot provides a single graphical representation of the samples, together with the predictor variables, response variables and their inter-relationships (in terms of the matrix of regression coefficients), two methods of representing the matrix of coefficients in this

biplot were proposed. First was the calibrated biplot axes style, where the coefficients were obtained directly from the orthogonal projections onto the axes that were representing the response variables. Here, *a different set of markers, differing from the one used for reading the response samples, is used to read off the coefficient values*. Thus, in this method, the response axes are carrying two different sets of markers, *one (in black) for reading the response samples* and *one (in purple) for reading the coefficients*. Secondly, the utilization of the area biplots was proposed. Here, the coefficients are represented by triangles in the area biplot. Akin to the advantages of biplots in general, the PLS biplot demonstrates, in graphic form, the association between samples and variables. It also provides a single graphical representation for displaying results from the PLSR, Sparse Partial Least Squares (SPLS), PLS-GLM and/or Sparse Partial Least Squares-Generalized Linear Model (SPLS-GLM) of a data set.

To James, Victoria, Yemisi, John, Ifeolu, and myself.

ACKNOWLEDGEMENTS

Firstly, and most importantly, I would like to thank God Almighty for giving me the strength and courage to face all the challenges I encountered, and for blessing me with His spiritual guidance as well as wisdom throughout this research.

I wish to express my sincere appreciation to my advisor, Associate Professor Sugnet Lubbe, for her valuable contributions, patience, guidance and support throughout this research project. Without her constant suggestions, encouragement and guidance, it would not have been possible for me to complete it. I convey my deepest thanks to the former Head of Statistical Sciences Department, Associate Professor Christien Thiart, for making my registration in the department possible and the current Head of Department, Associate Professor Francesca Little for continued support.

My heartfelt gratitude also goes to my parents, Professor James Akindele Oyedele and Professor Victoria Iyabode Oyedele, for their support, inspirational talks and advice throughout my studies.

I further wish to acknowledge the Postgraduate Centre and Funding Office, Faculty of Science, the Department of Statistical Sciences, Associate Professor Sugnet Lubbe, Professor Niël Le Roux from Stellenbosch University and my parents, for their financial assistance towards my studies.

Lastly, but not least importantly, I would like to express my thanks to all the people who directly or indirectly contributed to the finalization of this dissertation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

CHAPTER 1

PRELIMINARIES

1.1 Introduction

Graphical representations are often used as a quick way to summarize and display data sets effectively. They offer an opportunity to identify patterns, groupings, outliers, and relationships, amongst others, in the data set. Although there are various graphical representations, the most commonly used representations are the two- and three-dimensional scatter plots. In these representations, the coordinate axes represent the columns of the data, while the points represent the rows. In the usual two-dimensional scatter plot, two orthogonal Cartesian axes are used to represent the columns of the data. However, data sets are often (very) large, and therefore it can be time consuming to make a visual inspection of the data using a two- or three-dimensional scatter plot. In addition, all the inter-variable relationships are not displayed in either scatter plots. For this reason, biplots can be useful tools for analysis. First introduced by Gabriel (1971), biplots are often referred to as the multivariate version of scatter plots, in that they allow for the simultaneous display of each column in the data by a non-orthogonal axis on the plot.

At the core of multivariate statistics is the investigation of relationships between different sets of variables. More precisely, the inter-variable relationships and the casual relationships. The latter is a regression problem, where one set of variables is referred to as the response variables and the other set of variables as the predictor variables. In this situation, the effect of the predictors on the response variables is revealed through the regression coefficients. Different solutions have been developed to solve the regression problem. For example, the Multivariate Multiple Linear Regression (MMLR), Ridge Regression (RR), Principal Component Regression (PCR) and Partial Least Squares Regression (PLSR). Furthermore, the biplot can be employed to visualize the effect of the predictors on the response variables graphically, and to display results of the regression. Since their introduction, many forms of biplots have been developed, such as the Correspondence Analysis (CA) biplot, the Principal Component Analysis (PCA) biplot and the Canonical Variate Analysis (CVA) biplot. These forms of biplots are all of the type revealing the inter-variable relationships.

In this dissertation, a new addition to the biplot family is developed and termed the *Partial Least Squares (PLS) biplot*. This biplot allows for the simultaneous representation of sample points and variables. It further provides a single graphical representation for displaying results from the

PLSR analysis of a data set. The graphical visualization is extended to Partial Least Squares (PLS). These extensions are based on (i) rephrasing the biplot theory in the PLS context, (ii) applying PLS biplots to different data sets, and (iii) developing software for executing these applications. As the implementation was developed, different types of response variables led to another form of PLS, namely, the Partial Least Squares-Generalized Linear Model (PLS-GLM). In addition, the implementation was developed for a sparse version of PLS, referred to as Sparse Partial Least Squares (SPLS), and with the different types of response variables, the Sparse Partial Least Squares-Generalized Linear Model (SPLS-GLM) was developed. Though PLSR is a valuable method for modelling high-dimensional biological data, such as genomics, the PLS biplot is proposed as a graphical tool to help reveal any possible variables and inter-variable relationships, clustering and multivariate outliers of such data sets, seeing that no such plot currently exists.

1.2 Objectives of the dissertation

Given that PLSR is another popular multivariate method, the biplot is employed in PLS to form the PLS biplot. The main objective of this dissertation is to construct the PLS biplot. However, before such construction will be described, an overview will be given of biplots, PLS and its algorithms, as well as PLSR. After constructing the PLS biplot, it will be briefly compared to the popular PCA biplot and the covariance biplot. In addition, the use of the PLS biplot as a graphical tool for Generalized Linear Models (GLMs), as well as for a sparse version of PLS and PLS-GLM will be explored.

1.3 Software

A collection of functions has been developed in the R language (R Core Team, 2014). These functions can be found in the newly created R package called **PLSbiplot1** on the dropbox link:

https://www.dropbox.com/sh/wr66u07t1vjm9da/AACg_E4h8MvgOHuCXk69yDIya.

This package is also available on the Comprehensive R Archive Network (CRAN) repository, http://cran.r-project.org/. Furthermore, a brief introductory illustration on the use of these functions is provided in Section 8.5 of this dissertation. A detailed online documentation for all routines in this package can also be found on the dropbox link.

1.4 Organization of the dissertation

Chapter 2 of this dissertation explains the essential concepts behind biplots in general. As the simplest form of a biplot, the PCA biplot is used to explain the relevant biplot concepts. Chapter 3 states, explains and derives the necessary theory behind PLS as well as PLSR. It also discusses and compares three of the algorithms implemented in the R package **pls**, by Mevik & Wehrens (2007). These algorithms are the Nonlinear Iterative PArtial Least Squares (NIPALS) algorithm, the kernel algorithm and the Statistical Inspired Modification to Partial Least Squares (SIMPLS) algorithm. In addition, it includes a short comparison of PLSR with other popular multivariate regression techniques, such as Multivariate Multiple Linear Regression (MMLR) and Principal Component Regression (PCR). Since PLS algorithms are based on the covariance matrix between the predictor variables and the response variables, Chapter 4 investigates the visual representation of the covariance matrix in a biplot. This is followed in Chapter 5 by all the procedures and methods used in obtaining the PLS biplot. It also includes a discussion of the interpolation and prediction of points and axes of a PLS biplot. Alongside these, the PLS biplot is compared with the PCA and covariance biplots.

The PLS biplot constructed in Chapter 5 assumes a linear relationship between the predictors and the response variables. Sometimes the relationship between these two sets of variables may follow a non-linear function. Thus, Chapter 6 investigates the employment of the PLS biplot to explore the non-linear relationship between the predictors and response variables. In addition, the PLS biplot is explored as a graphical tool for displaying a sparse version of PLS in Chapter 7. This version of PLS, called Sparse Partial Least Squares (SPLS), is popularly used in biological data such as microarray expression data and genomics. The SPLS is further implemented into the Generalized Linear Model (GLM) framework, to form the Sparse Partial Least Squares-Generalized Linear Model (SPLS-GLM). Furthermore, in Chapters 2, 3, 4 and 5, the major theoretical concepts are illustrated using the olive oil data from Mevik & Wehrens (2007). This data shows the sensory and chemical quality evaluations of sixteen olive oil samples. Moreover, the PLS biplots developed in Chapters 5, 6 and 7 are applied to three chosen data sets in Chapter 8. A brief overview of how and why the chosen data sets were collected will be provided. In addition, introductory information on the use of the developed PLS biplot software (**PLSbiplot1**) is provided. Finally, Chapter 9 concludes the dissertation and suggests further research directions.

1.5 Notation

Throughout this dissertation, scalars are defined in italic lower-case characters (e.g., x, y) and vectors in bold non-italic lower-case characters (e.g., $\mathbf{x}$, $\mathbf{y}$), while matrices are represented in bold non-italic upper-case characters (e.g., $\mathbf{X}$, $\mathbf{Y}$). To emphasize their dimensions, matrices will be written clearly as, e.g., $\mathbf{X}\,(\mathrm{N} \times \mathrm{P})$, meaning a matrix $\mathbf{X}$ with N rows and P columns. The identity matrices are represented as $\mathbf{I}_{\mathrm{N}}$, with their appropriate dimensions $(\mathrm{N} \times \mathrm{N})$ indicated as a subscript. Matrix elements are represented by the corresponding italic lower-case characters with row and column index subscripts, e.g., x_{ij} is the $(\mathrm{i},\mathrm{j})^{\mathrm{th}}$ element of a matrix $\mathbf{X}$.

Superscripts T and -1 represent transpose and inverse operations respectively, e.g., $\mathbf{y}^{\mathrm{T}}, \mathbf{Y}^{\mathrm{T}}, \mathbf{X}^{-1}$. Estimated values are emphasized with a mark $\wedge$ on them, e.g., $\widehat{\mathbf{X}}$, $\hat{\mathbf{y}}$. Furthermore, the following notation is applicable throughout this dissertation.

A	Number of components in a PLS model
N	Number of samples
P	Number of predictor variables (X-variables)
M	Number of response variables (Y-variables)
a	Index of components; $\mathrm{a} = 1, 2, \cdots, \mathrm{A}$
$\mathbf{X}$	$(\mathrm{N} \times \mathrm{P})$ Matrix of predictors
$\mathbf{Y}$	$(\mathrm{N} \times \mathrm{M})$ Matrix of responses
$\mathbf{X}_0$	$(\mathrm{N} \times \mathrm{P})$ Matrix of centred predictors
$\mathbf{Y}_0$	$(\mathrm{N} \times \mathrm{M})$ Matrix of centred responses
$\mathbf{W}$	$(\mathrm{P} \times \mathrm{A})$ PLS X-weights matrix
$\mathbf{R}$	$(\mathrm{P} \times \mathrm{A})$ PLS transformed X-weights matrix
$\mathbf{C}$	$(\mathrm{M} \times \mathrm{A})$ PLS Y-weights matrix
$\mathbf{T}$	$(\mathrm{N} \times \mathrm{A})$ PLS X-scores or latent variables matrix
$\mathbf{U}$	$(\mathrm{N} \times \mathrm{A})$ PLS Y-scores matrix
$\mathbf{P}$	$(\mathrm{P} \times \mathrm{A})$ PLS X-loadings matrix
$\mathbf{Q}$	$(\mathrm{M} \times \mathrm{A})$ PLS Y-loadings matrix
$\mathbf{B}_{\mathrm{PLSR}}$	$(\mathrm{P} \times \mathrm{M})$ PLSR coefficients matrix
$\mathbf{B}_{\mathrm{SPLS}}$	$(\mathrm{P} \times \mathrm{M})$ SPLS coefficients matrix
$\mathbf{b}_{\mathrm{PLS}}$	$(\mathrm{P} \times 1)$ PLS coefficients vector
$\mathbf{b}_{\mathrm{SPLS}}$	$(\mathrm{P} \times 1)$ SPLS coefficients vector

$\mathbf{b}_{\mathrm{PLS-GLM}}$ $(\mathrm{P} \times 1)$ PLS-GLM coefficients vector

$\mathbf{b}_{\mathrm{SPLS-GLM}}$ $(\mathrm{P} \times 1)$ SPLS-GLM coefficients vector

$\mathbf{w}_{\mathrm{a}}$ a^{th} column of $\mathbf{W}$

$\mathbf{r}_{\mathrm{a}}$ a^{th} column of $\mathbf{R}$

$\mathbf{c}_{\mathrm{a}}$ a^{th} column of $\mathbf{C}$

$\mathbf{t}_{\mathrm{a}}$ a^{th} column of $\mathbf{T}$

$\mathbf{u}_{\mathrm{a}}$ a^{th} column of $\mathbf{U}$

$\mathbf{p}_{\mathrm{a}}$ a^{th} column of $\mathbf{P}$

$\mathbf{q}_{\mathrm{a}}$ a^{th} column of $\mathbf{Q}$

$\|\mathbf{w}_{\mathrm{a}}\|$ Norm of $\mathbf{w}_{\mathrm{a}}$; $\|\mathbf{w}_{\mathrm{a}}\| = \sqrt{\mathbf{w}_{\mathrm{a}}^{\mathrm{T}}\mathbf{w}_{\mathrm{a}}}$

$\mathbf{X}[\,,1]$ First column of $\mathbf{X}$.

$\mathbf{X}[\,,1{:}\,3]$ First three columns of $\mathbf{X}$

$\mathrm{var}(\mathbf{w}_{\mathrm{a}})$ Variance of $\mathbf{w}_{\mathrm{a}}$

$\mathrm{cov}(\mathbf{X})$ $(\mathrm{P} \times \mathrm{P})$ Covariance matrix of $\mathbf{X}$

$\mathrm{cov}(\mathbf{X}, \mathbf{Y})$ $(\mathrm{P} \times \mathrm{M})$ Covariance matrix of $\mathbf{X}$ and $\mathbf{Y}$.

Additional notation will be specified under the relevant chapters.

1.6 List of abbreviations

BAM	Bi-Additive Model
CA	Correspondence Analysis
CRAN	Comprehensive R Archive Network
CVA	Canonical Variate Analysis
DA	Discriminant Analysis
EV-VD	EigenValue-Vector Decomposition
GLM	Generalized Linear Model
GOPA	Generalized Orthogonal Procrustes Analysis
IWLS	Iterative Weighted Least Squares
LASSO	Least Absolute Shrinkage and Selection Operator
LRA	Log-Ratio Analysis
MCA	Multiple Correspondence Analysis
MDS	Multi-Dimensional Scaling
MET	Multi-Environment Trial
MMLR	Multivariate Multiple Linear Regression

MSE	Mean Squared Error
NIPALS	Nonlinear Iterative PArtial Least Squares
NLA	Non-Linear Analysis
OPA	Orthogonal Procrustes Analysis
PA	Procrustes Analysis
PCA	Principal Component Analysis
PCR	Principal Component Regression
PLS	Partial Least Squares
PLS-GLM	Partial Least Squares-Generalized Linear Model
PLSR	Partial Least Squares Regression
PRESS	PRediction Error Sum-of-Squares
RMSEP	Root Mean Squared Error of Prediction
RR	Ridge Regression
SIMPLS	Statistical Inspired Modification to Partial Least Squares
SPLS	Sparse Partial Least Squares
SPLS-GLM	Sparse Partial Least Squares-Generalized Linear Model
SVD	Singular Value Decomposition
VIP	Variable Importance in the Projection
WLS	Weighted Least Squares.

CHAPTER 2

PRINCIPAL COMPONENT ANALYSIS BIPLOTS

2.1 Introduction

A scatter diagram is a useful statistical tool for representing samples as points and variables by means of coordinate axes. A form of scatter diagram that can be used for multivariate data is the biplot. The biplot allows for the simultaneous display of samples as points, with each variable being represented by an axis on the plot. Generally, there are two main kinds of biplots, namely, the symmetric biplot and the asymmetric biplot. However, different forms of biplots have been developed over the years. Among them is the Principal Component Analysis (PCA) biplot, which is the most common and simplest form of asymmetric biplot. In this chapter, the PCA biplot will be used to explain the basic biplot concepts, although the specifics of the PCA biplot will only be discussed in detail in Section 2.10.

2.2 Kinds of Biplots

The asymmetric biplot gives information on the rows and columns of a data matrix, while the symmetric biplot gives information on the rows and columns of a two-way table (Gower *et al*., 2011). The roles of the columns and rows in the symmetric biplot can be interchanged without losing any information. However, this is not the case in the asymmetric biplot. In Partial Least Squares (PLS), the data has the form of N sample points measured on P predictor variables and M response variables. It can be represented as an $N \times (P + M)$ matrix $\mathbf{D} = [\mathbf{X} \quad \mathbf{Y}]$ in which the roles of the columns and rows cannot be interchanged. Thus, the resulting biplot will be an asymmetric biplot, and therefore the focus here will be on asymmetric biplots.

Biplots, to be precise, asymmetric biplots, are often referred to as the multivariate version of scatter plots. In the usual two-dimensional scatter plot, two orthogonal Cartesian axes are used for reading off the values of the sample points, as well as for adding points to the plot. The fact that biplots are referred to as multivariate scatter plots implies that more than two variables are represented by (non-orthogonal) axes (Gardner-Lubbe *et al*., 2009; Gower & Hand, 1996). Just like scatter plots, biplots are helpful for revealing clustering, multivariate outliers, variables and inter-variable relationships of a data set (Kohler & Luniak, 2005).

2.3 A brief review of the literature on biplots

Since the biplot was first introduced by Gabriel (1971), its theory has been significantly extended with Gower & Hand's (1996) monograph, Yan & Kang's (2003) description of various methods to visualize and interpret a biplot, Greenacre's (2010) text on the use of biplots in practice and Gower *et al.*'s (2011) illustration of the construction of various forms of biplots. In the first biplots introduced by Gabriel, the rows and columns of a data matrix were represented by vectors, but to differentiate between these two sets of vectors, Gabriel (1971) suggested that the rows of the data matrix be represented by points. Gower & Hand (1996) went a step further by introducing the idea of representing the columns of the data matrix by axes, rather than vectors, while still representing the rows of the data matrix by points. This was done to support their theory that biplots were the multivariate version of scatter plots. Gower & Hand's (1996) biplot representation is very useful when the data matrix under consideration is a matrix of samples by variables. Besides data matrices, other types of matrices and non-matrices can be biplotted too. Examples of these include the covariance or correlation matrices and a two-way contingency table (Bradu & Gabriel, 1978; Gower *et al*., 2011; Greenacre, 1993; Underhill, 1990). In the covariance or correlation biplot, only variables are represented, by axes, in the biplot display. As only the variables are represented in the display, and not both the samples and variables, Gower *et al*. (2011) termed this display the *monoplot*. The specifics of the monoplot will be discussed in detail in Section 4.2. The biplot of a two-way table is often referred to as a *symmetric biplot*, due to the advantage of interchanging the roles of the columns and rows of the table without any loss of information. An example of such a biplot is the Correspondence Analysis (CA) biplot (Greenacre, 1993).

Since 1971, biplots have been employed in a number of multivariate methods as a form of graphical representation of data, pattern and data inspection, as well as for displaying results found by well-known statistical methods of analysis (Bradu & Gabriel, 1978; Constantine & Gower, 1978; Gabriel, 1981; Gower & Hand, 1996; Gower *et al.*, 2011; Greenacre, 2010). The most well-known methods are PCA, CA, Multiple Correspondence Analysis (MCA), Canonical Variate Analysis (CVA), Multi-Dimensional Scaling (MDS), Discriminant Analysis (DA), regression analysis, and Generalized Linear Models (GLMs). In addition, biplots have been employed in some less well-known methods, such as Bi-Additive models (BAMs), Log-Ratio Analysis (LRA) and Non-Linear Analysis (NLA) (Gower & Harding, 1988; Greenacre, 2012). All these forms of biplots have been applied to diverse fields of specialization, according to different needs and requirements. For instance, biplots were applied in the fields of mineral

engineering (Gardner *et al.*, 2005), in plant and crop sciences (Cooper & De Lacy, 1994; Yan & Kang, 2003; Yan & Rajcan, 2002), agricultural science (Kempton, 1984; Underhill, 1990), microarray and gene expression (Gardner-Lubbe *et al.*, 2009), Multi-Environment Trials (METs) analysis (Bradu & Gabriel, 1978; Kempton, 1984; Yan & Tinker, 2006), medicine (Osmond, 1985), ecology (Ter Braak, 1983), accounting (Le Roux *et al.*, 2003) and economics (Barr & Affleck-Graves, 1987; Barr *et al.*, 1990).

2.4 Fundamental idea of biplots

A biplot is a joint graphical display of the rows and columns of a data matrix $\mathbf{D}$ (of G rows and H columns) by means of markers $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_{\mathrm{G}}$ for its rows and markers $\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_{\mathrm{H}}$ for its columns. Each marker is chosen in such a way that the inner product $\mathbf{a}_{\mathrm{i}}^{\mathrm{T}}\mathbf{b}_{\mathrm{j}}$ represents d_{ij}, the $(\mathrm{i},\mathrm{j})^{\mathrm{th}}$ element of the data matrix $\mathbf{D}$ (Barnett, 1981). Biplots are often constructed in two dimensions. This does not mean that they are limited to two dimensions, but this is the most convenient biplot display. However, with $\mathbf{D}$ being a (very) large matrix, the rank of $\mathbf{D}$ is almost always higher than two. As a result, some approximation is done on $\mathbf{D}$ to obtain a lower rank. Methods such as PCA can be used to perform this approximation. In PCA, the approximation is based on the method of least squares. To be precise, the sum-of-squares of the differences between $\mathbf{D}$ and its approximation $\widehat{\mathbf{D}}$ is minimized. That is,

$$\text{minimize } \operatorname{trace}\left\{\left(\mathbf{D}-\widehat{\mathbf{D}}\right)\left(\mathbf{D}-\widehat{\mathbf{D}}\right)^{\mathrm{T}}\right\} = \sum_{\mathrm{i}=1}^{\mathrm{G}}\sum_{\mathrm{j}=1}^{\mathrm{H}}\left(d_{\mathrm{ij}}-\hat{d}_{\mathrm{ij}}\right)^2. \qquad (2.1)$$

Taking $\widehat{\mathbf{D}}$ as the rank two approximation of $\mathbf{D}$, the biplot of a data matrix $\mathbf{D}$ relies on the decomposition of $\widehat{\mathbf{D}}$ into the product of two matrices,

$$\widehat{\mathbf{D}} = \mathbf{A}\mathbf{B}^{\mathrm{T}}, \qquad (2.2)$$

its row markers matrix $(\mathbf{A})$ and its column markers matrix $(\mathbf{B})$. Matrices $\mathbf{A}$ and $\mathbf{B}$ are defined as

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1\mathrm{r}} \\ a_{21} & a_{22} & \cdots & a_{2\mathrm{r}} \\ \vdots & \vdots & \ddots & \vdots \\ a_{\mathrm{G}1} & a_{\mathrm{G}2} & \cdots & a_{\mathrm{Gr}} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{(1)}^{\mathrm{T}} \\ \mathbf{a}_{(2)}^{\mathrm{T}} \\ \vdots \\ \mathbf{a}_{(\mathrm{G})}^{\mathrm{T}} \end{bmatrix} \text{ the } (\mathrm{G}\times\mathrm{r}) \text{ row markers matrix and}$$

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1\mathrm{r}} \\ b_{21} & b_{22} & \cdots & b_{2\mathrm{r}} \\ \vdots & \vdots & \ddots & \vdots \\ b_{\mathrm{H}1} & b_{\mathrm{H}2} & \cdots & b_{\mathrm{Hr}} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{(1)}^{\mathrm{T}} \\ \mathbf{b}_{(2)}^{\mathrm{T}} \\ \vdots \\ \mathbf{b}_{(\mathrm{H})}^{\mathrm{T}} \end{bmatrix} \text{ the } (\mathrm{H}\times\mathrm{r}) \text{ column markers matrix.}$$

Thus, the approximated rows and columns of a data matrix are represented in biplots. Generally, the number of columns in $\mathbf{A}$ and $\mathbf{B}$ are determined by the rank r approximation of $\mathbf{D}$. In practice, $\mathrm{r} = 2$ is usually preferred for a convenient biplot display.

2.5 Biplot points and axes

In asymmetric biplots, the rows of a data matrix are represented by points, while the columns are represented by vectors or axes. Traditionally, columns are represented by vectors (Gabriel, 1971), but Gower & Hand (1996) introduced axes to make the biplot similar to a scatter plot. This was done by extending the vectors, which represent the columns, through the biplot space to become axes. Thus, the biplot points will be defined by the row markers of the data matrix, whereas the biplot axes will be defined by the column markers. More precisely, for the biplot of a data matrix $\mathbf{D}$, G rows of $\mathbf{A}$ will serve as the biplot points, while H rows of $\mathbf{B}$ will be used in calculating the directions of the biplot axes.

2.6 Axes relation

With the focus being on asymmetric biplots, as mentioned in Section 2.2 above, in the asymmetric biplots presented in this dissertation, two out of three aspects can be represented optimally, but not all three at once. These aspects are: (i) the distances between the rows of $\mathbf{A}$, (ii) the correlations between the rows of $\mathbf{B}$, and, (iii) the relationship between the rows and columns in $\widehat{\mathbf{D}}$.

Consider a data matrix with Singular Value Decomposition (SVD) $\mathbf{D} = \mathbf{M\Lambda L}^{\mathrm{T}}$, for $\mathbf{M}$ (G × H), $\mathbf{\Lambda}$ (H × H) and $\mathbf{L}$ (H × H). Defining the matrices $\mathbf{J}$ (H × H) $= \begin{bmatrix} \mathbf{I}_{\mathrm{r}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ and $\mathbf{J}_{\mathrm{r}}$ (H × r) $= \begin{bmatrix} \mathbf{I}_{\mathrm{r}} \\ \mathbf{0} \end{bmatrix}$, it follows that

$$\mathbf{D} \cong \widehat{\mathbf{D}} = \mathbf{M\Lambda JL}^{\mathrm{T}} = \mathbf{MJ\Lambda L}^{\mathrm{T}} = \mathbf{MJ\Lambda JL}^{\mathrm{T}}.$$

If

$$\mathbf{A} = \mathbf{M\Lambda}^{\alpha}\mathbf{J}_{\mathrm{r}} \text{ and } \mathbf{B} = \mathbf{L\Lambda}^{\beta}\mathbf{J}_{\mathrm{r}}, \text{ where } \alpha + \beta = 1, \qquad (2.3)$$

then

$$\widehat{\mathbf{D}} = \mathbf{AB}^{\mathrm{T}}.$$

However, if $\alpha + \beta \neq 1$, $\widehat{\mathbf{D}} = \mathbf{MJ\Lambda}^{\alpha+\beta}\mathbf{JL}^{\mathrm{T}} \neq \mathbf{AB}^{\mathrm{T}}$.

From (2.3), choosing $\alpha = 1$ yields $\beta = 0$ and

$$\begin{aligned} \widehat{\mathbf{D}} = \mathbf{MJ\Lambda}^{\alpha+\beta}\mathbf{JL}^{\mathrm{T}} = \mathbf{MJ\Lambda}^{\alpha}\mathbf{\Lambda}^{\beta}\mathbf{JL}^{\mathrm{T}} &= \mathbf{MJ\Lambda}^{1}\mathbf{\Lambda}^{0}\mathbf{JL}^{\mathrm{T}} \\ &= \mathbf{MJ\Lambda JL}^{\mathrm{T}} \\ &= \mathbf{AJL}^{\mathrm{T}} \end{aligned}$$

with $\alpha + \beta = 1$. Since $\mathbf{J}_{\mathrm{r}}^{\mathrm{T}}\mathbf{J}_{\mathrm{r}} = [\mathbf{I}_{\mathrm{r}} \quad \mathbf{0}] \begin{bmatrix} \mathbf{I}_{\mathrm{r}} \\ \mathbf{0} \end{bmatrix} = \mathbf{I}_{\mathrm{r}}\mathbf{I}_{\mathrm{r}} = \mathbf{I}_{\mathrm{r}}$ and $\mathbf{L}^{\mathrm{T}}\mathbf{L} = \mathbf{I}_{\mathrm{H}}$,

$$\widehat{\mathbf{D}}_{[\mathrm{r}]}\widehat{\mathbf{D}}_{[\mathrm{r}]}^{\mathrm{T}} = \mathbf{AJL}^{\mathrm{T}}(\mathbf{AJL}^{\mathrm{T}})^{\mathrm{T}} = \mathbf{AJ}_{\mathrm{r}}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}}\mathbf{LJ}_{\mathrm{r}}\mathbf{A}^{\mathrm{T}} = \mathbf{AA}^{\mathrm{T}}$$

where $\widehat{\mathbf{D}}_{[\mathrm{r}]}$ is the r-approximation of $\mathbf{D}$. Thus, for row c and row e of $\mathbf{D}$,

$$\begin{aligned}\left\|\hat{\mathbf{d}}_{(c)}^{T}-\hat{\mathbf{d}}_{(e)}^{T}\right\| &= \left(\hat{\mathbf{d}}_{(c)}^{T}-\hat{\mathbf{d}}_{(e)}^{T}\right)\left(\hat{\mathbf{d}}_{(c)}^{T}-\hat{\mathbf{d}}_{(e)}^{T}\right)^{T}\\ &= \hat{\mathbf{d}}_{(c)}^{T}\hat{\mathbf{d}}_{(c)}-2\hat{\mathbf{d}}_{(c)}^{T}\hat{\mathbf{d}}_{(e)}+\hat{\mathbf{d}}_{(e)}^{T}\hat{\mathbf{d}}_{(e)}\\ &= \mathbf{a}_{(c)}^{T}\mathbf{a}_{(c)}-2\mathbf{a}_{(c)}^{T}\mathbf{a}_{(e)}+\mathbf{a}_{(e)}^{T}\mathbf{a}_{(e)}\\ &= \left\|\mathbf{a}_{(c)}^{T}-\mathbf{a}_{(e)}^{T}\right\|\end{aligned}$$

so that the squared distance between row c and row e of $\mathbf{D}$ is approximated by the squared distance between row markers $\mathbf{a}_{(c)}$ and $\mathbf{a}_{(e)}$.

Furthermore, with $\alpha+\beta=1$, choosing $\beta=1$ yields $\alpha=0$,

$$\begin{aligned}\widehat{\mathbf{D}}=\mathbf{MJ}\boldsymbol{\Lambda}^{\alpha+\beta}\mathbf{JL}^{T}=\mathbf{MJ}\boldsymbol{\Lambda}^{\alpha}\boldsymbol{\Lambda}^{\beta}\mathbf{JL}^{T} &= \mathbf{MJ}\boldsymbol{\Lambda}^{0}\boldsymbol{\Lambda}^{1}\mathbf{JL}^{T}\\ &= \mathbf{MJ}\boldsymbol{\Lambda}\mathbf{JL}^{T}\\ &= \mathbf{MJB}^{T}\end{aligned}$$

and

$$\widehat{\mathbf{D}}_{[r]}^{T}\widehat{\mathbf{D}}_{[r]}=(\mathbf{MJB}^{T})^{T}(\mathbf{MJB}^{T})=\mathbf{BJ}_{r}^{T}\mathbf{M}^{T}\mathbf{MJ}_{r}\mathbf{B}^{T}=\mathbf{BB}^{T},$$

where $\mathbf{M}^{T}\mathbf{M}=\mathbf{I}_{H}$.

For $\alpha=\beta$, with either $\alpha=\beta=0$ and $\alpha=\beta=1$, $\widehat{\mathbf{D}}=\mathbf{MJ}\boldsymbol{\Lambda}^{\alpha+\beta}\mathbf{JL}^{T}\neq\mathbf{AB}^{T}$. Here, the emphasis of the graphical displays will be on the relationship between the rows and columns in $\widehat{\mathbf{D}}$ as an approximation to the relationship between the rows and columns in $\mathbf{D}$. Greenacre (2009) refer to such display as the *symmetric plot*, for $\alpha=\beta=0.5$. Either ways, $\widehat{\mathbf{D}}\neq\mathbf{AB}^{T}$.

With $\alpha+\beta=1$, it is impossible to have both $\alpha=1$ (optimally representing the distances between the rows of $\widehat{\mathbf{D}}$) and $\beta=1$ (optimally representing the correlations between the columns of $\widehat{\mathbf{D}}$). For this reason, the arbitrary choice is made to optimally represent the distances between the rows, (i.e., $\alpha=1$), therefore, having $\beta=0$ and the angles between the vectors representing the columns only indicating the cosine of the correlations, rather than exactly approximating the cosine of the correlations between the columns. For a discussion on the biplot that optimally approximates the correlation between variables, see Chapter 4. The positions of the biplot axes give an indication of the correlations between the variables. Axes forming small angles are said to be strongly correlated – either positively or negatively. Axes are positively correlated when they lie in the same direction, while negatively correlated axes lie in opposite directions. In addition, axes that are close to forming right angles are said to be uncorrelated.

Since $(N-1)\hat{\mathbf{S}}=\widehat{\mathbf{D}}_{[r]}^{T}\widehat{\mathbf{D}}_{[r]}$, for column c and column e of $\mathbf{D}$,

$$(N-1)\hat{s}_{ce}=\hat{\mathbf{d}}_{c}^{T}\hat{\mathbf{d}}_{e}=\mathbf{b}_{(c)}^{T}\mathbf{b}_{(e)}=\left\|\mathbf{b}_{(c)}\right\|\times\left\|\mathbf{b}_{(e)}\right\|\times\cos\left(\mathbf{b}_{(c)},\mathbf{b}_{(e)}\right)$$

with correlation $\hat{r}_{ce}=\dfrac{\hat{\mathbf{d}}_{c}^{T}\hat{\mathbf{d}}_{e}}{\sqrt{\hat{\mathbf{d}}_{c}^{T}\hat{\mathbf{d}}_{c}}\sqrt{\hat{\mathbf{d}}_{e}^{T}\hat{\mathbf{d}}_{e}}}$ defined as

$$
\begin{aligned}
\hat{r}_{ce} &= \frac{\|\mathbf{b}_{(c)}\| \times \|\mathbf{b}_{(e)}\| \times \cos(\mathbf{b}_{(c)},\mathbf{b}_{(e)})}{\sqrt{\|\mathbf{b}_{(c)}\| \times \|\mathbf{b}_{(c)}\| \times \cos(\mathbf{b}_{(c)},\mathbf{b}_{(c)})}\sqrt{\|\mathbf{b}_{(e)}\| \times \|\mathbf{b}_{(e)}\| \times \cos(\mathbf{b}_{(e)},\mathbf{b}_{(e)})}} \\
&= \frac{\|\mathbf{b}_{(c)}\| \times \|\mathbf{b}_{(e)}\| \times \cos(\mathbf{b}_{(c)},\mathbf{b}_{(e)})}{\sqrt{\|\mathbf{b}_{(c)}\| \times \|\mathbf{b}_{(c)}\| \times 1}\sqrt{\|\mathbf{b}_{(e)}\| \times \|\mathbf{b}_{(e)}\| \times 1}} \\
&= \frac{\|\mathbf{b}_{(c)}\| \times \|\mathbf{b}_{(e)}\| \times \cos(\mathbf{b}_{(c)},\mathbf{b}_{(e)})}{\|\mathbf{b}_{(c)}\| \times \|\mathbf{b}_{(e)}\|} \\
&= \cos(\mathbf{b}_{(c)}, \mathbf{b}_{(e)}).
\end{aligned}
$$

This shows that the angle between the column markers approximates the correlation between the corresponding variables. For a single column of $\mathbf{D}$, say, c,

$$(N-1)\hat{s}_c^2 = \|\mathbf{b}_{(c)}\| \times \|\mathbf{b}_{(c)}\| \times \cos(\mathbf{b}_{(c)}, \mathbf{b}_{(c)}) = \|\mathbf{b}_{(c)}\|^2 \qquad (2.4)$$

where $\cos(\mathbf{b}_{(c)}, \mathbf{b}_{(c)}) = 1$ and $\|\mathbf{b}_{(c)}\| = (\mathbf{b}_{(c)}^T\mathbf{b}_{(c)})^{\frac{1}{2}}$.

2.7 Calibration of biplot axes

From (2.4), the length of a vector representing a single column (variable) of **D** is directly related to the standard deviation of the column (variable). In this dissertation, since the columns of **D** are represented by axes, the length of a vector is illustrated in the biplot displays by a (thicker) arrow, emanating from the origin, on their respective axis. However, the calibration of the axes is very important in biplots. This is because different calibrations are used for adding points to the biplot and for reading off values from the biplot axes. In addition, calibrated axes can be used for interpolation and prediction purposes. Given the values of the variables for a sample, interpolation describes the process of finding the position of the sample in the display (Gower & Hand, 1996). Conversely, prediction is the process of inferring or deducing the values of the original variables for any points in the display. Although calibrated interpolation axes can be used in general, these axes will not be used in this dissertation. Instead, *using the resulting biplots just like scatter plots and with the main objective of both PCA and PLS being the approximation of large data matrices, calibrated prediction axes will be used throughout the PCA and PLS biplots*. It is intuitive to read off values from axes when presented with a biplot containing sample points and biplot axes. Using interpolation axes will be misleading as these cannot be used to read off values. For this reason, only prediction biplot axes will be shown in the figures. Interpolation can be, and is usually, performed algebraically so that there is no need for representing biplots with interpolation axes, especially when users tend to automatically refer sample points to the biplot axes.

As each marker is chosen in such a way that the inner product $\mathbf{a}_{(i)}^T\mathbf{b}_{(j)}$ represents the $(i,j)^{th}$ element of $\mathbf{D}$, $\hat{d}_{ij} = \mathbf{a}_{(i)}^T\mathbf{b}_{(j)}$ prediction is performed graphically by reading off directly from the

orthogonal projections of the biplot points onto the calibrated biplot axes, just as in scatter plots, where points are projected onto the axes to read their values.

In general, calibration is done by placing a set of tick marks on each of the biplot axes and then labelling them with any set of markers (not necessarily equally spaced) as desired, e.g., $(0, 1, 2, 3, 5, \cdots)$.

2.7.1 *Calibration factor*

The row and column markers matrices in (2.2) can be rewritten as

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_{(1)}^{\mathrm{T}} \\ \mathbf{a}_{(2)}^{\mathrm{T}} \\ \vdots \\ \mathbf{a}_{(\mathrm{G})}^{\mathrm{T}} \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} \mathbf{b}_{(1)}^{\mathrm{T}} \\ \mathbf{b}_{(2)}^{\mathrm{T}} \\ \vdots \\ \mathbf{b}_{(\mathrm{H})}^{\mathrm{T}} \end{bmatrix},$$

with $\mathbf{a}_{(\mathrm{i})}^{\mathrm{T}} = (a_{\mathrm{i}1} \quad a_{\mathrm{i}2})$ being the (1×2) i^{th} row marker and $\mathbf{b}_{(\mathrm{j})}^{\mathrm{T}} = (b_{\mathrm{j}1} \quad b_{\mathrm{j}2})$ the (1×2) j^{th} column marker, where $\mathrm{i} = 1, 2, \cdots, \mathrm{G}, \mathrm{j} = 1, 2, \cdots, \mathrm{H}$ and $\mathrm{r} = 2$. Consider the projection of $\mathbf{a}_{(\mathrm{i})}$ on $\mathbf{b}_{(\mathrm{j})}$ as shown below in Figure 2.1.

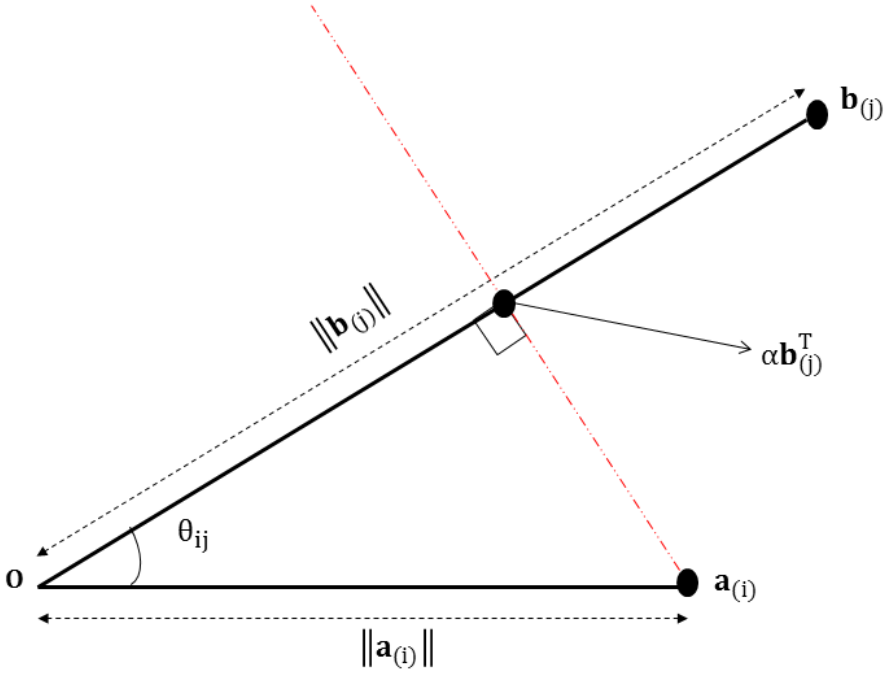

Figure 2.1 A schematic of the projection of $\mathbf{a}_{(\mathrm{i})}$ on $\mathbf{b}_{(\mathrm{j})}$.

The inner product of $\mathbf{a}_{(\mathrm{i})}$ on $\mathbf{b}_{(\mathrm{j})}$ gives the $(\mathrm{i}, \mathrm{j})^{\mathrm{th}}$ element of $\widehat{\mathbf{D}}$ as $\hat{d}_{\mathrm{ij}} = \mathbf{a}_{(\mathrm{i})}^{\mathrm{T}} \mathbf{b}_{(\mathrm{j})}$. From the origin $\mathbf{O}$, the length of $\mathbf{a}_{(\mathrm{i})}$ is given by $\|\mathbf{a}_{(\mathrm{i})}\| = (\mathbf{a}_{(\mathrm{i})}^{\mathrm{T}} \mathbf{a}_{(\mathrm{i})})^{\frac{1}{2}}$, while the length of $\mathbf{b}_{(\mathrm{j})}$ is obtained by $\|\mathbf{b}_{(\mathrm{j})}\| = (\mathbf{b}_{(\mathrm{j})}^{\mathrm{T}} \mathbf{b}_{(\mathrm{j})})^{\frac{1}{2}}$. The angle between $\mathbf{a}_{(\mathrm{i})}$ and $\mathbf{b}_{(\mathrm{j})}$ is defined by θ_{ij}. The point $\alpha \mathbf{b}_{(\mathrm{j})}^{\mathrm{T}}$ is the projection of $\mathbf{a}_{(\mathrm{i})}$ on $\mathbf{b}_{(\mathrm{j})}$ with a calibration factor α. The inner product $\mathbf{a}_{(\mathrm{i})}^{\mathrm{T}} \mathbf{b}_{(\mathrm{j})} = \|\mathbf{a}_{(\mathrm{i})}\| \times$

$\|\mathbf{b}_{(j)}\| \times \cos(\theta_{ij}) = \|\mathbf{b}_{(j)}\| \times \|\mathbf{a}_{(i)}\| \times \cos(\theta_{ij})$ can be written as the product of $\|\mathbf{b}_{(j)}\|$ and the length of the projection of $\mathbf{a}_{(i)}$ on $\mathbf{b}_{(j)}$ $(\|\mathbf{a}_{(i)}\| \times \cos(\theta_{ij}))$. All vectors $\mathbf{a}_{(k)}$ with end points on the red dotted line, in Figure 2.1, through $\mathbf{a}_{(i)}$ and $\alpha\mathbf{b}_{(j)}^T$ will have the same projection and therefore the same length of projection $\|\mathbf{a}_{(k)}\| \times \cos(\theta_{kj})$. Thus, $\mathbf{a}_{(i)}^T\mathbf{b}_{(j)}$ has a constant value (say, μ^*) for all vectors $\mathbf{a}_{(k)}$ with end points on the red dotted line through $\mathbf{a}_{(i)}$ and $\alpha\mathbf{b}_{(j)}^T$. That is,

$$\mathbf{a}_{(i)}^T\mathbf{b}_{(j)} = \mu^* = \mathbf{a}_{(k)}^T\mathbf{b}_{(j)} \qquad (2.5)$$

for all $\mathbf{a}_{(k)}$ on the red dotted line. Since $\alpha\mathbf{b}_{(j)}^T$ is a point on the red dotted line, $\alpha\mathbf{b}_{(j)}^T\mathbf{b}_{(j)} = \mu^*$. Solving for α yields

$$\alpha = \frac{\mu^*}{\mathbf{b}_{(j)}^T\mathbf{b}_{(j)}}. \qquad (2.6)$$

Equation (2.6) can be termed the *calibration factor*. Replacing α in $\alpha\mathbf{b}_{(j)}^T$ by (2.6) gives the coordinates of the point on the $\mathbf{b}_{(j)}$ axis that is calibrated with a value μ^* as

$$\frac{\mu^*}{\mathbf{b}_{(j)}^T\mathbf{b}_{(j)}}\mathbf{b}_{(j)}^T.$$

2.8 Example

The following example is an illustration of a biplot, using the olive oil data from Mevik & Wehrens (2007). This data shows the sensory and chemical quality evaluations of sixteen olive oil samples. There were five chemical quality measurements (Acidity, Peroxide, K232, K270 and DK) taken, and six sensory panel characteristics (Yellow, Green, Brown, Glossy, Transparent and Syrup) were used in this evaluation. The sixteen olive oils are assigned as samples, while the chemical quality measurements and sensory panel characteristics are the variables. As a result, the olive oil data can be viewed as a (16×11) data matrix. This data can be obtained from the **pls** package in R, downloaded freely from CRAN, http://cran.r-project.org/.

The row markers **A** and the column markers **B** are shown in Table 2.4. To illustrate the calibration of the biplot axes (see Section 2.7), consider the Yellow variable. The column marker vector for this variable, $\mathbf{b}_{(j=6)}$, is given by the sixth row in **B**. By the definition of a biplot, $\mathbf{a}_{(i)}^T\mathbf{b}_{(6)} = \hat{d}_{i6}$, for $i = 1, 2, \cdots, 16$ and $\mathbf{b}_{(6)}^T = (0.624 \quad -0.016)$. This defines the inner product of the samples and the Yellow variable. Substituting $\mu^* = \hat{d}_{i6}$ in equation (2.6) gives the calibration factor for the Yellow axis as $\alpha_Y = \frac{d_{i6}}{0.390}$. For values ranging between -75.6

and 58.0, $\alpha_{\mathrm{Y}}\mathbf{b}_{(6)}^{\mathrm{T}}$ gives the set of tick markers for the Yellow axis. More precisely, with the centred values under the Yellow variable given as

$$d_6 = \begin{pmatrix} -29.48,\ -27.48,\ -18.17,\ -20.68,\ 0.92,\ -10.17,\ 2.92,\ -24.48, \\ 14.83,\ -5.88,\ 20.03,\ 22.62,\ 17.22,\ 16.72,\ 20.53,\ 20.53 \end{pmatrix},$$

the set of tick markers for the Yellow axis is given by

$$\alpha_{\mathrm{Y}}\mathbf{b}_{(6)}^{\mathrm{T}} = \begin{pmatrix} -47.19,\ 1.15,\ -29.10,\ 0.87,\ 1.48,\ 0.43,\ 4.68,\ 1.03,\ 23.74, \\ 0.25,\ 32.06,\ -0.95,\ 27.58,\ -0.70,\ 32.86,\ -0.86 \end{pmatrix}.$$

These values are not yet corrected for the mean of Yellow. Correction for the mean of Yellow is done by adding the mean of Yellow to $\alpha_{\mathrm{Y}}\mathbf{b}_{(6)}^{\mathrm{T}}$. That is, $\alpha_{\mathrm{Y}}\mathbf{b}_{(6)}^{\mathrm{T}} + \bar{d}_6$, where $\bar{d}_6 = 50.9$ is the mean of Yellow, as shown in Table 2.3. Thus, corrected for the mean of Yellow, the set of tick markers for the Yellow axis is given as

$$\begin{pmatrix} 3.7,\ 52.0,\ 21.8,\ 51.7,\ 52.4,\ 51.3,\ 55.6,\ 51.9, \\ 74.6,\ 51.1,\ 82.9,\ 49.9,\ 78.5,\ 50.2,\ 83.7,\ 50.0 \end{pmatrix}.$$

Rather than using these disproportionate values as the scale markers on the Yellow axis, nicer scale markers can be used, such as $(0,\ 10,\ 20,\ 30,\ 40,\ 50,\ 60,\ 70,\ 80,\ 90)$, as seen in Figure 2.2. The uncentred and original values of the olive oil data are shown in Table 2.2. The asymmetric biplot of this data is shown in Figure 2.2. In this figure, the samples of the data are represented by the black points, while the variables are represented by the axes. *This biplot also shows a representation of the variance of each variable, represented by the thicker arrow (vector) on each axis*. From this display, the standard deviation of Acidity is smaller compared to the others, while DK has a large deviation. This is evident from the length of the vector on these axes. Furthermore, several relationships can be deduced from this biplot, such as a relation between Syrup, K232 and Peroxide. Observing their respective correlation values of 0.69, 0.76 and 0.87 from Figure 2.4, a fairly good relation exists between them. Another relationships deduction is the relation between K270, Transp and Glossy, and between Green, Yellow and DK. *Although the correlations between variables are not optimally approximated in this biplot (Figure 2.2), relationship deductions are done based on the angle between axes*. For example, the angle between Glossy and Transp is acute and larger than the angle between Syrup and K232, corresponding to the former having a higher correlation value of 0.97, while the latter has a correlation value of 0.69, as shown in Figure 2.4.

To get the approximated values of the olive oil data from the biplot in Figure 2.2, each sample point in the biplot is orthogonally projected onto the axes and their respective values are read off. For example, sample point G1 projected onto the Peroxide and Brown axes yields the values 13.5 and 8.5 respectively, as shown in Figure 2.3. The approximated values of the olive oil data are shown in Table 2.1.

Table 2.1 The approximated olive oil values.

	Acidity	Peroxide	K232	K270	DK	Yellow	Green	Brown	Glossy	Transp	Syrup
G1	0.7	13.2	1.8	0.14	0.001	17.0	76.8	8.4	76.0	69.8	49.3
G2	0.4	13.9	1.8	0.13	-0.001	39.9	47.0	12.2	78.5	74.7	48.9
G3	0.4	11.1	1.6	0.11	-0.002	50.9	35.5	8.4	83.2	80.7	46.5
G4	0.5	13.0	1.8	0.13	0.000	28.7	62.1	9.3	77.9	73.0	48.7
G5	0.5	12.1	1.7	0.12	-0.001	39.0	49.7	8.9	80.4	76.5	47.7
I1	0.3	19.3	2.2	0.15	0.000	34.4	49.3	21.2	71.8	67.2	52.8
I2	0.3	14.0	1.8	0.12	-0.002	50.9	32.9	13.6	80.0	77.4	48.5
I3	0.4	16.2	2.0	0.14	0.000	33.4	53.2	15.6	75.1	70.5	50.7
I4	0.3	13.5	1.7	0.12	-0.002	55.6	27.3	13.4	81.2	79.0	48.0
I5	0.1	20.5	2.2	0.14	-0.001	48.7	29.9	24.9	72.6	69.5	53.0
S1	0.2	10.8	1.5	0.10	-0.003	71.5	9.3	10.4	86.4	86.1	45.5
S2	0.2	9.2	1.4	0.09	-0.004	72.1	9.9	7.7	88.2	88.0	44.4
S3	0.2	12.3	1.6	0.11	-0.003	61.0	21.3	11.9	83.3	81.8	46.9
S4	0.2	12.0	1.6	0.10	-0.003	68.1	12.5	12.2	84.6	83.8	46.5
S5	0.2	10.2	1.5	0.09	-0.003	71.9	9.3	9.4	87.1	86.9	45.0
S6	0.2	10.5	1.5	0.10	-0.003	71.0	10.2	9.9	86.6	86.2	45.3

Table 2.2 The olive oil values.

	Acidity	Peroxide	K232	K270	DK	Yellow	Green	Brown	Glossy	Transp	Syrup
G1	0.7	12.7	1.9	0.14	0.003	21.4	73.4	10.1	79.7	75.2	50.3
G2	0.2	12.3	1.7	0.12	-0.004	23.4	66.3	9.8	77.8	68.7	51.7
G3	0.3	10.3	1.6	0.12	-0.005	32.7	53.5	8.7	82.3	83.2	45.4
G4	0.7	13.7	1.7	0.17	-0.002	30.2	58.3	12.2	81.1	77.1	47.8
G5	0.5	11.2	1.5	0.12	-0.001	51.8	32.5	8.0	72.4	65.3	46.5
I1	0.3	18.7	2.1	0.14	0.001	40.7	42.9	20.1	67.7	63.5	52.2
I2	0.2	15.3	1.9	0.12	0.000	53.8	30.4	11.5	77.8	77.3	45.2
I3	0.3	18.5	1.9	0.13	0.001	26.4	66.5	14.2	78.7	74.6	51.8
I4	0.4	15.6	1.8	0.10	0.000	65.7	12.1	10.3	81.6	79.6	48.3
I5	0.2	19.4	2.2	0.16	-0.003	45.0	31.9	28.4	75.7	72.9	52.8
S1	0.2	10.5	1.5	0.12	-0.004	70.9	12.2	10.8	87.7	88.1	44.5
S2	0.2	8.1	1.5	0.11	-0.002	73.5	9.7	8.3	89.9	89.7	42.3
S3	0.3	12.5	1.6	0.09	-0.002	68.1	12.0	10.8	78.4	75.1	46.4
S4	0.2	11.0	1.6	0.09	-0.003	67.6	13.9	11.9	84.6	83.8	48.5
S5	0.2	10.8	1.3	0.09	-0.003	71.4	10.6	10.8	88.1	88.5	46.7
S6	0.3	11.4	1.4	0.09	-0.004	71.4	10.0	11.4	89.5	88.5	47.2

Table 2.3 The column means of the olive oil data.

Acidity	Peroxide	K232	K270	DK	Yellow	Green	Brown	Glossy	Transp	Syrup
0.3	13.3	1.7	0.1	0.0	50.9	33.5	12.3	80.8	78.2	48.0

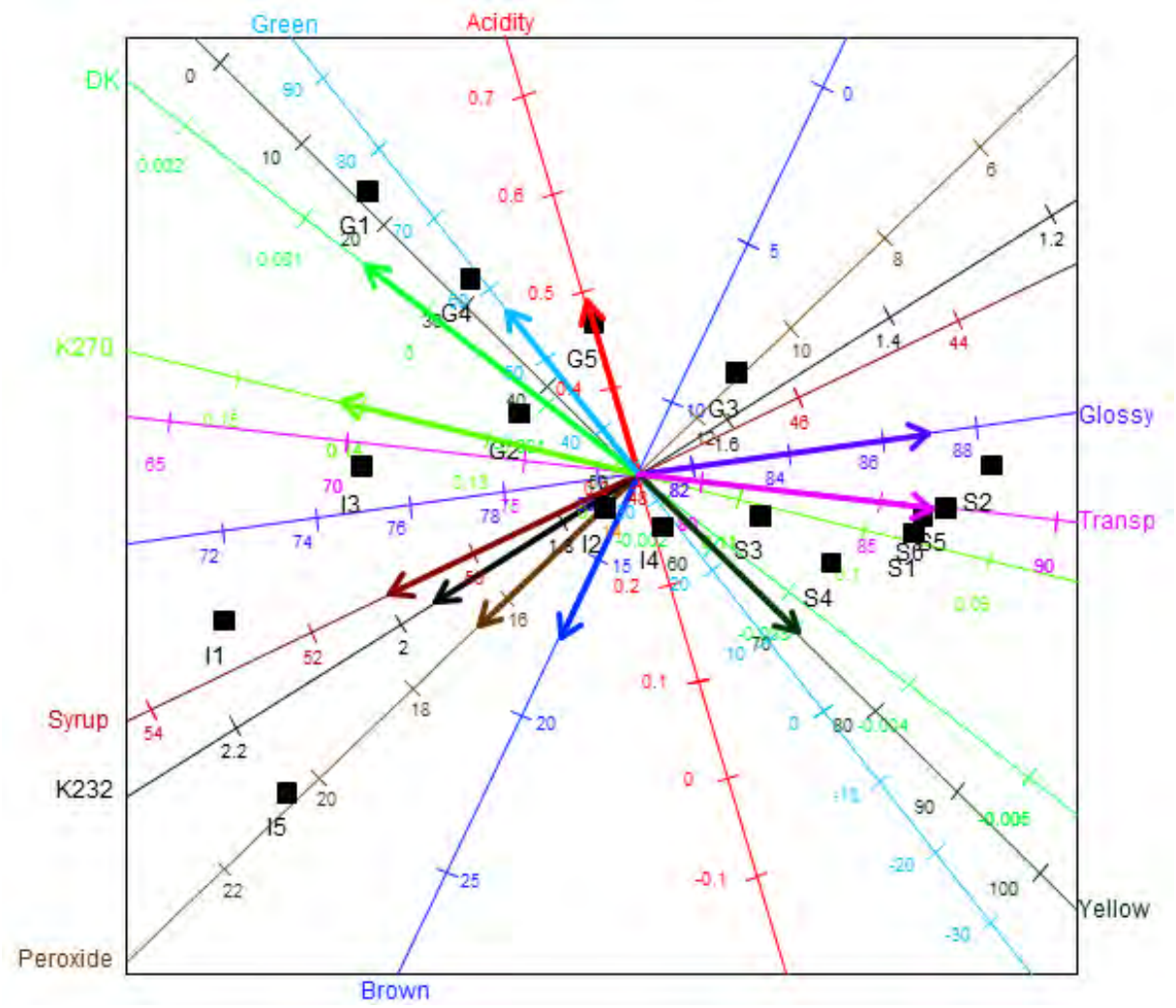

Figure 2. 2 The biplot of the olive oil data.

Table 2. 4 The row and column markers **A** and **B**.

	Component 1	Component 2
G1	-49.037	8.502
G2	-43.814	1.303
G3	-24.975	11.854
G4	-31.675	5.998
G5	-1.475	-11.163
I1	-18.004	-18.977
I2	3.774	-2.816
I3	-41.389	2.696
I4	25.581	-4.205
I5	-4.864	-14.392
S1	31.296	6.335
S2	35.605	9.690
S3	26.283	-8.117
S4	26.605	1.147
S5	32.776	5.985
S6	33.311	6.157

	Component 1	Component 2
Acidity	-0.003	0.002
Peroxide	-0.044	-0.239
K232	-0.004	-0.015
K270	-0.001	-0.001
DK	0.000	0.000
Yellow	0.624	-0.016
Green	-0.751	0.253
Brown	-0.024	-0.346
Glossy	0.111	0.540
Transp	0.168	0.665
Syrup	-0.060	-0.159

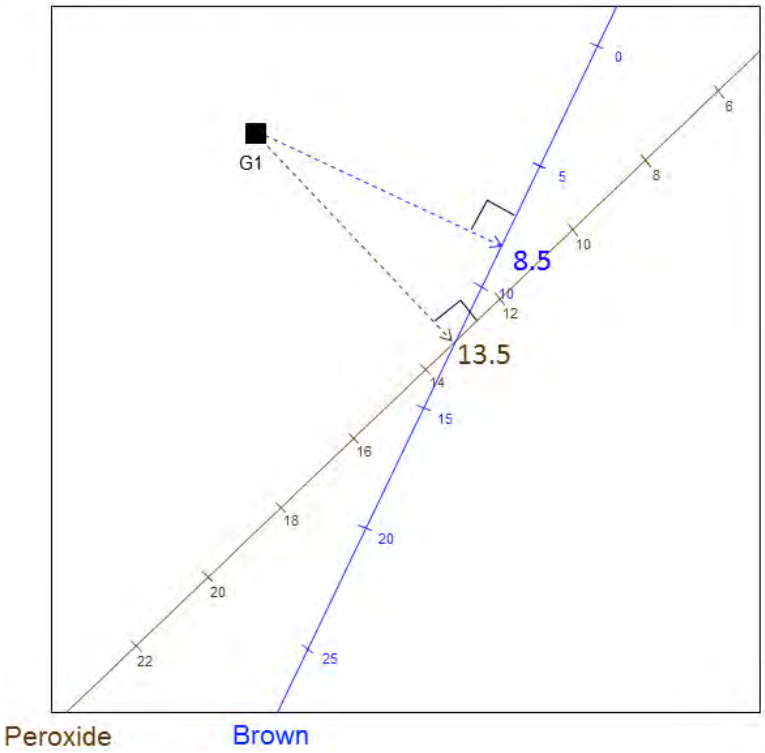

Figure 2. 3 Examples of orthogonal projections in the biplot of the olive oil data.

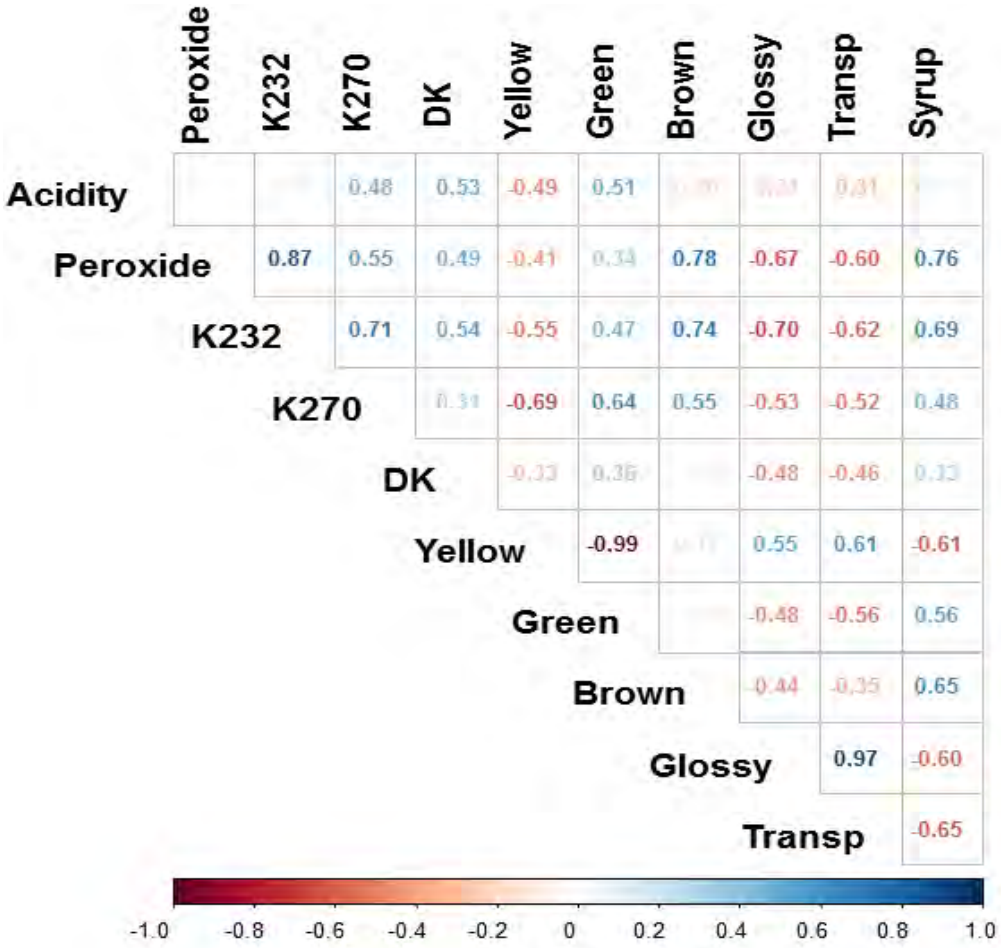

Figure 2. 4 The correlation values of the olive oil data. Nearly empty cells have a value very close to zero.

2.9 Principal Component Analysis

PCA is a multivariate technique that uses an orthogonal transformation to convert a data matrix of possibly correlated variables into a new data matrix of uncorrelated (orthogonal) variables called *principal components* (Abdi & Williams, 2010). These principal components are obtained as linear combinations of the original variables. The first principal component is chosen to have the largest possible variance. Each succeeding component is then computed under the restriction of being orthogonal to the previous components, and to have the largest possible variance (Holland, 2008). Consider a data matrix $\mathbf{D}$ with N samples and P variables, i.e., $\mathbf{D}$ $(\mathrm{N} \times \mathrm{P})$. With (2.1) in mind, the Huygens' Principle is implemented on $\mathbf{D}$. According to Huygens' Principle, the sum-of-squares about the mean of a data matrix is smaller than the sum-of-squares about any other point. Therefore, without loss of generality, the mean of a data matrix is taken to be the origin and all principal components will contain the origin. Let $\mathbf{D}$ be column centred by subtracting the mean of each variable respectively. Denote the centred $\mathbf{D}$ by $\mathbf{D}_0$. Now, the first principal component is given by the linear combination of the D-variables $\mathbf{d}_{0_1}, \mathbf{d}_{0_2}, \cdots, \mathbf{d}_{0\mathrm{P}}$. That is,

$$\mathbf{c}_1 = \varphi_{11}\mathbf{d}_{0_1} + \varphi_{12}\mathbf{d}_{0_2} + \cdots + \varphi_{1\mathrm{P}}\mathbf{d}_{0\mathrm{P}}.$$

Component $\mathbf{c}_1$ is calculated such that it accounts for the largest possible variation in $\mathbf{D}_0$. The second principal component is then calculated in the same way,

$$\mathbf{c}_2 = \varphi_{21}\mathbf{d}_{0_1} + \varphi_{22}\mathbf{d}_{0_2} + \cdots + \varphi_{2\mathrm{P}}\mathbf{d}_{0\mathrm{P}},$$

with the restriction that it is uncorrelated with $\mathbf{c}_1$ and accounts for the next highest variation in $\mathbf{D}_0$. This continues until a total of $\mathrm{r}(\leq \mathrm{P})$ principal components have been calculated. Collectively, the principal components can be expressed as r linear combinations of $\mathbf{D}_0$. That is,

$$\mathbf{C} = \mathbf{D}_0\mathbf{\Phi} \tag{2.7}$$

where

$$\mathbf{\Phi}^{\mathrm{T}} = \begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1\mathrm{P}} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2\mathrm{P}} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{\mathrm{r}1} & \varphi_{\mathrm{r}2} & \cdots & \varphi_{\mathrm{r}\mathrm{P}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varphi}_{(1)}^{\mathrm{T}} \\ \boldsymbol{\varphi}_{(2)}^{\mathrm{T}} \\ \vdots \\ \boldsymbol{\varphi}_{(\mathrm{r})}^{\mathrm{T}} \end{bmatrix}$$

is the $(\mathrm{r} \times \mathrm{P})$ coefficient matrix of the linear combinations and $\mathbf{C} = (\mathbf{c}_1^{\mathrm{T}} \quad \mathbf{c}_2^{\mathrm{T}} \quad \cdots \quad \mathbf{c}_{\mathrm{r}}^{\mathrm{T}})^{\mathrm{T}}$ is the $(\mathrm{N} \times \mathrm{r})$ matrix of the principal component scores.

In general, there are two ways to view PCA, namely, as a dimension reduction technique and as an approximation technique. As a dimension reduction technique, PCA reduces the dimensionality of a large data matrix, described by several inter-correlated variables, while

retaining as much as possible of the variation present in the original data matrix (Jolliffe, 1986; Mardia *et al*., 1979). That is,

$$\mathbf{S}_{\mathrm{C}} = \mathrm{var}(\mathbf{D}_0\mathbf{\Phi}^{\mathrm{T}}) = \mathbf{\Phi}\mathbf{S}_{\mathrm{D}}\mathbf{\Phi}^{\mathrm{T}}$$

is maximized, where $\mathbf{S}_{\mathrm{D}}$ is the covariance matrix of $\mathbf{D}_0$. From (2.7), the original P-dimensional data matrix has been reduced to r dimensions. However, if $\mathrm{r} = \mathrm{P}$, then there is no dimension reduction. Conversely, PCA as an approximation technique approximates a large data matrix using only a few (r) principal components. It works by decomposing the large data matrix as the product of two smaller matrices, which can be called the *scores* and *loadings* matrices respectively. The loadings matrix contains information about the columns of the data matrix, while the scores matrix contains information about its rows. To accomplish this, analysis is based on the SVD. Consider the centred data matrix $\mathbf{D}_0$. By the SVD, $\mathbf{D}_0 = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}$, for $\mathbf{U}$ $(\mathrm{N} \times \mathrm{P})$, $\mathbf{\Lambda}$ $(\mathrm{P} \times \mathrm{P})$ and $\mathbf{V}$ $(\mathrm{P} \times \mathrm{P})$. Since $\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}_{\mathrm{P}}$, it is possible to write the matrix product $\mathbf{U}\mathbf{\Lambda}$ $(\mathrm{N} \times \mathrm{P})$ as

$$\mathbf{U}\mathbf{\Lambda} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{D}_0\mathbf{V}. \qquad (2.8)$$

This expression has an arrangement similar to (2.7), with $\mathbf{C} = \mathbf{U}\mathbf{\Lambda}$ and $\mathbf{\Phi} = \mathbf{V}$, for $\mathrm{r} = \mathrm{P}$. Thus, the matrix product $\mathbf{U}\mathbf{\Lambda}$ gives the values of the principal components. Substituting (2.8) into $\mathbf{D}_0 = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}$ approximates $\mathbf{D}_0$ as $\widehat{\mathbf{D}}_0 = \mathbf{D}_0\mathbf{V}\mathbf{V}^{\mathrm{T}}$. This makes PCA viewable as an approximation technique. From the SVD of $\mathbf{D}_0$, it follows that

$$\mathbf{D}_0^{\mathrm{T}}\mathbf{D}_0 = (\mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}})^{\mathrm{T}}(\mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}) = \mathbf{V}\mathbf{\Lambda}\mathbf{U}^{\mathrm{T}}\mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}} = \mathbf{V}\mathbf{\Lambda}^2\mathbf{V}^{\mathrm{T}},$$

where $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}_{\mathrm{P}}$, for $\mathbf{D}_0^{\mathrm{T}}\mathbf{D}_0 = (\mathrm{N}-1)\,\mathbf{S}_{\mathrm{D}}$ the covariance matrix of $\mathbf{D}$ and $\mathbf{D}_0$, and $\mathbf{\Lambda}^2$ the (ordered) eigenvalues of $\mathbf{D}_0^{\mathrm{T}}\mathbf{D}_0$. In this PCA technique, information is preserved as much as possible by ensuring that the sum-of-squares of the differences between the data matrix $\mathbf{D}_0$ and its approximation $\widehat{\mathbf{D}}_0$ is minimized. That is,

$$\left\|\mathbf{D}_0 - \widehat{\mathbf{D}}_0\right\|^2 = \mathrm{trace}\left\{\left(\mathbf{D}_0 - \widehat{\mathbf{D}}_0\right)\left(\mathbf{D}_0 - \widehat{\mathbf{D}}_0\right)^{\mathrm{T}}\right\} \text{ is minimal.}$$

According to Eckart & Young (1936), the best r-dimensional approximation of $\mathbf{D}_0$ that minimizes $\left\|\mathbf{D}_0 - \widehat{\mathbf{D}}_{0[\mathrm{r}]}\right\|^2$ is obtained by

$$\widehat{\mathbf{D}}_{0[\mathrm{r}]} = \mathbf{D}_0\mathbf{V}_{[\mathrm{r}]}\mathbf{V}_{[\mathrm{r}]}^{\mathrm{T}} \qquad (2.9)$$

where $\widehat{\mathbf{D}}_{0[\mathrm{r}]}$ is the r-dimensional approximation of $\mathbf{D}_0$ and $\mathbf{V}_{[\mathrm{r}]}$ contains the first r columns of $\mathbf{V}$. The matrix product $\mathbf{D}_0\mathbf{V}_{[\mathrm{r}]}$ in (2.9) is often referred to as the $(\mathrm{N} \times \mathrm{r})$ matrix of (principal component) scores or latent variables and denoted by $\mathbf{Z}$. Also, the matrix $\mathbf{V}_{[\mathrm{r}]}$ is termed the $(\mathrm{P} \times \mathrm{r})$ loadings matrix. When comparing $\mathbf{Z} = \mathbf{D}_0\mathbf{V}_{[\mathrm{r}]}$ to (2.7), similar arrangement can be observed. With $\mathbf{V}_{[\mathrm{r}]} = \mathbf{\Phi}$, $\mathbf{Z} = \mathbf{D}_0\mathbf{V}_{[\mathrm{r}]} = \mathbf{D}_0\mathbf{\Phi} = \mathbf{C}$.

2.10 PCA Biplot

Substituting $\mathbf{Z} = \mathbf{D}_0\mathbf{V}_{[r]}$ into (2.9) gives

$$\widehat{\mathbf{D}}_{0[r]} = \mathbf{Z}\mathbf{V}_{[r]}^{T}. \qquad (2.10)$$

This has an arrangement similar to (2.2). Thus, for its biplot, $\widehat{\mathbf{D}}_{0[r]}$ is decomposed as a product of its row and column markers matrices $\mathbf{Z}$ and $\mathbf{V}_{[r]}$ respectively. Given that the approximated rows and columns of a data matrix are represented in biplots generally, the rows and columns of $\widehat{\mathbf{D}}_{0[r]}$ (2.10) will be represented in the biplot. This biplot is termed the *PCA biplot* because $\widehat{\mathbf{D}}_{0[r]}$ was decomposed using PCA. An advantage of the PCA biplot is that it allows for the visual assessment of a high-dimensional data matrix in a two- or three-dimensional plot.

In order to construct a two-dimensional PCA biplot, the best two-dimensional plane, passing through the origin, is first obtained. Here, the plane is the vector space generated by the columns of $\mathbf{V}_{[r]}$, where $r = 2$. This plane has to pass through the origin in order to represent the P-dimensional plot of $\mathbf{D}_0$ optimally in a two-dimensional plane. That is, $\left\|\mathbf{D}_0 - \widehat{\mathbf{D}}_{0[r]}\right\|^2$ is minimal. Once the biplot plane has been obtained, each point in the P-dimensional plot is projected onto this plane (2.10). These projections are done orthogonally so that the sum-of-squares between the original point and its projection is minimized (Gower *et al*., 2011). The biplot plane with the projected points on it is then extracted from the P-dimensional space to a new set of orthogonal Cartesian axes, defined by the columns of $\mathbf{V}_{[r]}$. These axes are referred to as the *scaffolding axes*, and they represent the latent variables ($\mathbf{Z}$) on which the biplot is built.

2.10.1 *Interpolation of samples*

In the PCA biplot, interpolation is achieved by orthogonally projecting each sample point onto the biplot plane (Gower *et al*., 2011). Suppose $\mathbf{d}_0$ is a centred sample. With $\mathbf{V}_{[r]}^{T}\mathbf{V}_{[r]} = \mathbf{I}_r$, $\mathbf{Z}\mathbf{V}_{[r]}^{T}$ in (2.10) can be written as

$$\mathbf{Z}\mathbf{V}_{[r]}^{T} = \mathbf{D}_0\mathbf{V}_{[r]}\mathbf{V}_{[r]}^{T} = \mathbf{D}_0\mathbf{V}_{[r]}\left(\mathbf{V}_{[r]}^{T}\mathbf{V}_{[r]}\right)^{-1}\mathbf{V}_{[r]}^{T}.$$

From (2.9) and (2.10), the representation of sample $\mathbf{d}_0$ projected onto the biplot plane is given by

$$\mathbf{d}_{0\,\mathrm{proj}}^{T} = \mathbf{d}_0^{T}\mathbf{V}_{[r]}\left(\mathbf{V}_{[r]}^{T}\mathbf{V}_{[r]}\right)^{-1}\mathbf{V}_{[r]}^{T} = \mathbf{z}^{T}\mathbf{V}_{[r]}^{T}. \qquad (2.11)$$

As a result, the coordinates of the projections of sample $\mathbf{d}_0$ onto the biplot plane are given by $\mathbf{z}^{T}$. That is, sample $\mathbf{d}_0$ is interpolated into the biplot plane by

$$\mathbf{z}^{T} = \mathbf{d}_0^{T}\mathbf{V}_{[r]}. \qquad (2.12)$$

2.10.2 *Prediction of samples*

Consider a point $\mathbf{z}^*$ $(r \times 1)$ described in terms of the coordinate system of the biplot plane. The point representing $\mathbf{z}^*$ in the biplot plane also has a coordinate representation $\mathbf{d}_0^*$ relative to the axes of the P-dimensional space. This is true because the biplot plane is a subspace of the P-dimensional space. In (2.11), the coordinates of the point in the P-dimensional space is given by $\mathbf{d}_{0_{\text{proj}}}$ and the coordinates of the point in the r-dimensional space are given by $\mathbf{z}^*$. So, any point $\mathbf{z}^*$ $(r \times 1)$ in terms of the basis for the biplot plane is also a point $\mathbf{d}_{0_{\text{proj}}}^*$ $(P \times 1)$ in terms of the basis for the P-dimensional space of $\mathbf{D}_0$ and such a point will project onto itself. To be precise,

$$\mathbf{d}_{0_{\text{proj}}}^{*\text{T}} = \mathbf{d}_{0_{\text{proj}}}^{*\text{T}} \mathbf{V}_{[r]} \mathbf{V}_{[r]}^{\text{T}}.$$

With the interpolation of a point $\mathbf{d}_0^*$ given by $\mathbf{z}^{*\text{T}} = \mathbf{d}_0^{*\text{T}} \mathbf{V}_{[r]}$, $\mathbf{d}_{0_{\text{proj}}}^{*\text{T}} = \mathbf{z}^{*\text{T}} \mathbf{V}_{[r]}^{\text{T}}$. Therefore, sample $\mathbf{d}_0^*$ is predicted by

$$\hat{\mathbf{d}}_0^{*\text{T}} = \mathbf{z}^{*\text{T}} \mathbf{V}_{[r]}^{\text{T}}. \qquad (2.13)$$

2.10.3 *Prediction of axes*

The column markers for the PCA biplot are defined by the rows of the matrix $\mathbf{V}_{[r]}$. From the axis calibration factor discussed in Section 2.7, replacing $\mathbf{b}_j$ in (2.6) by $\mathbf{V}_{[r]}^{\text{T}} \mathbf{e}_k$ gives the calibration factor for the k^{th} axis as

$$\alpha = \frac{\mu^*}{\mathbf{e}_k^{\text{T}} \mathbf{V}_{[r]} \mathbf{V}_{[r]}^{\text{T}} \mathbf{e}_k}$$

where $\mathbf{e}_k$ is the unit vector with zeros except for a one in the k^{th} position. Therefore, the marker μ^* on the k^{th} prediction biplot axis is obtained by the expression

$$\frac{\mu^*}{\mathbf{e}_k^{\text{T}} \mathbf{V}_{[r]} \mathbf{V}_{[r]}^{\text{T}} \mathbf{e}_k} \mathbf{V}_{[r]}^{\text{T}} \mathbf{e}_k.$$

2.11 Measures of fit for PCA biplots

Since the best two-dimensional biplot plane in which to project the points has been found and so have the coordinates of the points when projected onto the plane, the quality of the representation provided by these projections is required to determine the adequacy of the representation of the original data matrix $\mathbf{D}_0$. In other words, how close is $\widehat{\mathbf{D}}_{0_{[r]}}$ to $\mathbf{D}_0$? To evaluate the quality of representation, consider $\mathbf{D}_0$ partitioned into a fitted part $\widehat{\mathbf{D}}_{0_{[r]}}$ and a residual part $\left(\mathbf{D}_0 - \widehat{\mathbf{D}}_{0_{[r]}}\right)$. That is,

$$\mathbf{D}_0 = \widehat{\mathbf{D}}_{0_{[r]}} + \left(\mathbf{D}_0 - \widehat{\mathbf{D}}_{0_{[r]}}\right). \qquad (2.14)$$

Equation (2.14) can be considered as an orthogonal decomposition of $\mathbf{D}_0$, in that

$$\|\mathbf{D}_0\|^2 = \left\|\widehat{\mathbf{D}}_{0[r]}\right\|^2 + \left\|\mathbf{D}_0 - \widehat{\mathbf{D}}_{0[r]}\right\|^2.$$

This is the orthogonality condition of the sum-of-squares decomposition. From (2.14), two types of orthogonality can be obtained. First is

$$\mathbf{D}_0\mathbf{D}_0^{\mathrm{T}} = \widehat{\mathbf{D}}_{0[r]}\widehat{\mathbf{D}}_{0[r]}^{\mathrm{T}} + \left(\mathbf{D}_0 - \widehat{\mathbf{D}}_{0[r]}\right)\left(\mathbf{D}_0 - \widehat{\mathbf{D}}_{0[r]}\right)^{\mathrm{T}}. \qquad (2.15)$$

This is true because, from (2.9),

$$\begin{aligned}\mathbf{D}_0\left(\mathbf{D}_0 - \widehat{\mathbf{D}}_{0[r]}\right)^{\mathrm{T}} &= \mathbf{D}_0\mathbf{D}_0^{\mathrm{T}} - \mathbf{D}_0\mathbf{V}_{[r]}\mathbf{V}_{[r]}^{\mathrm{T}}\mathbf{D}_0^{\mathrm{T}} \\ &= \mathbf{D}_0\mathbf{D}_0^{\mathrm{T}} - \mathbf{D}_0\mathbf{D}_0^{\mathrm{T}} \\ &= \mathbf{0}\end{aligned}$$

and

$$\begin{aligned}\widehat{\mathbf{D}}_{0[r]}\left(\mathbf{D}_0 - \widehat{\mathbf{D}}_{0[r]}\right)^{\mathrm{T}} &= \mathbf{D}_0\mathbf{V}_{[r]}\mathbf{V}_{[r]}^{\mathrm{T}}\mathbf{D}_0^{\mathrm{T}} - \mathbf{D}_0\mathbf{V}_{[r]}\mathbf{V}_{[r]}^{\mathrm{T}}\mathbf{V}_{[r]}\mathbf{V}_{[r]}^{\mathrm{T}}\mathbf{D}_0^{\mathrm{T}} \\ &= \mathbf{D}_0\mathbf{D}_0^{\mathrm{T}} - \mathbf{D}_0\mathbf{V}_{[r]}\mathbf{V}_{[r]}^{\mathrm{T}}\mathbf{D}_0^{\mathrm{T}} \\ &= \mathbf{D}_0\mathbf{D}_0^{\mathrm{T}} - \mathbf{D}_0\mathbf{D}_0^{\mathrm{T}} \\ &= \mathbf{0}\end{aligned}$$

for $\mathbf{V}_{[r]}^{\mathrm{T}}\mathbf{V}_{[r]} = \mathbf{I}_{\mathrm{r}}$ and $\mathbf{V}_{[r]}\mathbf{V}_{[r]}^{\mathrm{T}} = \mathbf{I}_{\mathrm{P}}$. The second type of orthogonality is

$$\mathbf{D}_0^{\mathrm{T}}\mathbf{D}_0 = \widehat{\mathbf{D}}_{0[r]}^{\mathrm{T}}\widehat{\mathbf{D}}_{0[r]} + \left(\mathbf{D}_0 - \widehat{\mathbf{D}}_{0[r]}\right)^{\mathrm{T}}\left(\mathbf{D}_0 - \widehat{\mathbf{D}}_{0[r]}\right) \qquad (2.16)$$

where

$$\begin{aligned}\mathbf{D}_0^{\mathrm{T}}\left(\mathbf{D}_0 - \widehat{\mathbf{D}}_{0[r]}\right) &= \mathbf{D}_0^{\mathrm{T}}\mathbf{D}_0 - \mathbf{D}_0^{\mathrm{T}}\mathbf{D}_0\mathbf{V}_{[r]}\mathbf{V}_{[r]}^{\mathrm{T}} \\ &= \mathbf{D}_0^{\mathrm{T}}\mathbf{D}_0 - \mathbf{D}_0^{\mathrm{T}}\mathbf{D}_0 \\ &= \mathbf{0}\end{aligned}$$

and

$$\begin{aligned}\widehat{\mathbf{D}}_{0[r]}^{\mathrm{T}}\left(\mathbf{D}_0 - \widehat{\mathbf{D}}_{0[r]}\right) &= \mathbf{V}_{[r]}\mathbf{V}_{[r]}^{\mathrm{T}}\mathbf{D}_0^{\mathrm{T}}\mathbf{D}_0 - \mathbf{V}_{[r]}\mathbf{V}_{[r]}^{\mathrm{T}}\mathbf{D}_0^{\mathrm{T}}\mathbf{D}_0\mathbf{V}_{[r]}\mathbf{V}_{[r]}^{\mathrm{T}} \\ &= \mathbf{D}_0^{\mathrm{T}}\mathbf{D}_0 - \mathbf{V}_{[r]}\mathbf{V}_{[r]}^{\mathrm{T}}\mathbf{D}_0^{\mathrm{T}}\mathbf{D}_0 \\ &= \mathbf{D}_0^{\mathrm{T}}\mathbf{D}_0 - \mathbf{D}_0^{\mathrm{T}}\mathbf{D}_0 \\ &= \mathbf{0}\end{aligned}$$

for $\mathbf{V}_{[r]}\mathbf{V}_{[r]}^{\mathrm{T}} = \mathbf{I}_{\mathrm{P}}$. Gardner-Lubbe *et al*. (2008) termed (2.15) and (2.16) the *Type A* and the *Type B orthogonality* respectively.

2.11.1 *Sample predictivity*

The degree to which the rows of $\widehat{\mathbf{D}}_{0[r]}$ agree with the corresponding rows of $\mathbf{D}_0$ measures how far each sample is from its r-dimensional approximation (Gardner-Lubbe *et al*., 2008). With the sum-of-squares of the approximated values for each sample given by the diagonal elements of $\left(\widehat{\mathbf{D}}_{0[r]}\widehat{\mathbf{D}}_{0[r]}^{\mathrm{T}}\right)$, expressing these sum-of-squares as a proportion of their respective total sum-of-squares yield the predictive power of each sample. To be precise,

$$\text{Sample predictivity} \;=\; \text{diag}\left(\widehat{\mathbf{D}}_{0[r]}\widehat{\mathbf{D}}_{0[r]}^{T}\right)\left[\,\text{diag}(\mathbf{D}_0\mathbf{D}_0^{T})\,\right]^{-1}.$$

Due to Type A orthogonality (2.15), the sample predictivity values lie between 0 and 1, with 0 *indicating that the sample is orthogonal to the two-dimensional biplot plane of approximation and* 1 *implying that the sample is in the plane*.

2.11.2 *Axis predictivity*

The evaluation of how well the individual biplot axes reproduce the variables of $\mathbf{D}_0$ can be done by measuring the degree to which the columns of $\widehat{\mathbf{D}}_{0[r]}$ agree with the corresponding columns of $\mathbf{D}_0$. Expressing the sum-of-squares of the approximated values for each variable, given by $\text{diag}\left(\widehat{\mathbf{D}}_{0[r]}^{T}\widehat{\mathbf{D}}_{0[r]}\right)$, as a proportion of their respective total sum-of-squares yields the predictive power of each axis. More precisely,

$$\text{Axis predictivity} \;=\; \text{diag}\left(\widehat{\mathbf{D}}_{0[r]}^{T}\widehat{\mathbf{D}}_{0[r]}\right)\left[\,\text{diag}(\mathbf{D}_0^{T}\mathbf{D}_0)\,\right]^{-1}.$$

Because of Type B orthogonality (2.16), the predictivity values lie between 0 and 1. *An axis predictivity of* 1 *means that all values can be read off the axis exactly. The lower the axis predictivity value, the less accurately the axis approximates the observed values under that variable*.

2.11.3 *Overall quality of approximation*

Overall, the quality of approximation can be measured in terms of the percentage of variation in $\mathbf{D}_0$ explained by $\mathbf{Z} = \mathbf{D}_0\mathbf{V}_{[r]}$. From the SVD of $\mathbf{D}_0$, with $\mathbf{U}^{T}\mathbf{U} = \mathbf{I}_P$,

$$\mathbf{D}_0^{T}\mathbf{D}_0 = (\mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^{T})^{T}(\mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^{T}) = \mathbf{V}\boldsymbol{\Lambda}\mathbf{U}^{T}\mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^{T} = \mathbf{V}\boldsymbol{\Lambda}^{2}\mathbf{V}^{T}.$$

Since $\mathbf{V}^{T}\mathbf{V} = \mathbf{I}_P = \mathbf{V}\mathbf{V}^{T}$,

$$\text{tr}\{\,\mathbf{D}_0^{T}\mathbf{D}_0\} = \text{tr}\{\mathbf{V}\boldsymbol{\Lambda}^{2}\mathbf{V}^{T}\} = \text{tr}\{\boldsymbol{\Lambda}^{2}\mathbf{V}\mathbf{V}^{T}\} = \sum_{j=1}^{P}\lambda_j^{2} = \sum_{j=1}^{P}\sigma_j^{2},$$

where $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_P)$, λ_j is the j^{th} singular value of $\mathbf{D}_0$ and $\sigma_j^2 = \lambda_j^2$ is the j^{th} eigenvalue (and singular value) of $\text{cov}(\mathbf{D}_0)$. Thus,

$$\begin{aligned}\text{Overall quality} \;&=\; \frac{\text{tr}\left\{\widehat{\mathbf{D}}_{0[r]}^{T}\widehat{\mathbf{D}}_{0[r]}\right\}}{\text{tr}\{\mathbf{D}_0^{T}\mathbf{D}_0\}}\\ &=\; \frac{\sum_{j=1}^{r}\lambda_j^{2}}{\sum_{j=1}^{P}\lambda_j^{2}}\\ &=\; \frac{\sum_{j=1}^{r}\sigma_j^{2}}{\sum_{j=1}^{P}\sigma_j^{2}}.\end{aligned}$$

2.12 Example

The following example is an illustration of a PCA biplot using the olive oil data discussed in Section 2.8. The resulting biplot is shown in Figure 2.2 above. This biplot has an overall

quality of 0.990. Furthermore, the predictivity of each biplot axis is estimated and is shown in Table 2.5.

Table 2.5 The axis predictivity of the PCA biplot of the olive oil data.

Acidity	Peroxide	K232	K270	DK	Yellow	Green	Brown
0.935	0.993	0.997	0.987	0.635	0.978	0.940	0.981

Glossy	Transp	Syrup
0.998	0.997	0.999

Each of these axes represents the original data quite well, with the Syrup axis having the highest predictive power of 0.999. However, the DK axis has the lowest predictivity power of 0.635. This means that the axis represents the original data, but not quite as well as the other axes. Likewise, the predictivity of each sample is estimated and is shown in Table 2.6. All sixteen samples are in the biplot plane of approximation. These predictivities along with the overall quality value indicate that the PCA biplot approximates the olive oil data very well.

Table 2.6 The sample predictivity of the PCA biplot of the olive oil data.

G1	G2	G3	G4	G5	I1	I2	I3
1	1	1	1	1	1	1	1

I4	I5	S1	S2	S3	S4	S5	S6
1	1	1	1	1	1	1	1

2.13 Summary

The biplot, a joint graphical display of the rows and of columns of a data matrix, is often referred to as the multivariate version of a scatter plot, because it allows for the display of rows (samples) as points and each column (variable) by an axis on the plot. An advantage of the biplot includes the revelation of the association between rows and columns of a data matrix.

If the roles of the columns and rows of a data matrix are not interchangeable, the resulting biplot will be an asymmetric biplot.

The simplest form of an asymmetric biplot is the PCA biplot. To construct this biplot, the best two-dimensional plane, passing through the origin, is first obtained and the orthogonal projections of each point in the high-dimensional plot of the data matrix onto this plane are done. Afterwards, the plane, with the projected points on it, is extracted from the high-dimensional

space to the scaffolding axes on which the biplot is built. These scaffolding axes are not shown, but are defined by the first two columns of the loading matrix **V**. In the PCA biplot, points are defined by the principal components scores (**Z**), while the directions of the biplot axes are calculated using the rows of the loading matrix **V**.

To evaluate how good the PCA biplot representation is, the overall quality of approximation along with the sample and axis predictivities are required. The latter is measured by expressing the sum-of-squares of the approximated values for each variable as a proportion of their respective total sum-of-squares. Likewise, sample predictivity is measured by expressing the sum-of-squares of the approximated values for each sample as a proportion of their respective total sum-of-squares. Overall, the quality of approximation is measured in terms of the percentage of variation in the data matrix explained by the extracted principal components scores (**Z**).

Next, in Chapter 3, another popular multivariate technique that can be used to approximate large data matrices is discussed.

CHAPTER 3

PARTIAL LEAST SQUARES REGRESSION

3.1 Introduction

One of the most frequently asked questions in data analysis is how to model one or more response variables using a set of predictor variables. In the field of economics, for example, how can the supply and demand of commodities be modelled, using their prices and economic conditions? In chemistry, how can the properties of chemical samples be modelled, using their chemical composition? In microarray and gene expression, how can the type of patients' tissues be modelled, using their gene expression levels? And in quantitative structure activity relationship studies, how can the quality and quantity of manufactured products be modelled, using the conditions of the manufacturing process (Wold *et al*., 2001)? Usually the response variables are modelled by means of Multivariate Multiple Linear Regression (MMLR), which works well as long as the predictors are fairly few and are poorly correlated to each other. However, with modern day measuring instruments, data can be very large, strongly correlated and sometimes incomplete. As a result, MMLR cannot be used in such cases, unless a careful variable selection is carried out. Several methods such as Principal Component Regression (PCR) and Partial Least Squares Regression (PLSR) can be useful tools for such modelling (Abdi, 2010; Martens & Naes, 1989). When modelling the responses, PCR incorporates PCA. More specifically, it uses PCA to extract a set of principal components from the (large set of) predictors and takes these components as predictors in the modelling of the responses. PCA aims to represent the variation in the predictors optimally, without taking into account their relationship with the responses. Conversely, PLSR uses a set of orthogonal latent variables, which have been chosen to represent the relationship between predictors and responses optimally, as predictors in the modelling of the responses. These latent variables are extracted from the (large set of) predictors using PLS. The central idea of both methods is to approximate the large set of predictors by a few components and then to regress the responses against these components. In this chapter, an outline is presented of the necessary theory behind PLS and PLSR, and PLSR is briefly compared to MMLR and PCR.

3.2 Notation

The notation introduced in Chapter 1 is applicable in this chapter.

3.3 The goal of PLS

In general, PLS searches for a set of orthogonal factors, components or latent variables, that perform a simultaneous decomposition of $\mathbf{X}$ and $\mathbf{Y}$, with the restriction that these components explain as much as possible of the covariance between $\mathbf{X}$ and $\mathbf{Y}$. Thus, the PLS model aims to find a few ($A < P$) latent variables called X-scores. These scores are denoted by $\mathbf{T}$ and, since the latent variables are orthogonal, $\mathbf{T}^{T}\mathbf{T} = \mathbf{I}_{A}$.

3.4 A brief review of the literature on PLS

PLS was first developed by Wold (1966) in econometrics, but became popular in chemometrics (Wold, 2001). First presented as an algorithm similar to the power method used for computing eigenvectors, PLS was quickly interpreted in a statistical framework (Phatak & De Jong, 1997; Tenenhaus, 1998; Ter Braak & De Jong, 1998), and then became the tool of choice as a multivariate technique for non-experimental and experimental data like neuroimaging (McIntosh & Lobaugh, 2004). Due to its popularity in chemometrics, PLS is popular in industries that collect (very) large correlated data. It is also popularly used for the purpose of monitoring and controlling of industrial processes, among others. To date, PLS has been used in several disciplines for various areas of interest. These have included the application of PLS in sensory evaluation (Martens & Naes, 1989), social sciences (Tobias, 1997), microarray and gene expression (Boulesteix & Strimmer, 2005; Johansson *et al*., 2003), chemometrics (Helland, 1988; Wold, 2001), biological sciences (Palermo *et al*., 2009), behaviour and brain sciences (Krishnan *et al*., 2011), quality control (Wold *et al*., 2001), psychology, medicine and pharmaceutical science (Wold, 1994).

3.5 PLS algorithms

To compute the latent variables, several algorithms have been developed (Abdi, 2010; De Jong, 1993; Golub & Kahan, 1965; Lindgren *et al*., 1993; Martins *et al*., 2010; Rännar *et al*., 1994; Wold *et al*., 1984; Wold *et al*., 2001). Among the most used algorithms are the Nonlinear Iterative PArtial Least Squares (NIPALS), kernel, and Statistical Inspired Modification to Partial Least Squares (SIMPLS) algorithms. All these algorithms are based on some iterative procedures. Let $a = 1, 2, \ldots, A$ represent the successive latent variables. The first step is to column centre both the $\mathbf{X}$ and $\mathbf{Y}$ matrices by subtracting the mean of each variable respectively. Let $\mathbf{X}_0$ and $\mathbf{Y}_0$ denote the centred matrices. The columns of the weights ($\mathbf{W}$, $\mathbf{R}$ and $\mathbf{C}$), scores ($\mathbf{T}$ and $\mathbf{U}$) and loadings ($\mathbf{P}$ and $\mathbf{Q}$) matrices are then calculated successively. The different

algorithms can result in different PLS decompositions, but in all cases, $\mathbf{T} = \mathbf{X}_0\mathbf{R}$. The objective function of PLS can be formulated as follows. For $a = 1, 2, \ldots, A$,

$$\text{maximize } \operatorname{cov}(\mathbf{w}_a^T\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w}_a) = \operatorname{cov}(\mathbf{t}_a^T\mathbf{Y}\mathbf{Y}^T\mathbf{t}_a)$$
$$\text{subject to } \mathbf{t}_a^T\mathbf{t}_a = \mathbf{w}_a^T\mathbf{X}^T\mathbf{X}\mathbf{w}_a = 1 \text{ and } \mathbf{t}_a^T\mathbf{t}_b = \mathbf{w}_a^T\mathbf{X}^T\mathbf{X}\mathbf{w}_b = 0,$$

where $\mathbf{t}_a = \mathbf{X}\mathbf{w}_a$, $b = 1, 2, \cdots, A-1$ and $b \neq a$. This function is solved using several iterative algorithms such as the NIPALS and kernel algorithms. For the SIMPLS algorithm, the objective function can be expressed as

$$\text{maximize } \operatorname{cov}(\mathbf{r}_a^T\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{r}_a) = \operatorname{cov}(\mathbf{t}_a^T\mathbf{Y}\mathbf{Y}^T\mathbf{t}_a)$$
$$\text{subject to } \mathbf{t}_a^T\mathbf{t}_a = \mathbf{r}_a^T\mathbf{X}^T\mathbf{X}\mathbf{r}_a = 1 \text{ and } \mathbf{t}_a^T\mathbf{t}_b = \mathbf{r}_a^T\mathbf{X}^T\mathbf{X}\mathbf{r}_b = 0$$

where $\mathbf{t}_a = \mathbf{X}\mathbf{r}_a$. Three algorithms implemented in the R package **pls** by Mevik & Wehrens (2007) are discussed below.

3.5.1 *NIPALS algorithm*

This algorithm starts with $\mathbf{X}$ and $\mathbf{Y}$ column centred. According to Abdi (2010), the iterative process begins with the first PLS score vector $\mathbf{u}_1$ $(N \times 1)$ defined by random values. It can also be chosen as one of the columns in $\mathbf{Y}_0$, for example, the one having the maximum variance. The PLS weight and score vectors for the X-variables are then obtained as $\mathbf{w}_1 = \mathbf{X}_0^T\mathbf{u}_1$ and $\mathbf{t}_1 = \mathbf{X}_0\mathbf{w}_1$ respectively. Both $\mathbf{w}_1$ and $\mathbf{t}_1$ are later normalized such that $\|\mathbf{w}_1\| = 1$ and $\|\mathbf{t}_1\| = 1$. After this step, the weight vector for the Y-variables is calculated as $\mathbf{c}_1 = \mathbf{Y}_0^T\mathbf{t}_1$ and $\mathbf{u}_1$ is updated as $\mathbf{u}_1 = \mathbf{Y}_0\mathbf{c}_1$. These iteration steps for obtaining the first PLS vectors are repeated until $\mathbf{t}_1$ converges. Convergence is determined with respect to the change in $\mathbf{t}_1$ and is reached when this change is small (Wold *et al.*, 2001). Subsequently, the matrices $\mathbf{X}_0$ and $\mathbf{Y}_0$ are updated as $\mathbf{X}_0 = \mathbf{X}_0 - \mathbf{t}_1\mathbf{p}_1^T$ and $\mathbf{Y}_0 = \mathbf{Y}_0 - \mathbf{t}_1\mathbf{c}_1^T$, where $\mathbf{p}_1 = \mathbf{X}_0^T\mathbf{t}_1$ represents the loading of $\mathbf{t}_1$ on the X-variables. The updated matrices are then used in the computation of the next set of PLS vectors. The NIPALS algorithm is summarized as follows.

Algorithm 3.1:

(1) $\mathbf{X}_1 = \mathbf{X}_0$, $\mathbf{Y}_1 = \mathbf{Y}_0$ and $a = 1$.

(2) Choose $\mathbf{u}_1$ $(N \times 1)$ as one of the columns in $\mathbf{Y}_0$ or as a vector with random values.

(3) Compute $\mathbf{w}_a$ $(P \times 1)$, $\mathbf{t}_a$ $(N \times 1)$ and $\mathbf{c}_a$ $(M \times 1)$ as

$$\mathbf{w}_a = \mathbf{X}_a^T\mathbf{u}_a / \|\mathbf{X}_a^T\mathbf{u}_a\|$$
$$\mathbf{t}_a = \mathbf{X}_a\mathbf{w}_a / \|\mathbf{X}_a\mathbf{w}_a\|$$
$$\mathbf{c}_a = \mathbf{Y}_a^T\mathbf{t}_a / \|\mathbf{Y}_a^T\mathbf{t}_a\|.$$

(4) Update $\mathbf{u}_a$ $(N \times 1)$ as

$$\mathbf{u}_a = \mathbf{Y}_a\mathbf{c}_a / \|\mathbf{Y}_a\mathbf{c}_a\|.$$

(5) Check whether $\mathbf{t}_a$ converges, i.e., is $\|\mathbf{t}_{1_{new}} - \mathbf{t}_{1_{old}}\| / \|\mathbf{t}_{1_{new}}\| < 10^{-8}$? If not, return to step (3).

(6) Compute $\mathbf{p}_a$ $(P \times 1)$ and $\mathbf{q}_a$ $(M \times 1)$ as

$$\mathbf{p}_a = \mathbf{X}_a^T \mathbf{t}_a$$
$$\mathbf{q}_a = \mathbf{Y}_a^T \mathbf{t}_a.$$

(7) Update $\mathbf{X}_a$ $(N \times P)$ and $\mathbf{Y}_a$ $(N \times M)$, for the next latent variable, as

$$\mathbf{X}_{a+1} = \mathbf{X}_a - \mathbf{t}_a \mathbf{p}_a^T$$
$$\mathbf{Y}_{a+1} = \mathbf{Y}_a - \mathbf{t}_a \mathbf{c}_a^T.$$

(8) Return to step (3), with $a = a + 1$, to compute the next latent variable until $a = A$ latent variables have been reached.

(9) Store $\mathbf{w}_a$, $\mathbf{c}_a$, $\mathbf{t}_a$, $\mathbf{u}_a$, $\mathbf{p}_a$ and $\mathbf{q}_a$ into the successive columns of $\mathbf{W}$, $\mathbf{C}$, $\mathbf{T}$, $\mathbf{U}$, $\mathbf{P}$ and $\mathbf{Q}$.

3.5.2 *Kernel algorithm*

The iterations in steps (3) to (5) of the NIPALS algorithm can be time-consuming, especially when dealing with very large data. For this reason, a quick and efficient algorithm, called the *kernel algorithm*, was developed (Lindgren *et al*., 1993; Rännar *et al*., 1994). *The algorithm developed by Lindgren et al. (1993) handles large data with few(er) variables, while Rännar et al. (1994) developed an algorithm suitable for handling large data with many variables but fewer samples*. From $\mathbf{c}_a = \mathbf{Y}_a^T \mathbf{t}_a$ and $\mathbf{t}_a = \mathbf{X}_a \mathbf{w}_a$, in the NIPALS algorithm,

$$\mathbf{u}_a = \mathbf{Y}_a \mathbf{c}_a = \mathbf{Y}_a \mathbf{Y}_a^T \mathbf{t}_a = \mathbf{Y}_a \mathbf{Y}_a^T \mathbf{X}_a \mathbf{w}_a = \mathbf{Y}_a \mathbf{Y}_a^T \mathbf{X}_a \mathbf{X}_a^T \mathbf{u}_a \quad (3.1)$$

where $\mathbf{w}_a = \mathbf{X}_a^T \mathbf{u}_a$. Likewise, vectors $\mathbf{w}_a$, $\mathbf{t}_a$ and $\mathbf{c}_a$ can be formulated as

$$\mathbf{w}_a = \mathbf{X}_a^T \mathbf{u}_a = \mathbf{X}_a^T \mathbf{Y}_a \mathbf{c}_a = \mathbf{X}_a^T \mathbf{Y}_a \mathbf{Y}_a^T \mathbf{t}_a = \mathbf{X}_a^T \mathbf{Y}_a \mathbf{Y}_a^T \mathbf{X}_a \mathbf{w}_a$$
$$\mathbf{t}_a = \mathbf{X}_a \mathbf{w}_a = \mathbf{X}_a \mathbf{X}_a^T \mathbf{u}_a = \mathbf{X}_a \mathbf{X}_a^T \mathbf{Y}_a \mathbf{c}_a = \mathbf{X}_a \mathbf{X}_a^T \mathbf{Y}_a \mathbf{Y}_a^T \mathbf{t}_a$$
$$\mathbf{c}_a = \mathbf{Y}_a^T \mathbf{t}_a = \mathbf{Y}_a^T \mathbf{X}_a \mathbf{w}_a = \mathbf{Y}_a^T \mathbf{X}_a \mathbf{X}_a^T \mathbf{u}_a = \mathbf{Y}_a^T \mathbf{X}_a \mathbf{X}_a^T \mathbf{Y}_a \mathbf{c}_a. \quad (3.2)$$

These equations, (3.1) and (3.2), form the core of the power algorithm for calculating eigenvectors and they can be written in such a way that vectors $\mathbf{u}_a$, $\mathbf{w}_a$, $\mathbf{t}_a$ and $\mathbf{c}_a$ are determined as eigenvectors to a set of variance-covariance matrices (Höskuldsson, 1988). That is,

$$\mathbf{Y}_a \mathbf{Y}_a^T \mathbf{X}_a \mathbf{X}_a^T \mathbf{u}_a = \phi_1 \mathbf{u}_a$$
$$\mathbf{X}_a^T \mathbf{Y}_a \mathbf{Y}_a^T \mathbf{X}_a \mathbf{w}_a = \phi_2 \mathbf{w}_a$$
$$\mathbf{X}_a \mathbf{X}_a^T \mathbf{Y}_a \mathbf{Y}_a^T \mathbf{t}_a = \phi_3 \mathbf{t}_a$$
$$\mathbf{Y}_a^T \mathbf{X}_a \mathbf{X}_a^T \mathbf{Y}_a \mathbf{c}_a = \phi_4 \mathbf{c}_a. \quad (3.3)$$

According to the power algorithm for finding the EigenValue-Vector Decomposition (EV-VD) of a matrix, ϕ_1 to ϕ_4 in (3.3) are the maximum eigenvalues of the eigenproblems. The vectors $\mathbf{u}_a$, $\mathbf{w}_a$, $\mathbf{t}_a$ and $\mathbf{c}_a$ in (3.3) are the associated eigenvectors of the respective variance-covariance matrices corresponding to ϕ_1, ϕ_2, ϕ_3 and ϕ_4. The values of ϕ_1 to ϕ_4 are of no interest in this chapter. In general, the kernel algorithm replaces the power algorithm with a single

eigendecomposition step. With $\mathbf{X}$ and $\mathbf{Y}$ column centred, two association matrices $\mathbf{X}_0\mathbf{X}_0^{\mathrm{T}}$ (N × N) and $\mathbf{Y}_0\mathbf{Y}_0^{\mathrm{T}}$ (N × N) are created. These two matrices are then multiplied together to form the kernel matrix $\mathbf{X}_0\mathbf{X}_0^{\mathrm{T}}\mathbf{Y}_0\mathbf{Y}_0^{\mathrm{T}}$ (N × N). Here, the PLS score vector $\mathbf{t}_1$ is obtained as the first eigenvector from the EV-VD of $\mathbf{X}_0\mathbf{X}_0^{\mathrm{T}}\mathbf{Y}_0\mathbf{Y}_0^{\mathrm{T}}$, while the first score vector $\mathbf{u}_1$ for the Y-variables is obtained as $\mathbf{Y}_0\mathbf{Y}_0^{\mathrm{T}}\,\mathbf{t}_1$. After these first score vectors have been obtained, $\mathbf{X}_0\mathbf{X}_0^{\mathrm{T}}$ and $\mathbf{Y}_0\mathbf{Y}_0^{\mathrm{T}}$ are updated using the matrix $\mathbf{L}_1$ (N × N) $= \mathbf{I}_{\mathrm{N}} - \mathbf{t}_1\mathbf{t}_1^{\mathrm{T}}$. That is, $\mathbf{X}_0\mathbf{X}_0^{\mathrm{T}} = \mathbf{L}_1\mathbf{X}_0\mathbf{X}_0^{\mathrm{T}}\mathbf{L}_1$ and $\mathbf{Y}_0\mathbf{Y}_0^{\mathrm{T}} = \mathbf{L}_1\mathbf{Y}_0\mathbf{Y}_0^{\mathrm{T}}\mathbf{L}_1$. These updated matrices are later used in the computation of the next set of PLS score vectors. Once A sets of score vectors have been calculated, the loadings matrix for the X-variables is obtained by $\mathbf{P} = \mathbf{X}_0^{\mathrm{T}}\mathbf{T}$, for $\mathbf{T} = [\mathbf{t}_1 \quad \mathbf{t}_2 \quad \cdots \quad \mathbf{t}_{\mathrm{A}}]$. Similarly, the loadings matrix for the Y-variables is obtained by $\mathbf{Q} = \mathbf{Y}_0^{\mathrm{T}}\mathbf{T}$. This is the kernel algorithm of Rännar *et al*. (1994) and it can be summarized as follows.

Algorithm 3.2:

(1) $(\mathbf{X}\mathbf{X}^{\mathrm{T}})_1 = \mathbf{X}_0\mathbf{X}_0^{\mathrm{T}}$, $(\mathbf{Y}\mathbf{Y}^{\mathrm{T}})_1 = \mathbf{Y}_0\mathbf{Y}_0^{\mathrm{T}}$ and $\mathrm{a} = 1$.

(2) Compute $\mathbf{t}_{\mathrm{a}}$ (N × 1) and $\mathbf{u}_{\mathrm{a}}$ (N × 1) as

$$\left((\mathbf{X}\mathbf{X}^{\mathrm{T}})_{\mathrm{a}}(\mathbf{Y}\mathbf{Y}^{\mathrm{T}})_{\mathrm{a}} \right) = \mathbf{J}\Delta\mathbf{J}^{\mathrm{T}}, \text{ by the EV-VD}$$
$$\mathbf{t}_{\mathrm{a}} = \mathbf{J}[,\ 1]$$
$$\mathbf{u}_{\mathrm{a}} = (\mathbf{Y}\mathbf{Y}^{\mathrm{T}})_{\mathrm{a}}\,\mathbf{t}_{\mathrm{a}}.$$

(3) Update the association matrices $(\mathbf{X}\mathbf{X}^{\mathrm{T}})_{\mathrm{a}}$ (N × N) and $(\mathbf{Y}\mathbf{Y}^{\mathrm{T}})_{\mathrm{a}}$ (N × N), for the next latent variable, as

$$\mathbf{L}_{\mathrm{a}} = (\mathbf{I}_{\mathrm{N}} - \mathbf{t}_{\mathrm{a}}\mathbf{t}_{\mathrm{a}}^{\mathrm{T}})$$
$$(\mathbf{X}\mathbf{X}^{\mathrm{T}})_{\mathrm{a}+1} = \mathbf{L}_{\mathrm{a}}\,(\mathbf{X}\mathbf{X}^{\mathrm{T}})_{\mathrm{a}}\,\mathbf{L}_{\mathrm{a}}$$
$$(\mathbf{Y}\mathbf{Y}^{\mathrm{T}})_{\mathrm{a}+1} = \mathbf{L}_{\mathrm{a}}\,(\mathbf{Y}\mathbf{Y}^{\mathrm{T}})_{\mathrm{a}}\,\mathbf{L}_{\mathrm{a}}.$$

(4) Return to step (2), with $\mathrm{a} = \mathrm{a} + 1$, to compute the next latent variable until $\mathrm{a} = \mathrm{A}$ latent variables have been reached.

(5) Store $\mathbf{t}_{\mathrm{a}}$ and $\mathbf{u}_{\mathrm{a}}$ into the successive columns of $\mathbf{T}$ and $\mathbf{U}$.

(6) Compute $\mathbf{P}$ (P × A) and $\mathbf{Q}$ (M × A) as

$$\mathbf{P} = \mathbf{X}_0^{\mathrm{T}}\mathbf{T}$$
$$\mathbf{Q} = \mathbf{Y}_0^{\mathrm{T}}\mathbf{T}.$$

For (very) large data with few(er) variables, the kernel algorithm of Lindgren *et al*. (1993) is more appropriate. This algorithm is similar to the one by Rännar *et al*. (1994), except, instead of working with the kernel matrix $\mathbf{X}_0\mathbf{X}_0^{\mathrm{T}}\mathbf{Y}_0\mathbf{Y}_0^{\mathrm{T}}$ (N × N), matrix $\mathbf{X}_0^{\mathrm{T}}\mathbf{Y}_0$ (P × M) and its transposed $\mathbf{Y}_0^{\mathrm{T}}\mathbf{X}_0$ (M × P) are multiplied together to form a new kernel matrix $\mathbf{X}_0^{\mathrm{T}}\mathbf{Y}_0\mathbf{Y}_0^{\mathrm{T}}\mathbf{X}_0$ (P × P) on which the first PLS weight vector $\mathbf{w}_1$ is calculated. The Lindgren *et al*. (1993) algorithm is summarized below.

Algorithm 3.3:

(1) $(\mathbf{X}^{\mathrm{T}}\mathbf{X})_1 = \mathbf{X}_0^{\mathrm{T}}\mathbf{X}_0$, $(\mathbf{X}^{\mathrm{T}}\mathbf{Y})_1 = \mathbf{X}_0^{\mathrm{T}}\mathbf{Y}_0$ and $\mathrm{a} = 1$.

(2) Compute $\mathbf{w}_\mathrm{a}$ $(\mathrm{P} \times 1)$, $\mathbf{p}_\mathrm{a}$ $(\mathrm{P} \times 1)$ and $\mathbf{q}_\mathrm{a}$ $(\mathrm{M} \times 1)$ as

$$\begin{aligned} \left((\mathbf{X}^{\mathrm{T}}\mathbf{Y})_\mathrm{a} (\mathbf{Y}^{\mathrm{T}}\mathbf{X})_\mathrm{a} \right) &= \mathbf{K}\Theta\mathbf{K}^{\mathrm{T}}, \text{ by the EV-VD} \\ \mathbf{w}_\mathrm{a} &= \mathbf{K}[,\ 1] \\ \mathbf{p}_\mathrm{a} &= (\mathbf{X}^{\mathrm{T}}\mathbf{X})_\mathrm{a}\mathbf{w}_\mathrm{a} \\ \mathbf{q}_\mathrm{a} &= (\mathbf{Y}^{\mathrm{T}}\mathbf{X})_\mathrm{a}\mathbf{w}_\mathrm{a}. \end{aligned}$$

(3) Update $(\mathbf{X}^{\mathrm{T}}\mathbf{X})_\mathrm{a}$ $(\mathrm{P} \times \mathrm{P})$ and $(\mathbf{X}^{\mathrm{T}}\mathbf{Y})_\mathrm{a}$ $(\mathrm{P} \times \mathrm{M})$, for the next latent variable, as

$$\begin{aligned} \mathbf{O}_\mathrm{a} &= (\mathbf{I}_\mathrm{P} - \mathbf{w}_\mathrm{a}\mathbf{p}_\mathrm{a}^{\mathrm{T}}) \\ (\mathbf{X}^{\mathrm{T}}\mathbf{X})_{\mathrm{a}+1} &= \mathbf{O}_\mathrm{a}^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X})_\mathrm{a}\mathbf{O}_\mathrm{a} \\ (\mathbf{X}^{\mathrm{T}}\mathbf{Y})_{\mathrm{a}+1} &= \mathbf{O}_\mathrm{a}^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{Y})_\mathrm{a}. \end{aligned}$$

(4) Return to step (2), with $\mathrm{a} = \mathrm{a} + 1$, to compute the next latent variable until $\mathrm{a} = \mathrm{A}$ latent variables have been reached.

(5) Store $\mathbf{w}_\mathrm{a}$, $\mathbf{p}_\mathrm{a}$ and $\mathbf{q}_\mathrm{a}$ into the successive columns of $\mathbf{W}$, $\mathbf{P}$ and $\mathbf{Q}$.

(6) Compute $\mathbf{T}$ $(\mathrm{N} \times \mathrm{A})$ as

$$\mathbf{T} = \mathbf{X}_0\mathbf{W}.$$

3.5.3 *SIMPLS algorithm*

Another method for computing the latent variables is the SIMPLS algorithm. Proposed by De Jong (1993), this algorithm derives the latent variables directly as linear combinations of the original centred X-variables. An advantage of this algorithm is that it is not compulsory to update $\mathbf{X}_0$ or $\mathbf{Y}_0$, which may result in a faster computation (Martins *et al*., 2010). This algorithm starts with $\mathbf{X}$ and $\mathbf{Y}$ column centred and the covariance matrix $(\mathrm{N}-1)\mathbf{S} = \mathbf{X}_0^{\mathrm{T}}\mathbf{Y}_0$. The first PLS weight vectors $\mathbf{r}_1$ and $\mathbf{c}_1$ are computed in such a way that they can be applied directly to the centred data. De Jong (1993) defined $\mathbf{r}_1$ and $\mathbf{c}_1$ as the first left and right singular vectors from the SVD of $\mathbf{S}$ respectively. The first score and loading vectors of $\mathbf{X}$ are then calculated as $\mathbf{t}_1 = \mathbf{X}_0\mathbf{r}_1$ and $\mathbf{p}_1 = \mathbf{X}_0^{\mathrm{T}}\mathbf{t}_1$ respectively. After these vectors have been obtained, $\mathbf{S}$ is updated and used in the computation of the next set of PLS vectors. This update is done by projecting $\mathbf{S}$ onto a subspace orthogonal to $\mathbf{p}_1$. That is, $\mathbf{S} = \mathbf{S} - \mathbf{p}_1(\mathbf{p}_1^{\mathrm{T}}\mathbf{p}_1)^{-1}\mathbf{p}_1^{\mathrm{T}}\mathbf{S}$. The SIMPLS algorithm is summarized as follows.

Algorithm 3.4:

(1) $\mathbf{S}_1 = \mathbf{X}_0^{\mathrm{T}}\mathbf{Y}_0$ and $\mathrm{a} = 1$.

(2) Compute $\mathbf{c}_\mathrm{a}$ $(\mathrm{M} \times 1)$, $\mathbf{r}_\mathrm{a}$ $(\mathrm{P} \times 1)$, $\mathbf{t}_\mathrm{a}$ $(\mathrm{N} \times 1)$, $\mathbf{p}_\mathrm{a}$ $(\mathrm{P} \times 1)$ and $\mathbf{q}_\mathrm{a}$ $(\mathrm{M} \times 1)$ as

$$\begin{aligned} \mathbf{S}_\mathrm{a} &= \mathbf{D}\Lambda\mathbf{F}^{\mathrm{T}}, \text{ by the SVD} \\ \mathbf{c}_\mathrm{a} &= \mathbf{F}[,\ 1] \end{aligned}$$

$$\mathbf{r}_a = \mathbf{D}[,\ 1]$$
$$\mathbf{t}_a = \mathbf{X}_0\mathbf{r}_a / \|\mathbf{X}_0\mathbf{r}_a\|$$
$$\mathbf{p}_a = \mathbf{X}_0^T\mathbf{t}_a$$
$$\mathbf{q}_a = \mathbf{Y}_0^T\mathbf{t}_a.$$

(3) Update the covariance matrix $\mathbf{S}_a$ (P × M), for the next latent variable, as
$$\mathbf{S}_{a+1} = (\mathbf{I}_P - \mathbf{p}_a(\mathbf{p}_a^T\mathbf{p}_a)^{-1}\mathbf{p}_a^T)\mathbf{S}_a.$$

(4) Return to step (2), with $a = a + 1$, to compute the next latent variable until $a = A$ latent variables have been reached.

(5) Store $\mathbf{c}_a$, $\mathbf{r}_a$, $\mathbf{t}_a$, $\mathbf{p}_a$ and $\mathbf{q}_a$ into the successive columns of $\mathbf{C}$, $\mathbf{R}$, $\mathbf{T}$, $\mathbf{P}$ and $\mathbf{Q}$.

The X-weights matrix in the SIMPLS algorithm is defined as $\mathbf{R}$, while in the NIPALS and kernel algorithms, it is defined as $\mathbf{W}$. The difference between these two matrices is that the linear combinations defined by $\mathbf{W}$ are for the updated (centred) X-variables, while the linear combinations defined by $\mathbf{R}$ are for the original (centred) X-variables. The matrix $\mathbf{W}$ needs to be converted in such a way that its linear combinations are now defined for the original (centred) X-variables. With $\mathbf{X}_1 = \mathbf{X}_0$,

$$\mathbf{t}_1 = \mathbf{X}_1\mathbf{w}_1 = \mathbf{X}_0\mathbf{w}_1 \quad (3.4)$$

which makes $\mathbf{r}_1 = \mathbf{w}_1$. For $a = 2$, $\mathbf{t}_2 = \mathbf{X}_2\mathbf{w}_2$. Since $\mathbf{X}_{a+1} = \mathbf{X}_a - \mathbf{t}_a\mathbf{p}_a^T$, for $a = 2, 3, \cdots, A$, $\mathbf{X}_2 = \mathbf{X}_1 - \mathbf{t}_1\mathbf{p}_1^T = \mathbf{X}_0 - \mathbf{t}_1\mathbf{p}_1^T$. Then,

$$\mathbf{t}_2 = \mathbf{X}_2\mathbf{w}_2 = (\mathbf{X}_0 - \mathbf{t}_1\mathbf{p}_1^T)\mathbf{w}_2. \quad (3.5)$$

Substituting (3.4) into (3.5) yields

$$\mathbf{t}_2 = (\mathbf{X}_0 - \mathbf{X}_0\mathbf{w}_1\mathbf{p}_1^T)\mathbf{w}_2 = \mathbf{X}_0(\mathbf{I}_P - \mathbf{w}_1\mathbf{p}_1^T)\mathbf{w}_2. \quad (3.6)$$

This makes $\mathbf{r}_2 = (\mathbf{I}_P - \mathbf{w}_1\mathbf{p}_1^T)\mathbf{w}_2$. Likewise, for $a = 3$,

$$\mathbf{X}_3 = \mathbf{X}_2 - \mathbf{t}_2\mathbf{p}_2^T = \mathbf{X}_0 - \mathbf{t}_1\mathbf{p}_1^T - \mathbf{t}_2\mathbf{p}_2^T$$

and

$$\mathbf{t}_3 = \mathbf{X}_3\mathbf{w}_3 = (\mathbf{X}_0 - \mathbf{t}_1\mathbf{p}_1^T - \mathbf{t}_2\mathbf{p}_2^T)\mathbf{w}_3.$$

From (3.4) and (3.5),

$$\mathbf{t}_3 = (\mathbf{X}_0 - \mathbf{X}_0\mathbf{w}_1\mathbf{p}_1^T - (\mathbf{X}_0 - \mathbf{X}_0\mathbf{w}_1\mathbf{p}_1^T)\mathbf{w}_2\mathbf{p}_2^T)\mathbf{w}_3$$

which simplifies to

$$\mathbf{t}_3 = \mathbf{X}_0(\mathbf{I}_P - \mathbf{w}_1\mathbf{p}_1^T)(\mathbf{I}_P - \mathbf{w}_2\mathbf{p}_2^T)\mathbf{w}_3. \quad (3.7)$$

Therefore, $\mathbf{r}_3 = (\mathbf{I}_P - \mathbf{w}_1\mathbf{p}_1^T)(\mathbf{I}_P - \mathbf{w}_2\mathbf{p}_2^T)\mathbf{w}_3$. Thus, for $a = A$,

$$\mathbf{t}_A = \mathbf{X}_A\mathbf{w}_A = \mathbf{X}_0(\mathbf{I}_P - \mathbf{w}_1\mathbf{p}_1^T)(\mathbf{I}_P - \mathbf{w}_2\mathbf{p}_2^T)\ \cdots\ (\mathbf{I}_P - \mathbf{w}_{A-1}\mathbf{p}_{A-1}^T)\mathbf{w}_A$$

and

$$\mathbf{r}_A = (\mathbf{I}_P - \mathbf{w}_1\mathbf{p}_1^T)(\mathbf{I}_P - \mathbf{w}_2\mathbf{p}_2^T)\ \cdots\ (\mathbf{I}_P - \mathbf{w}_{A-1}\mathbf{p}_{A-1}^T)\mathbf{w}_A. \quad (3.8)$$

De Jong (1993) suggested that (3.8) can be represented as $\mathbf{r}_a = \mathbf{\Omega}_a\mathbf{w}_a$, such that $\mathbf{\Omega}_1$ (P × P) = $\mathbf{I}_P$ and $\mathbf{\Omega}_{a+1} = \mathbf{\Omega}_a(\mathbf{I}_P - \mathbf{w}_a\mathbf{p}_a^T)$. Since $\mathbf{r}_a = \mathbf{\Omega}_a\mathbf{w}_a$,

$$\mathbf{\Omega}_{a+1} = \mathbf{\Omega}_a - \mathbf{\Omega}_a\mathbf{w}_a\mathbf{p}_a^T = \mathbf{\Omega}_a - \mathbf{r}_a\mathbf{p}_a^T, \text{ for } a = 1, 2, 3, \cdots, A.$$

Hence, for $\mathbf{r}_1 = \mathbf{w}_1$, $\mathbf{\Omega}_1 = \mathbf{I}_P$ and $a = 2, 3, \cdots, A$,

$$\mathbf{r}_a = \mathbf{\Omega}_a \mathbf{w}_a \text{ and } \mathbf{\Omega}_a = \mathbf{\Omega}_{a-1} - \mathbf{r}_{a-1}\mathbf{p}_{a-1}^T \quad (3.9)$$

from which the columns of $\mathbf{R}$ (P × A) are computed successively. Therefore, the matrix $\mathbf{R}$ is defined for the original (centred) X-variables through (3.9). Alternatively, the PLSR model can be used as a means of conversion for $\mathbf{W}$, as shown in the next section.

3.6 Partial Least Squares Regression

Usually the modelling of one Y-variable by means of $\mathbf{X}$ is done by solving the equation $\mathbf{y} = \mathbf{Xb}$, where $\mathbf{b}$ (P × 1) is the unknown coefficient vector estimated as $\hat{\mathbf{b}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. This is the general idea in regression. However, for M Y-variables, the modelling is done by solving the equation

$$\mathbf{Y} = \mathbf{XB}.$$

The unknown coefficients matrix $\mathbf{B}$ (P × M) is estimated by $\hat{\mathbf{B}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$. Therefore, $\mathbf{Y}$ is estimated as

$$\hat{\mathbf{Y}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} = \mathbf{X}\hat{\mathbf{B}}. \quad (3.10)$$

PLSR, on the other hand, models $\mathbf{Y}$ by means of the X-scores $\mathbf{T}$. For this reason, $\mathbf{Y}$ is estimated as

$$\hat{\mathbf{Y}}_0 = \mathbf{T}(\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{Y}_0 = \mathbf{TB} \quad (3.11)$$

where $\mathbf{B} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{Y}_0$.

To convert the matrix $\mathbf{W}$ in such a way that its linear combinations are now defined for the original (centred) X-variables, consider $\mathbf{T} = \mathbf{X}_0\mathbf{W}$, as in the NIPALS algorithm (Algorithm 3.1). Replacing $\mathbf{T}$ in (3.11) by $\mathbf{T} = \mathbf{X}_0\mathbf{W}$ yields,

$$\hat{\mathbf{Y}}_0 = \mathbf{X}_0\mathbf{WB}. \quad (3.12)$$

Here, in (3.12), the estimated regression coefficients matrix $\hat{\mathbf{B}}_{PLSR}$ (P × M) is defined as the product matrix $\mathbf{WB}$. That is,

$$\hat{\mathbf{B}}_{PLSR} = \mathbf{WB} = \mathbf{W}(\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{Y}_0.$$

Since $\mathbf{T} = \mathbf{X}_0\mathbf{W}$,

$$\hat{\mathbf{B}}_{PLSR} = \mathbf{W}(\mathbf{W}^T\mathbf{X}_0^T\mathbf{X}_0\mathbf{W})^{-1}\mathbf{W}^T\mathbf{X}_0^T\mathbf{Y}_0. \quad (3.13)$$

Moreover, with $\mathbf{P} = \mathbf{X}_0^T\mathbf{T} = \mathbf{X}_0^T\mathbf{X}_0\mathbf{W}$ and $\mathbf{Q} = \mathbf{Y}_0^T\mathbf{T} = \mathbf{Y}_0^T\mathbf{X}_0\mathbf{W}$,

$$\mathbf{W}^T\mathbf{X}_0^T\mathbf{X}_0 = \mathbf{P}^T \text{ and } \mathbf{W}^T\mathbf{X}_0^T\mathbf{Y}_0 = \mathbf{Q}^T. \quad (3.14)$$

Substituting (3.14) into (3.13) yields

$$\hat{\mathbf{B}}_{PLSR} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}\mathbf{Q}^T. \quad (3.15)$$

From (3.15), let the product matrix $\mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}$ be denoted by $\mathbf{R}$. That is, $\mathbf{R} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}$. Equation (3.15) is then express-able as $\hat{\mathbf{B}}_{PLSR} = \mathbf{RQ}^T$. Therefore, the matrix $\mathbf{W}$ can be defined for the original (centred) X-variables through

$$\mathbf{R} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}. \quad (3.16)$$

Table 3.1 Summary of the PLSR parameters.

	NIPALS	[+]Kernel	SIMPLS
For $a = 1, 2, \cdots, A$			
$a = 1$	$\mathbf{X}_a = \mathbf{X}_0$ $\mathbf{Y}_a = \mathbf{Y}_0$	$(\mathbf{X}^T\mathbf{X})_a = \mathbf{X}_0^T\mathbf{X}_0$ $(\mathbf{X}^T\mathbf{Y})_a = \mathbf{X}_0^T\mathbf{Y}_0$	$\mathbf{S}_a = \mathbf{X}_0^T\mathbf{Y}_0$
$a > 1$	$\mathbf{X}_a = \mathbf{X}_{a-1} - \mathbf{X}_{a-1}\mathbf{w}_{a-1}\mathbf{p}_{a-1}^T$ [*]$\mathbf{Y}_a = \mathbf{Y}_{a-1} - \mathbf{X}_{a-1}\mathbf{w}_{a-1}\mathbf{q}_{a-1}^T$	$\mathbf{O}_a = (\mathbf{I}_P - \mathbf{w}_{a-1}\mathbf{p}_{a-1}^T)$ $(\mathbf{X}^T\mathbf{X})_a = \mathbf{O}_a^T(\mathbf{X}^T\mathbf{X})_{a-1}\mathbf{O}_a$ $(\mathbf{X}^T\mathbf{Y})_a = \mathbf{O}_a^T(\mathbf{X}^T\mathbf{Y})_{a-1}$	[++]$\mathbf{S}_a = \text{Ð}\mathbf{S}_{a-1}$
Get weight	[#]$\mathbf{u}_a = \mathbf{Y}_a\mathbf{Y}_a^T\mathbf{X}_a\mathbf{X}_a^T\mathbf{u}_{a-1}$ $\mathbf{w}_a = \mathbf{X}_a^T\mathbf{u}_a / \lVert\mathbf{X}_a^T\mathbf{u}_a\rVert$	$\mathbf{w}_a$ = first eigenvector of $(\mathbf{X}^T\mathbf{Y})_a(\mathbf{Y}^T\mathbf{X})_a$	$\mathbf{r}_a$ = first left singular vector of $\mathbf{S}_a$
Compute loadings	$\mathbf{p}_a = \mathbf{X}_a^T\mathbf{X}_a\mathbf{w}_a$ $\mathbf{q}_a = \mathbf{Y}_a^T\mathbf{X}_a\mathbf{w}_a$	$\mathbf{p}_a = (\mathbf{X}^T\mathbf{X})_a\mathbf{w}_a$ $\mathbf{q}_a = (\mathbf{Y}^T\mathbf{X})_a\mathbf{w}_a$	$\mathbf{p}_a = \mathbf{X}_0^T\mathbf{X}_0\mathbf{r}_a$ $\mathbf{q}_a = \mathbf{Y}_0^T\mathbf{X}_0\mathbf{r}_a$
Store	$\mathbf{w}_a$, $\mathbf{p}_a$ and $\mathbf{q}_a$	$\mathbf{w}_a$, $\mathbf{p}_a$ and $\mathbf{q}_a$	$\mathbf{r}_a$, $\mathbf{p}_a$ and $\mathbf{q}_a$
into the successive column of	$\mathbf{W}$, $\mathbf{P}$ and $\mathbf{Q}$	$\mathbf{W}$, $\mathbf{P}$ and $\mathbf{Q}$	$\mathbf{R}$, $\mathbf{P}$ and $\mathbf{Q}$
End.			
Compute scores and coefficients	$\mathbf{R} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}$ $\mathbf{T} = \mathbf{X}_0\mathbf{R}$ $\hat{\mathbf{B}}_{PLSR} = \mathbf{R}\mathbf{Q}^T$	$\mathbf{R} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}$ $\mathbf{T} = \mathbf{X}_0\mathbf{R}$ $\hat{\mathbf{B}}_{PLSR} = \mathbf{R}\mathbf{Q}^T$	$\mathbf{T} = \mathbf{X}_0\mathbf{R}$ $\hat{\mathbf{B}}_{PLSR} = \mathbf{R}\mathbf{Q}^T$

[+] Of Lindgren et al. (1993).

[*] Since $\mathbf{q}_a = \mathbf{Y}_a^T\mathbf{X}_a\mathbf{w}_a = \mathbf{c}_a$.

[++] $\text{Ð} = (\mathbf{I}_P - \mathbf{p}_{a-1}(\mathbf{p}_{a-1}^T\mathbf{p}_{a-1})^{-1}\mathbf{p}_{a-1}^T)$.

[#] $\mathbf{u}_0$ = vector with random values or any column in $\mathbf{Y}_0$.

Equation (3.16) makes matrix $\mathbf{T}$, in terms of the original (centred) X-variables, obtainable as $\mathbf{T} = \mathbf{X}_0\mathbf{R}$. Thus, with matrix $\mathbf{T}$ defined in terms of the original (centred) X-variables through $\mathbf{T} = \mathbf{X}_0\mathbf{R}$, $\mathbf{Y}$, in PLSR, is estimated as

$$\widehat{\mathbf{Y}}_0 = \mathbf{X}_0\widehat{\mathbf{B}}_{\text{PLSR}} \qquad (3.17)$$

where $\widehat{\mathbf{B}}_{\text{PLSR}} = \mathbf{R}\mathbf{Q}^{\text{T}}$ is the estimated coefficients matrix for the PLSR model. Table 3.1 above summarizes the PLSR parameters obtained under the PLS algorithms in Section 3.5.

3.7 Number of PLS components

When extracting the PLS components from a large set of (correlated) predictors, there is a chance of obtaining an over-fitted model for the responses. For this reason, a test for the predictive significance of each PLS component is necessary and this can be used to determine the appropriate number of components to use in the modelling of the responses. Several methods are used to carry out the predictive significance test (Höskuldsson, 1988; Wakeling & Morris, 1993; Wold *et al*., 2001). Among the most used tests are the PRediction Error Sum-of-Squares (PRESS) and Root Mean Squared Error of Prediction (RMSEP) measures.

3.7.1 *PRediction Error Sum-of-Squares measure*

Cross-validation is often applied to test the predictive significance of each PLS component (Höskuldsson, 1988; Wold *et al*., 1993). It is performed by randomly dividing the data (equally) into K groups, e.g., $\text{K} = 5$ to 10, and then developing a number of fitted models from the divided data, with one of the groups deleted. After developing a model, the differences between the actual and fitted Y-values are calculated for the deleted data. The sum-of-squares of these differences are then computed and collected across all the fitted models to form the PRESS values, which estimate the predictive ability of the model. To be precise,

$$\text{PRESS} = \sum_{\text{i}=1}^{\text{N}}\sum_{\text{j}=1}^{\text{M}}\left(y_{\text{ij}} - \hat{y}_{\text{ij}}\right)^2.$$

To determine the significance of each component, the ratio of the PRESS to the sum-of-squares before the current component ($\text{SS}_{\text{a}-1}$) is calculated after each component. Here, $\text{SS}_{\text{a}-1}$ denotes the sum-of-squares before component a. According to Wakeling & Morris (1993), a component is said to be significant if it has a $\text{PRESS}/\text{SS}_{\text{a}-1}$ value smaller than 0.9 for at least one of the Y-variables.

3.7.2 *Root Mean Squared Error of Prediction measure*

Alternatively, the RMSEP values, per cumulative components, can be evaluated to determine the appropriate number of PLS components to use in the final modelling of the responses. This is

done by taking the square root of the average of the PRESS values. That is,

$$\text{RMSEP} = \sqrt{\frac{\text{PRESS}}{\text{N}}}.$$

A plot of the RMSEP values can further assist, graphically, in deciding how many PLS components to use. Here, the optimistically biased error rate is used, rather than a more computing intensive cross-validated version. *The number of components corresponding to the elbow in the* RMSEP *plot can be suggested as the number of components to use in the final model*. For example, in Figure 3.1 below, three components (i.e., $\text{A} = 3$) can be suggested as the final number of components to use in the final PLSR model for this data.

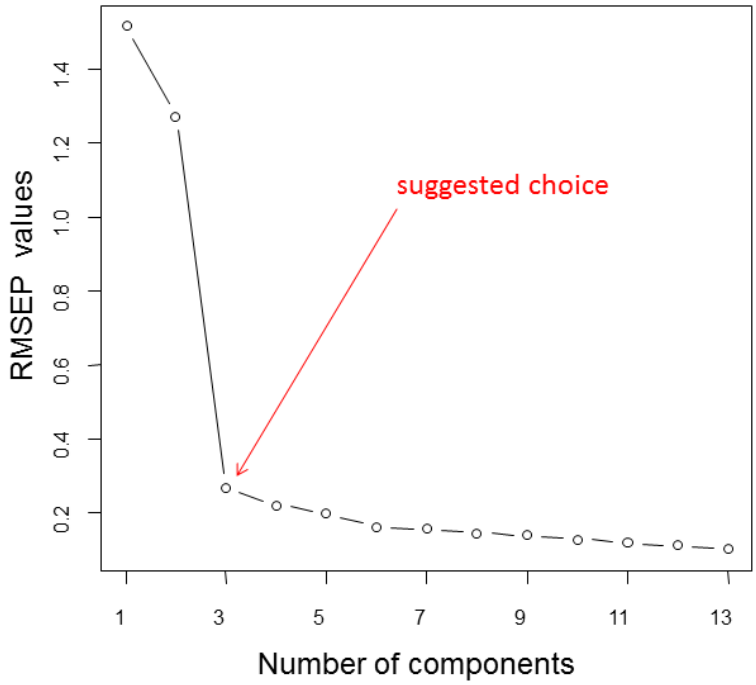

Figure 3.1 The RMSEP plot of an artificial data.

3.8 Example

The following example is an illustration of a PLSR using the olive oil data discussed in Section 2.8. In this data, the five chemical quality measurements (Acidity, Peroxide, K232, K270 and DK) and the six sensory panel characteristics (Yellow, Green, Brown, Glossy, Transp and Syrup) are assigned as the predictor and response variables respectively. As a result, the olive oil data can be viewed as a (16×11) data matrix, comprising of an $\mathbf{X}$ (16×5) matrix as well as a $\mathbf{Y}$ (16×6) matrix. A 5-component PLS was performed using the SIMPLS algorithm, but after the inspection of the RMSEP plot in Figure 3.2 below, two components (i.e., $\text{A} = 2$) can be suggested as the final number of components to use in the modelling of $\mathbf{Y}$. These are shown in Table 3.4. Since the extracted PLS components are uncorrelated to each other, i.e., they are

orthogonal, in Table 3.4, the values obtained under the first component are different from the ones obtained under the second components.

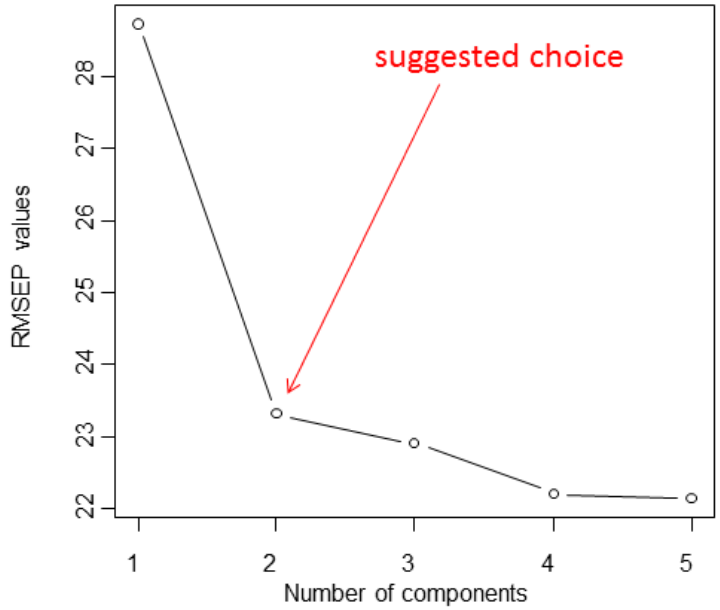

Figure 3.2 The RMSEP plot of the olive oil data.

To see how a typical output of the PLS X-loadings (**P**), X-weights (**R**), Y-loadings (**Q**) and X-scores (**T**) matrices are, using the olive oil data set, see Tables 3.2, 3.3 and 3.4 respectively.

Table 3.2 The weights **R** and loadings **P** values of the chemical quality measurements.

	Component 1	Component 2
Acidity	0.052	0.883
Peroxide	0.994	-0.032
K232	0.092	0.456
K270	0.010	0.109
DK	0.001	0.004

	Component 1	Component 2
Acidity	0.034	0.647
Peroxide	12.956	-0.053
K232	0.844	0.206
K270	0.051	0.052
DK	0.004	0.005

Since $\mathbf{T} = \mathbf{X}_0\mathbf{R}$, the X-weights matrix **R** provides the coefficients of the X-variables to form the PLS components. *Large absolute* X*-weights values can be used to deduce which* X*-variables are important under the specific PLS component*. From the weights matrix in Table 3.2, Peroxide can be said to be important for the first component, while Acidity, K232 and K270 are important for the second component. However, looking at the value for DK, under both components, one can conclude that DK is not important for the first two components. *The loadings matrices* **P** *and* **Q** in the relationships $\hat{\mathbf{X}}_0 = \mathbf{T}\mathbf{P}^{\mathrm{T}}$ and $\hat{\mathbf{Y}}_0 = \mathbf{T}\mathbf{Q}^{\mathrm{T}}$ (see Section 5.2), *can be considered as an indication of how strong the original variables are related to the PLS components* (De Jong, 1993). Note that **R** provides coefficients for forming the PLS components, while $\mathbf{P}^{\mathrm{T}}$ provides coefficients for linear combinations of the PLS components.

From Table 3.2 above, under the first component, Peroxide is strongly related to the first PLS component. Under the second component, Acidity and K232 is related to the second PLS component. From these deductions, one can conclude that X-variables Peroxide and K232 are important in the first component, while Acidity and K232 are important in the second component. Also, K270 can roughly be said to be important in both components, but not as strongly as Peroxide, K232 and Acidity. As for DK, one can conclude that it is least important in the first two components, prompting the question of its importance in the modelling of **Y**. A brief discussion on how to identify important X-variables for the modelling of the Y-variables is given in the next section. Furthermore, using the two components in Table 3.4, the estimated response values $\hat{\mathbf{Y}}_{\text{PLSR}}$ are shown below in Table 3.5.

Table 3.3 The loadings **Q** values of the sensory panel characteristics.

	Component 1	Component 2
Yellow	-31.380	-41.050
Green	31.040	51.580
Brown	15.440	-3.340
Glossy	-16.080	-6.090
Transp	-19.250	-10.100
Syrup	9.010	1.290

Table 3.4 The orthogonal latent variables **T**.

	Component 1	Component 2
G1	-0.039	0.709
G2	-0.074	-0.135
G3	-0.227	0.020
G4	0.036	0.452
G5	-0.158	0.258
I1	0.421	-0.050
I2	0.158	-0.070
I3	0.404	-0.132
I4	0.181	0.013
I5	0.475	-0.102
S1	-0.213	-0.206
S2	-0.394	-0.078
S3	-0.059	-0.127
S4	-0.174	-0.186
S5	-0.191	-0.237
S6	-0.144	-0.129

Table 3.5 The estimated sensory panel characteristics values $\hat{\mathbf{Y}}_{\text{PLSR}}$.

	Yellow	Green	Brown	Glossy	Transp	Syrup
G1	23.0	68.9	9.4	77.1	71.8	48.5
G2	58.7	24.3	11.6	82.8	81.0	47.1
G3	57.2	27.5	8.8	84.3	82.4	46.0
G4	31.2	57.9	11.4	77.5	72.9	48.9
G5	45.2	41.9	9.0	81.8	78.6	46.9
I1	39.7	44.0	19.0	74.4	70.6	51.7
I2	48.8	34.8	15.0	78.7	75.9	49.3
I3	43.6	39.2	19.0	75.1	71.8	51.4
I4	44.7	39.8	15.1	77.8	74.6	49.6
I5	40.2	43.0	20.0	73.8	70.1	52.1
S1	66.0	16.3	9.7	85.5	84.4	45.8
S2	66.4	17.3	6.5	87.6	86.6	44.3
S3	57.9	25.1	11.8	82.5	80.6	47.3
S4	64.0	18.5	10.3	84.8	83.4	46.2
S5	66.6	15.3	10.2	85.3	84.3	45.9
S6	60.7	22.4	10.5	83.9	82.3	46.5

3.9 Importance of PLSR coefficients

Generally, in regression, important predictor variables are selected for the modelling of the responses by using the regression coefficients. Specifically, important variables are identified by large coefficient values. Since the sign of each coefficient value gives an indication of the effect direction on the response variables, relevant predictor variables can be selected based on the magnitude of the absolute values of the coefficients. Once the relevant predictor variables have been selected, all the unimportant variables may be deleted. Afterwards, a final regression model is developed. To date, several variable selection methods have been developed for selecting the relevant predictor variables (Chong & Jun, 2005; Wold 1994; Wold *et al*., 1993; Wold *et al*., 2001). Among them is the Variable Importance in the Projection (VIP) method. In this method, the VIP value of each predictor is used as a summary of their importance. Chong & Jun (2005) proposed the VIP method to select the relevant predictor variables in PLSR. The VIP value is calculated for each predictor by

$$\text{VIP}_{\text{i}} = \sqrt{\left(\frac{\text{P}}{\sum_{\text{a}=1}^{\text{A}} \mathbf{b}_{\text{l}}^{\text{T}}\mathbf{b}_{\text{l}}\mathbf{t}_{\text{a}}^{\text{T}}\mathbf{t}_{\text{a}}}\right)\sum_{\text{a}=1}^{\text{A}} \mathbf{b}_{\text{i}}^{\text{T}}\mathbf{b}_{\text{i}}\mathbf{t}_{\text{a}}^{\text{T}}\mathbf{t}_{\text{a}}(r_{\text{ia}})^2} \qquad (3.18)$$

for $\text{i} = 1, 2, \cdots, \text{P}$, where P is the number of (predictor) X-variables, A the number of PLS components and r_{ia} the $(\text{i}, \text{a})^{\text{th}}$ element of the transformed X-weights matrix (**R**). The vectors $\mathbf{b}_{\text{i}}$ and $\mathbf{t}_{\text{a}}$ are the i^{th} and a^{th} columns of the coefficients ($\mathbf{B}_{\text{PLSR}}$) and X-scores (**T**) matrices

respectively. *A predictor variable having a* VIP *value greater than* 0.8 *is generally considered a relevant variable* (Nash & Lopez, 2010; Wold, 1994; Wold *et al*., 2001). Moreover, variable selection methods can be used to determine which PLSR coefficients are influential. Given that the matrix of PLSR coefficients $\mathbf{B}_{\text{PLSR}} = \mathbf{R}\mathbf{Q}^{\text{T}}$ gives the effect of the predictors on the responses variables, the magnitude of the absolute values of $\mathbf{B}_{\text{PLSR}}$ provides an indication of the influential coefficients. A small magnitude indicates that the coefficient is not quite influential (Wold, 1994).

3.9.1 *Example*

Consider the olive oil data discussed in Section 2.8. From the plot of VIP values shown in Figure 3.3, it appears that all the chemical quality measurements (X-variables) are important and should be included in the PLSR analysis.

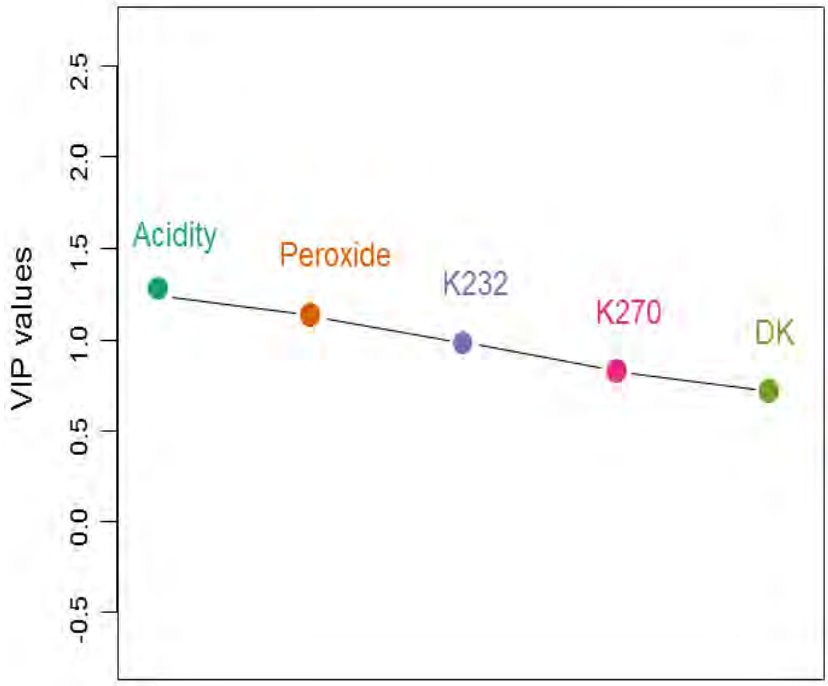

Figure 3. 3 Plot of the VIP values of the chemical quality measurements.

Furthermore, the absolute values of the predicted PLSR coefficients of this data are shown in Table 3.6. Observing the mean plot of the absolute values of the coefficients, shown in Figure 3.4, the following influential coefficients can be deduced. Choosing a benchmark of 0.7 as moderate, the respective magnitude is given in parenthesis: **b1**: Acidity (low), **b2**: Peroxide (higher), **b3**: K232 (higher), **b4**: K270 (high), and **b5**: DK (low).

Table 3.6 Absolute values of the estimated PLSR coefficients.

	Yellow	Green	Brown	Glossy	Transp	Syrup
Acidity	1.175	1.343	1.112	0.357	0.604	0.119
Peroxide	0.921	0.675	2.379	1.638	1.364	1.862
K232	1.169	0.977	1.984	1.646	1.448	1.741
K270	1.365	1.301	0.942	1.348	1.314	1.206
DK	1.042	1.071	0.046	0.748	0.818	0.515

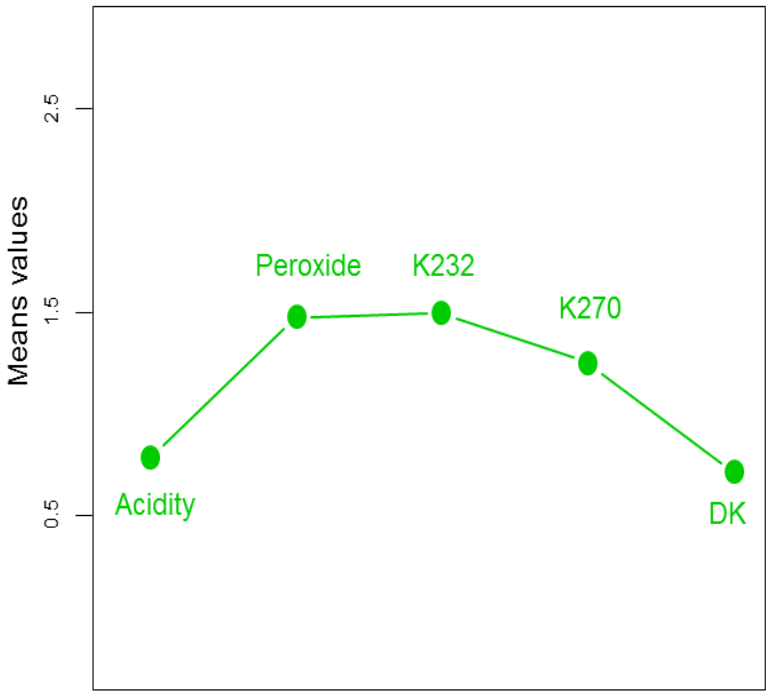

Figure 3.4 Mean plot of the absolute PLSR coefficients of the olive oil data.

3.10 Comparison of methods

In this section, PLSR is compared to other popular multivariate regression analysis methods, such as MMLR and PCR.

3.10.1 *MMLR*

In MMLR, the modelling of $\mathbf{Y}$ is done from $\mathbf{X}$. That is, $\mathbf{Y}$ is estimated as

$$\hat{\mathbf{Y}} = \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y} = \mathbf{X}\hat{\mathbf{B}}_{\mathrm{MMLR}},$$

where

$$\hat{\mathbf{B}}_{\mathrm{MMLR}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}$$

is the $(\mathrm{P} \times \mathrm{M})$ estimated MMLR coefficients matrix, for $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ non-singular.

3.10.2 *PCR*

PCR uses the few, A, principal components, extracted from $\mathbf{X}$, as predictors in the modelling of $\mathbf{Y}$. Specifically, it uses $\mathbf{Z} = \mathbf{X}\mathbf{V}_{[\mathrm{A}]}$ (2.9) as the predictors. As a result, $\mathbf{Y}$ is estimated as

$$\widehat{\mathbf{Y}} = \mathbf{Z}(\mathbf{Z}^{\mathrm{T}}\mathbf{Z})^{-1}\mathbf{Z}^{\mathrm{T}}\mathbf{Y} = \mathbf{Z}\mathbf{Q}_{+}^{\mathrm{T}} = \mathbf{X}\mathbf{V}_{[\mathrm{A}]}\mathbf{Q}_{+}^{\mathrm{T}} = \mathbf{X}\widehat{\mathbf{B}}_{\mathrm{PCR}},$$

where

$$\widehat{\mathbf{B}}_{\mathrm{PCR}} = \mathbf{V}_{[\mathrm{A}]}\mathbf{Q}_{+}^{\mathrm{T}}$$

is the (P × M) estimated PCR coefficients matrix and $\mathbf{Q}_{+} = \mathbf{Y}^{\mathrm{T}}\mathbf{Z}(\mathbf{Z}^{\mathrm{T}}\mathbf{Z})^{-1}$. The matrix $\mathbf{V}_{[\mathrm{A}]}$ is the first A columns of $\mathbf{V}$ (2.9). The main idea here is the dimension reduction of $\mathbf{X}$ for regression purposes. If $\mathrm{A} = \mathrm{P}$, there is no dimension reduction, thus, the resulting PCR performed is a MMLR with predictors $\mathbf{Z}$. There is another way to perform dimension reduction for regression purposes, namely, PLSR. This way (PLSR) is better suited because it takes into consideration both the X- and Y-variables when extracting the components, while PCR only considers the X-variables.

3.10.3 *PLSR*

In PLSR, the few (A) latent variables extracted from $\mathbf{X}$ are used as predictors in the modelling of $\mathbf{Y}$. That is, the X-scores $\mathbf{T}$ (see Section 3.5) are used as predictors. Thus, $\mathbf{Y}$ is estimated as

$$\widehat{\mathbf{Y}} = \mathbf{T}(\mathbf{T}^{\mathrm{T}}\mathbf{T})^{-1}\mathbf{T}^{\mathrm{T}}\mathbf{Y} = \mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{Y} \cong \mathbf{T}\mathbf{Q}^{\mathrm{T}} = \mathbf{X}\mathbf{R}\mathbf{Q}^{\mathrm{T}} = \mathbf{X}\widehat{\mathbf{B}}_{\mathrm{PLSR}},$$

where

$$\widehat{\mathbf{B}}_{\mathrm{PLSR}} = \mathbf{R}\mathbf{Q}^{\mathrm{T}}$$

is the (P × M) estimated PLSR coefficients matrix, for $(\mathbf{T}^{\mathrm{T}}\mathbf{T}) = \mathbf{I}_{\mathrm{A}}$ and $\mathbf{Q} = \mathbf{Y}^{\mathrm{T}}\mathbf{T}$. If A = P, there is no dimension reduction, hence, the resulting PLSR performed is a MMLR with predictors $\mathbf{T}$.

3.10.4 *Example*

Consider the olive oil data discussed in Section 2.8. The values of the MMLR-estimated sensory panel characteristics are shown in Table 3.8. These values are similar to the ones obtained from the PLSR analysis in Table 3.5 above. Also, a 2-component PCR analysis was performed and the resulting estimated sensory panel characteristics values are shown in Table 3.9. These values are similar to those obtained from the PLSR analysis in Table 3.5 above.

Table 3.7 The MSE values.

	PLSR	MMLR	PCR
Yellow	188.150	160.620	204.520
Green	290.650	265.400	312.220
Brown	9.060	5.310	8.640
Glossy	17.420	16.830	17.960
Transp	35.160	34.960	35.880
Syrup	3.630	3.310	3.630

PLSR(90.7), MMLR(81.1) and PCR(97.1)

Table 3.8 The estimated sensory panel characteristics values from the MMLR analysis.

	Yellow	Green	Brown	Glossy	Transp	Syrup
G1	26.7	65.0	8.1	76.3	71.4	48.4
G2	53.7	28.3	13.5	82.5	80.8	47.6
G3	49.1	32.9	11.6	83.7	82.0	47.0
G4	25.4	64.6	13.4	78.9	73.6	48.9
G5	48.1	39.5	7.9	82.4	79.0	46.8
I1	40.1	45.2	19.0	73.8	70.3	51.0
I2	51.2	32.9	14.2	78.1	75.5	49.0
I3	50.1	34.8	16.7	75.8	72.2	50.7
I4	50.7	32.3	12.8	77.8	74.7	50.0
I5	29.0	52.2	24.1	73.1	69.6	52.9
S1	61.8	22.1	11.4	85.4	84.2	45.3
S2	64.9	21.2	7.4	86.4	85.7	43.3
S3	62.8	19.7	10.0	82.9	80.9	47.5
S4	64.5	17.4	10.1	84.3	83.2	46.3
S5	72.7	9.9	7.8	86.5	85.0	45.8
S6	63.0	18.4	9.5	85.0	83.0	47.2

Table 3.9 The estimated sensory panel characteristics values from the PCR analysis.

	Yellow	Green	Brown	Glossy	Transp	Syrup
G1	27.4	63.9	8.9	77.9	72.7	48.5
G2	59.3	23.4	11.9	82.9	81.2	47.1
G3	59.3	24.8	9.0	84.7	82.9	45.9
G4	30.7	58.9	10.8	77.5	72.7	48.9
G5	44.6	42.9	8.6	81.8	78.4	46.9
I1	40.7	42.7	19.2	74.5	70.9	51.7
I2	49.6	33.7	15.2	78.8	76.1	49.3
I3	41.0	42.5	18.9	74.7	71.1	51.5
I4	43.9	40.8	15.0	77.7	74.4	49.7
I5	42.3	40.2	20.4	74.1	70.6	52.0
S1	65.9	16.3	10.0	85.4	84.4	45.8
S2	69.2	13.6	6.9	88.0	87.3	44.2
S3	55.9	27.6	11.8	82.2	80.1	47.3
S4	64.1	18.2	10.5	84.7	83.5	46.1
S5	62.5	20.4	10.0	84.7	83.3	46.1
S6	57.4	26.4	10.3	83.4	81.5	46.6

Comparing their respective Mean Squared Error (MSE) values for the sensory panel characteristics, shown above in Table 3.7, the MMLR analysis can be said to have the lowest MSE value of 81.1, followed by the PLSR analysis with 90.7, while the PCR analysis has the highest MSE value of 97.1. Overall, the PLSR analysis performed better than the PCR analysis. For the individual Y-variables, it can be seen that both analyses performed the same for Syrup,

but the PCR analysis performed better than the PLSR analysis for Brown. However, the fit is multivariate and should be compared on an overall level. For example, removing Brown from the data set will change the PCR and PLSR for all the other variables.

Furthermore, comparing the estimated PLSR, MMLR and PCR coefficient values in Tables 5.3 (see Chapter 5), 3.10 and 3.11 respectively, it can be seen that the values obtained under one technique are different from those obtained in the other two techniques.

Table 3.10 The estimated MMLR coefficient values ($\widehat{\mathbf{B}}_{\text{MMLR}}$).

	Yellow	Green	Brown	Glossy	Transp	Syrup
Acidity	-0.463	0.386	-0.223	-0.182	-0.251	0.394
Peroxide	0.111	-0.155	0.589	-0.285	-0.303	0.696
K232	-0.590	0.415	0.220	-0.443	-0.311	0.437
K270	-0.178	0.268	0.261	0.038	-0.008	-0.297
DK	0.228	-0.073	-0.283	-0.020	-0.013	-0.357

Table 3.11 The estimated PCR coefficient values ($\widehat{\mathbf{B}}_{\text{PCR}}$).

	Yellow	Green	Brown	Glossy	Transp	Syrup
Acidity	-0.220	0.268	-0.347	-0.021	-0.087	-0.090
Peroxide	-0.123	0.081	0.399	-0.252	-0.203	0.293
K232	-0.148	0.110	0.374	-0.260	-0.216	0.291
K270	-0.186	0.177	0.131	-0.189	-0.184	0.170
DK	-0.205	0.216	-0.030	-0.138	-0.158	0.086

From the MMLR analysis (Table 3.10), chemical DK can be seen to have a low effect on characteristics Green and Glossy, while chemical K270 has a low effect on characteristics Transp and Syrup. However, when compared to their counterparts in the PCR analysis (Table 3.11), different conclusions can be reached. For example, from Table 3.11, chemical Acidity has a low effect on characteristics Glossy and Transp, while chemical DK has a low effect on characteristic Syrup. This is not the case in the MMLR analysis (Table 3.10). Also, from the PCR analysis (Table 3.11), all five chemical measurements can be seen to have a negative effect on Yellow, Glossy and Transp, but a positive effect on Green and Syrup, with the exception of Acidity on Syrup. This is not the case in Table 3.10 (MMLR analysis), but it is in the PLSR analysis (Table 5.3, see Chapter 5). However, the values in Table 5.3 are different from those in Table 3.11. In the end, different multivariate regression analysis techniques can result in different estimations of the coefficients matrix **B** and the responses **Y**.

3.11 Summary

When there is the need to model a set of response variables from a (very) large set of predictor variables and there is no practical need to limit the number of predictor variables, PLSR can be a useful tool. PLSR does its modelling by using the extracted set of orthogonal components (or latent variables) as predictors. The extraction of the latent variable(s) can be done by using any of the PLS algorithms developed.

With numerous and correlated predictor variables, there is a chance of obtaining an over-fitted model. For this reason, a test for the predictive significance of each PLS component is necessary and this can be used to determine the appropriate number of components to use in the modelling of the responses. Measures such as the PRESS and RMSEP can be used to perform the test.

Compared with PCR and MMLR, PLSR maximizes the covariance between the predictor and response variables, while PCR maximizes the variance in the predictor variables. Conversely, MMLR maximizes the correlation between the predictor and response variables. A PLSR can be a MMLR (with predictors defined by the latent variables $\mathbf{T}$), if the final number of latent variables used in the analysis is the same as the number of predictor variables.

Next, in Chapter 4, the graphical representation of the relationships between the predictor and response variables is explored using covariance biplots.

CHAPTER 4

COVARIANCE BIPLOTS

4.1 Introduction

In a situation where the relationships between different sets of variables are of interest, various statistical techniques can be useful tools for analysis. Among them is the covariance matrix. Consider two centred matrices $\mathbf{X}_0$ (N × P) and $\mathbf{Y}_0$ (N × M). The covariance between $\mathbf{X}_0$ and $\mathbf{Y}_0$ is defined by

$$\underbrace{\text{cov}(\mathbf{X}_0, \mathbf{Y}_0)}_{(\text{P}\times\text{M})} = \frac{1}{\text{N}-1}(\mathbf{X}_0^{\text{T}}\mathbf{Y}_0). \tag{4.1}$$

However, when only one set of variables is under consideration, the variance-covariance matrix is defined by

$$\underbrace{\text{cov}(\mathbf{X}_0, \mathbf{X}_0)}_{(\text{P}\times\text{P})} = \frac{1}{\text{N}-1}(\mathbf{X}_0^{\text{T}}\mathbf{X}_0). \tag{4.2}$$

This is also written as $\text{cov}(\mathbf{X}_0, \mathbf{X}_0) = \text{var}(\mathbf{X}_0, \mathbf{X}_0) = \text{cov}(\mathbf{X}_0)$. Here, the variances of $\mathbf{X}_0$ are given in the diagonal of (4.2), while the covariances are shown off-diagonal. The relationships between different sets of variables can be explored using some form of graphical display such as biplots (see Chapter 2). Since biplots are useful graphical tools for exploring the relationships between variables, the biplot is employed in the form of the *covariance biplot*.

In this chapter, the general idea behind the covariance biplot is discussed. It further demonstrates, with graphical illustrations, how the covariance biplot can help to reveal variables and inter-variables relationships.

4.2 Covariance monoplot

In general, there are two kinds of features displayed in biplots. These features can be specified as two sets of variables, or as a set of variables and samples, as in the case of the PCA biplot (see Chapter 2). This does not mean that biplots cannot be constructed by using only one kind of feature, but depending on the data matrix and the choice of features to be analyzed, biplots can be constructed to display only one kind of feature. Gower *et al.* (2011) termed such biplots *monoplots*. In a monoplot, the kind of feature to be represented may be the samples only or one set of variables. Including an additional feature in the monoplot, say, another set of variables, would result in a biplot. As only variables are represented in the covariance and variance-covariance matrices (4.1) and (4.2), both monoplots and biplots can be used as graphical tools

to explore their relationships. More precisely, a monoplot would be suitable for representing a variance-covariance matrix (4.2) graphically, while a biplot would be more appropriate for a covariance matrix (4.1).

Consider the X-variables only. By the SVD, $\mathbf{X}_0 = \mathbf{U}\Delta\mathbf{V}^{\mathrm{T}}$, for $\mathbf{U}$ (N × P), Δ (P × P) and $\mathbf{V}$ (P × P). Defining the matrices $\mathbf{J}$ (P × P) $= \begin{bmatrix} \mathbf{I}_{\mathrm{r}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ and $\mathbf{J}_{\mathrm{r}}$ (P × r) $= \begin{bmatrix} \mathbf{I}_{\mathrm{r}} \\ \mathbf{0} \end{bmatrix}$, it follows that

$$\mathbf{X}_0 \cong \widehat{\mathbf{X}}_0 = \mathbf{U}\Delta\mathbf{J}\mathbf{V}^{\mathrm{T}} = \mathbf{U}\mathbf{J}\Delta\mathbf{V}^{\mathrm{T}} = \mathbf{U}\mathbf{J}\Delta\mathbf{J}\mathbf{V}^{\mathrm{T}}.$$

While $\widehat{\mathbf{X}}_0 \cong \mathbf{U}\mathbf{J}\Delta\mathbf{J}\mathbf{V}^{\mathrm{T}}$,

$$\begin{aligned} \widehat{\mathbf{X}}_0^{\mathrm{T}}\widehat{\mathbf{X}}_0 \;=\; (\mathbf{U}\mathbf{J}\Delta\mathbf{J}\mathbf{V}^{\mathrm{T}})^{\mathrm{T}}(\mathbf{U}\mathbf{J}\Delta\mathbf{J}\mathbf{V}^{\mathrm{T}}) \;&=\; \mathbf{V}\mathbf{J}^{\mathrm{T}}\Delta\mathbf{J}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\mathbf{U}\mathbf{J}\Delta\mathbf{J}\mathbf{V}^{\mathrm{T}} \\ &=\; \mathbf{V}\mathbf{J}^{\mathrm{T}}\Delta\mathbf{J}^{\mathrm{T}}\mathbf{J}\Delta\mathbf{J}\mathbf{V}^{\mathrm{T}} \\ &=\; \mathbf{V}\mathbf{J}^{\mathrm{T}}\Delta^2\mathbf{J}\mathbf{V}^{\mathrm{T}} \\ &=\; \mathbf{V}\mathbf{J}\Delta^2\mathbf{J}\mathbf{V}^{\mathrm{T}} \\ &=\; \mathbf{G}_{\mathrm{X}}\mathbf{H}_{\mathrm{X}}^{\mathrm{T}} \end{aligned}$$

where

$$\mathbf{G}_{\mathrm{X}}\ (\mathrm{P} \times \mathrm{r}) = \mathbf{V}\Delta\mathbf{J}_{\mathrm{r}} \text{ and } \mathbf{H}_{\mathrm{X}}\ (\mathrm{P} \times \mathrm{r}) = \mathbf{V}\Delta\mathbf{J}_{\mathrm{r}}, \qquad (4.3)$$

for $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}_{\mathrm{P}}$, $\mathbf{J}^{\mathrm{T}}\mathbf{J} = \mathbf{I}_{\mathrm{P}}$ and $\mathbf{J}^{\mathrm{T}} = \mathbf{J}$. Since $\mathbf{G}_{\mathrm{X}} = \mathbf{H}_{\mathrm{X}}$ in (4.3), the rows of either $\mathbf{G}_{\mathrm{X}}$ or $\mathbf{H}_{\mathrm{X}}$ will be used in the monoplot. Let $\mathbf{S}_{\mathrm{XX}} = \mathrm{cov}(\mathbf{X}_0, \mathbf{X}_0)$. From (4.2), $\mathbf{S}_{\mathrm{XX}} = \frac{1}{\mathrm{N}-1}(\mathbf{X}_0^{\mathrm{T}}\mathbf{X}_0)$, making $\mathbf{X}_0^{\mathrm{T}}\mathbf{X}_0 = (\mathrm{N}-1)\mathbf{S}_{\mathrm{XX}}$. Thus, $\widehat{\mathbf{X}}_0^{\mathrm{T}}\widehat{\mathbf{X}}_0 = \mathbf{V}\mathbf{J}\Delta^2\mathbf{J}\mathbf{V}^{\mathrm{T}}$ approximates the variance-covariance matrix $(\mathrm{N}-1)\mathbf{S}_{\mathrm{XX}}$. That is,

$$\widehat{\mathbf{X}}_0^{\mathrm{T}}\widehat{\mathbf{X}}_0 = (\mathrm{N}-1)\widehat{\mathbf{S}}_{\mathrm{XX}}.$$

Moreover, given that only one set of variables, in this case $\mathbf{X}$, is under consideration, and as the focus is on revealing the relationships within these variables, only one set of axes is needed. From (4.3), the directions of these axes are calculated by the P rows of either $\mathbf{G}_{\mathrm{X}}$ or $\mathbf{H}_{\mathrm{X}}$.

4.3 Covariance biplot

Consider both $\mathbf{X}_0$ and $\mathbf{Y}_0$. The (P × M) covariance matrix between $\mathbf{X}_0$ and $\mathbf{Y}_0$ is defined in (4.1). Let $\mathbf{S}_{\mathrm{XY}} = \mathrm{cov}(\mathbf{X}_0, \mathbf{Y}_0)$. By the SVD, $\mathbf{S}_{\mathrm{XY}} = \mathbf{D}\Lambda\mathbf{F}^{\mathrm{T}}$, for $\mathbf{D}$ (P × M), Λ (M × M) and $\mathbf{F}$ (M × M). The matrix $\mathbf{S}_{\mathrm{XY}} = \mathbf{D}\Lambda\mathbf{F}^{\mathrm{T}}$ can be written as

$$\mathbf{S}_{\mathrm{XY}} \cong \widehat{\mathbf{S}}_{\mathrm{XY}} = \mathbf{D}\Lambda\mathbf{J}\mathbf{F}^{\mathrm{T}} = \mathbf{D}\mathbf{J}\Lambda\mathbf{F}^{\mathrm{T}} = \mathbf{D}\mathbf{J}\Lambda\mathbf{J}\mathbf{F}^{\mathrm{T}} = \mathbf{G}\mathbf{H}^{\mathrm{T}}$$

where

$$\mathbf{G} = \mathbf{D}\Lambda^{\beta}\mathbf{J}_{\mathrm{r}} \text{ and } \mathbf{H} = \mathbf{F}\Lambda^{1-\beta}\mathbf{J}_{\mathrm{r}}, \text{ for any value of } \beta \in (0,1). \qquad (4.4)$$

In (4.4), the matrix $\mathbf{J}$ has dimension (M × M), while the matrix $\mathbf{J}_{\mathrm{r}}$ has dimension (M × r). The matrix $\mathbf{G}$ (P × r) contains the information about the X-variables, while $\mathbf{H}$ (M × r) contains the information about the Y-variables. Since $\widehat{\mathbf{S}}_{\mathrm{XY}} = \mathbf{G}\mathbf{H}^{\mathrm{T}}$, the innerproduct between the rows of the

matrix $\mathbf{G}$ and the rows of the matrix $\mathbf{H}$ approximates the covariances between the X-variables and the Y-variables. Here, the rows of $\mathbf{G}$ associates with the X-variables, while the rows of $\mathbf{H}$ associates with the Y-variables. Focusing on revealing the relationships between two sets of variables, $\mathbf{X}$ and $\mathbf{Y}$, only axes will be present in the resulting biplot. However, two sets of axes are needed, a set for the X-variables and a set for the Y-variables. From $\hat{\mathbf{S}}_{XY} = \mathbf{G}\mathbf{H}^{T}$, the directions of the axes representing the X-variables are calculated using the P rows of $\mathbf{G}$, while M rows of $\mathbf{H}$ are used to calculate the directions of the axes representing the Y-variables. This biplot, called the *covariance biplot*, reveals the relationships between the two sets of variables as well as within each set.

From (4.4), when $\beta = 1$,

$$\mathbf{G} = \mathbf{D}\boldsymbol{\Lambda}^{1}\mathbf{J}_{r} = \mathbf{D}\boldsymbol{\Lambda}\mathbf{J}_{r} \text{ and } \mathbf{H} = \mathbf{F}\boldsymbol{\Lambda}^{1-1}\mathbf{J}_{r} = \mathbf{F}\mathbf{J}_{r}.$$

Also,

$$\mathbf{H}^{T}\mathbf{H} = (\mathbf{F}\mathbf{J}_{r})^{T}(\mathbf{F}\mathbf{J}_{r}) = \mathbf{J}_{r}^{T}\mathbf{F}^{T}\mathbf{F}\mathbf{J}_{r} = \mathbf{J}_{r}^{T}\mathbf{J}_{r} = \mathbf{I}_{r}$$

and

$$\mathbf{G}^{T}\mathbf{G} = (\mathbf{D}\boldsymbol{\Lambda}\mathbf{J}_{r})^{T}(\mathbf{D}\boldsymbol{\Lambda}\mathbf{J}_{r}) = \mathbf{J}_{r}^{T}\boldsymbol{\Lambda}\mathbf{D}^{T}\mathbf{D}\boldsymbol{\Lambda}\mathbf{J}_{r} = \mathbf{J}_{r}^{T}\boldsymbol{\Lambda}^{2}\mathbf{J}_{r} \neq \mathbf{I}_{r}$$

where $\mathbf{F}^{T}\mathbf{F} = \mathbf{I}_{M}$, $\mathbf{J}_{r}^{T}\mathbf{J}_{r} = \mathbf{I}_{r}$ and $\mathbf{D}^{T}\mathbf{D} = \mathbf{I}_{M}$. For this choice of β,

$$\mathbf{S}_{XY}\mathbf{S}_{XY}^{T} \cong \mathbf{G}\mathbf{H}^{T}\mathbf{H}\mathbf{G}^{T} = \mathbf{G}\mathbf{G}^{T}$$

but

$$\mathbf{S}_{XY}^{T}\mathbf{S}_{XY} \cong \mathbf{H}\mathbf{G}^{T}\mathbf{G}\mathbf{H}^{T} \neq \mathbf{H}\mathbf{H}^{T}.$$

Therefore, from $\mathbf{S}_{XY}\mathbf{S}_{XY}^{T} \cong \mathbf{G}\mathbf{G}^{T}$, the row markers $\mathbf{G}$ approximate the covariance between the rows of $\mathbf{S}_{XY}$. Since the rows of $\mathbf{S}_{XY}$ are associated with the X-variables, the rows of $\mathbf{G}$ approximate the covariance between these X-variables. Conversely, when $\beta = 0$,

$$\mathbf{G} = \mathbf{D}\boldsymbol{\Lambda}^{0}\mathbf{J}_{r} = \mathbf{D}\mathbf{J}_{r} \text{ and } \mathbf{H} = \mathbf{F}\boldsymbol{\Lambda}^{1-0}\mathbf{J}_{r} = \mathbf{F}\boldsymbol{\Lambda}\mathbf{J}_{r}.$$

Now,

$$\mathbf{G}^{T}\mathbf{G} = (\mathbf{D}\mathbf{J}_{r})^{T}(\mathbf{D}\mathbf{J}_{r}) = \mathbf{J}_{r}^{T}\mathbf{D}^{T}\mathbf{D}\mathbf{J}_{r} = \mathbf{J}_{r}^{T}\mathbf{J}_{r} = \mathbf{I}_{r}$$

for $\mathbf{D}^{T}\mathbf{D} = \mathbf{I}_{M}$ and $\mathbf{J}_{r}^{T}\mathbf{J}_{r} = \mathbf{I}_{r}$, but

$$\mathbf{H}^{T}\mathbf{H} = (\mathbf{F}\boldsymbol{\Lambda}\mathbf{J}_{r})^{T}(\mathbf{F}\boldsymbol{\Lambda}\mathbf{J}_{r}) = \mathbf{J}_{r}^{T}\boldsymbol{\Lambda}\mathbf{F}^{T}\mathbf{F}\boldsymbol{\Lambda}\mathbf{J}_{r} = \mathbf{J}_{r}^{T}\boldsymbol{\Lambda}^{2}\mathbf{J}_{r} \neq \mathbf{I}_{r}$$

where $\mathbf{F}^{T}\mathbf{F} = \mathbf{I}_{M}$. Thus,

$$\mathbf{S}_{XY}^{T}\mathbf{S}_{XY} \cong \mathbf{H}\mathbf{G}^{T}\mathbf{G}\mathbf{H}^{T} = \mathbf{H}\mathbf{H}^{T}$$

and

$$\mathbf{S}_{XY}\mathbf{S}_{XY}^{T} \cong \mathbf{G}\mathbf{H}^{T}\mathbf{H}\mathbf{G}^{T} \neq \mathbf{G}\mathbf{G}^{T}.$$

From $\mathbf{S}_{XY}^{T}\mathbf{S}_{XY} \cong \mathbf{H}\mathbf{H}^{T}$, the column markers $\mathbf{H}$ approximate the covariance between the columns of $\mathbf{S}_{XY}$. In other words, with the columns of $\mathbf{S}_{XY}$ associated with the Y-variables, the rows of $\mathbf{H}$ approximate the covariance between these Y-variables.

Moreover, any β value between 0 and 1 will neither optimally approximate the covariance between the X-variables nor the covariance between the Y-variables, but rather, it will give an indication of both. *Since choosing β closer to 1 better approximates the covariance between the X-variables and choosing β closer to 0 better approximates the covariance between the Y-variables, the symmetric choice of $\beta = \frac{1}{2} = 0.5$ will be used in the biplot*. With this choice of β, covariance between the X- and Y-variables is both equally approximated, although not optimal for either. That is, for $\mathbf{G} = \mathbf{D}\boldsymbol{\Lambda}^{0.5}\mathbf{J}_r$ and $\mathbf{H} = \mathbf{F}\boldsymbol{\Lambda}^{1-0.5}\mathbf{J}_r = \mathbf{F}\boldsymbol{\Lambda}^{0.5}\mathbf{J}_r$,

$$\mathbf{H}^T\mathbf{H} = (\mathbf{F}\boldsymbol{\Lambda}^{0.5}\mathbf{J}_r)^T(\mathbf{F}\boldsymbol{\Lambda}^{0.5}\mathbf{J}_r) = \mathbf{J}_r^T\boldsymbol{\Lambda}^{0.5}\mathbf{F}^T\mathbf{F}\boldsymbol{\Lambda}^{0.5}\mathbf{J}_r = \mathbf{J}_r^T\boldsymbol{\Lambda}\mathbf{J}_r \neq \mathbf{I}_r,$$
$$\mathbf{G}^T\mathbf{G} = (\mathbf{D}\boldsymbol{\Lambda}^{0.5}\mathbf{J}_r)^T(\mathbf{D}\boldsymbol{\Lambda}^{0.5}\mathbf{J}_r) = \mathbf{J}_r^T\boldsymbol{\Lambda}^{0.5}\mathbf{D}^T\mathbf{D}\boldsymbol{\Lambda}^{0.5}\mathbf{J}_r = \mathbf{J}_r^T\boldsymbol{\Lambda}\mathbf{J}_r \neq \mathbf{I}_r,$$
$$\mathbf{S}_{XY}^T\mathbf{S}_{XY} \cong \mathbf{H}\mathbf{G}^T\mathbf{G}\mathbf{H}^T = \mathbf{H}\mathbf{J}_r^T\boldsymbol{\Lambda}\mathbf{J}_r\mathbf{H}^T \text{ and } \mathbf{S}_{XY}\mathbf{S}_{XY}^T \cong \mathbf{G}\mathbf{H}^T\mathbf{H}\mathbf{G}^T = \mathbf{G}\mathbf{J}_r^T\boldsymbol{\Lambda}\mathbf{J}_r\mathbf{G}^T,$$

where $\mathbf{F}^T\mathbf{F} = \mathbf{I}_M$ and $\mathbf{D}^T\mathbf{D} = \mathbf{I}_M$. In this situation, the rows of $\mathbf{H}\mathbf{J}_r^T\boldsymbol{\Lambda}^{0.5}$ approximate (non-optimally) the covariance between the Y-variables, while the rows of $\mathbf{G}\mathbf{J}_r^T\boldsymbol{\Lambda}^{0.5}$ approximate (non-optimally) the covariance between the X-variables. In line with (4.1), the approximated covariance can be written as

$$\begin{aligned} \widehat{\text{cov}(\mathbf{X}_0, \mathbf{Y}_0)} &= \mathbf{G}\mathbf{J}_r^T\boldsymbol{\Lambda}^{0.5}(\mathbf{H}\mathbf{J}_r^T\boldsymbol{\Lambda}^{0.5})^T \\ &= \mathbf{G}\mathbf{J}_r^T\boldsymbol{\Lambda}^{0.5}\boldsymbol{\Lambda}^{0.5}\mathbf{J}_r\mathbf{H}^T \\ &= \mathbf{G}\mathbf{J}_r^T\boldsymbol{\Lambda}\mathbf{J}_r\mathbf{H}^T. \end{aligned} \quad (4.5)$$

Thus, in (4.4), $\beta = 0$ only caters for the covariance between the Y-variables optimally, while $\beta = 1$ only caters for the X-variables optimally. On the other hand, $\beta = 0.5$ caters for both X- and Y-variables equally, although not as optimally as when only one set is been catered for. *Seeing as only variables are being represented in the covariance monoplot/biplot and there are no samples to (orthogonally) project onto the axes representing these variables, calibration markers are not necessary on these axes*.

4.4 Example

The following example is an illustration of a covariance biplot, using the olive oil data discussed in Section 2.8. The five chemical quality measurements (Acidity, Peroxide, K232, K270 and DK) and the six sensory panel characteristics (Yellow, Green, Brown, Glossy, Transp and Syrup) are assigned as the X- and Y-variables respectively. The covariance biplot for the data is shown below in Figure 4.1, with $\beta = 0.5$. Since this choice of β does not optimally approximate the covariance between the X- and Y-variables, using (4.5), the approximated correlation values are shown in Table 4.4. The $\mathbf{G}$ (5×2) and $\mathbf{H}$ (6×2) matrices are shown in Tables 4.1 and 4.2 respectively.

Table 4.1 The matrix **G**.

	Component 1	Component 2
Acidity	0.358	0.686
Peroxide	0.886	-0.387
K232	0.932	-0.196
K270	0.834	0.179
DK	0.510	0.256

Table 4.2 The matrix **H**.

	Component 1	Component 2
Yellow	-0.655	-0.363
Green	0.600	0.444
Brown	0.662	-0.611
Glossy	-0.735	0.022
Transp	-0.688	-0.096
Syrup	0.705	-0.211

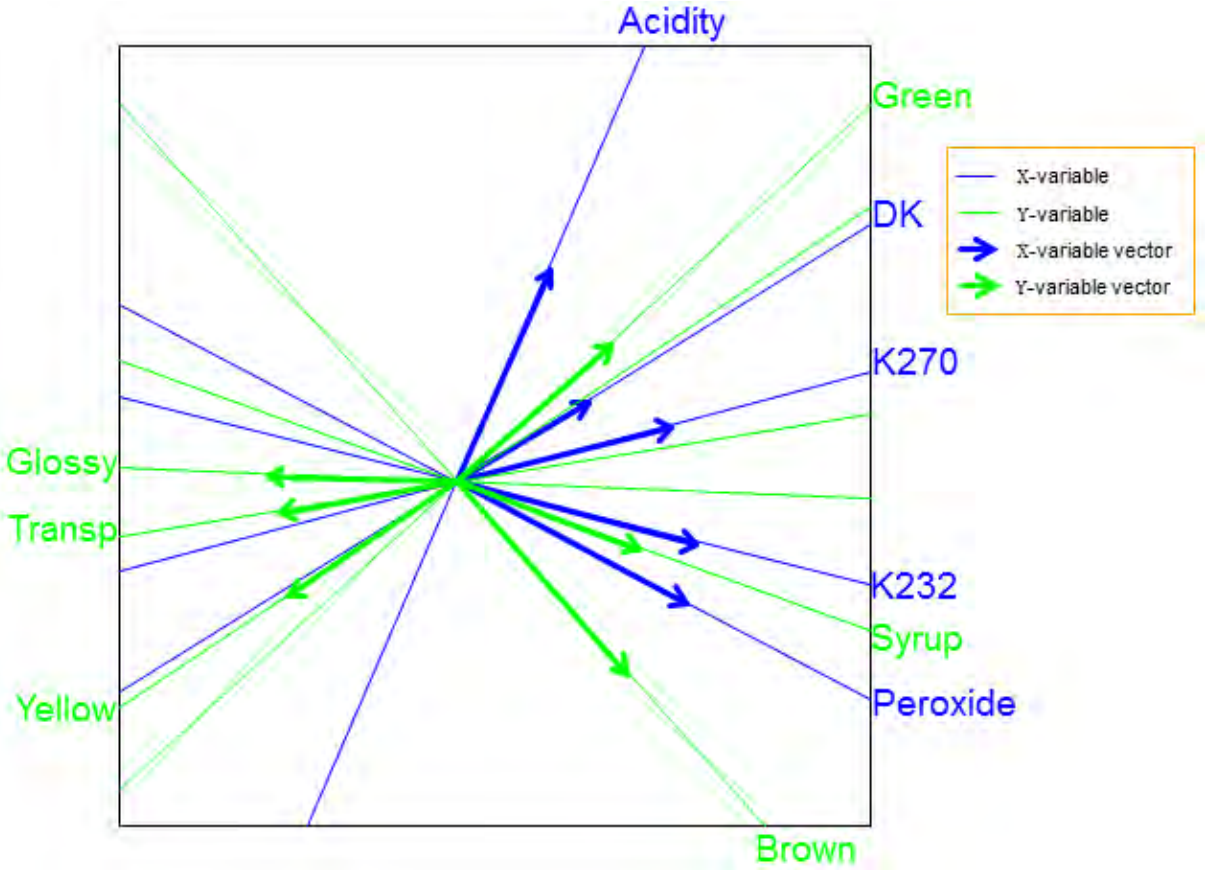

Figure 4.1 The covariance biplot of the olive oil data, $\beta = 0.5$.

Another display of Figure 4.1, with calibration markers on the axes is shown below in Figure 4.2. Seeing as only variables are being represented in the covariance biplot and there are no samples points to project onto the axes representing these variables, calibration markers are not necessary on these axes. For this reason, the form of the display in Figure 4.1 above (no calibration markers) will be used as the covariance biplot display throughout this dissertation. In Figure 4.1, the representation of the variance of each variable, represented by the thicker arrow (vector) on each axis, is shown. From this biplot, the standard deviation of chemical DK is smaller compared to the other chemicals. This is apparent from the length of the thicker arrow

(vector) on the DK axis. Likewise, characteristic Brown can be seen to have a larger standard deviations compared to the other characteristics.

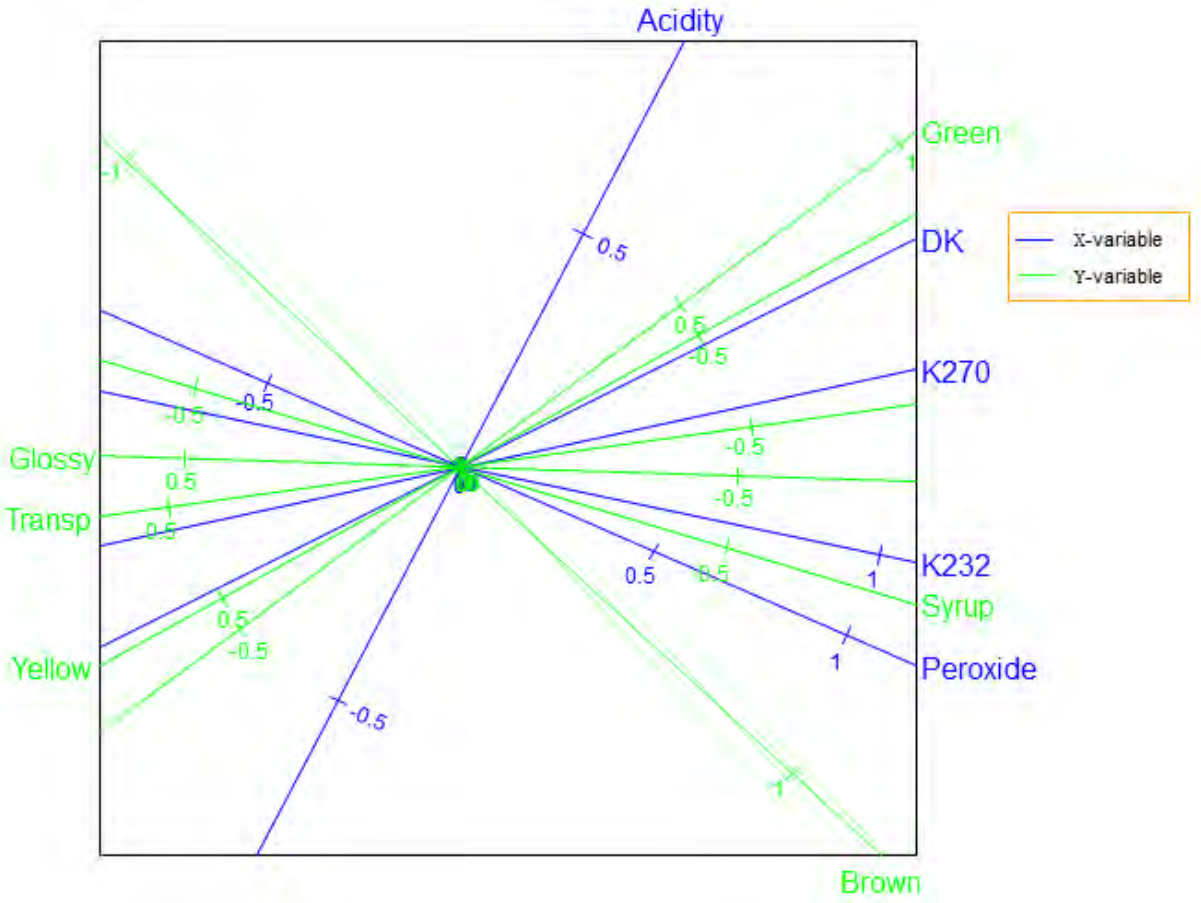

Figure 4.2 The covariance biplot of the olive oil data, $\beta = 0.5$, with calibration markers.

In addition, looking at the angles between the blue vectors in Figure 4.1, all the chemicals measurements can be said to be positive related with each other. Similarly, characteristics Glossy and Transp can be said to be positively related, while characteristics Green and Yellow are said to be negatively related. Characteristics Brown and Syrup can be said to be (some-what) positively related. The actual correlation values of this data are shown in Figure 2.4. However, an extract from Figure 2.4, showing only the correlation values between the chemical measurements and characteristics, is shown below in Table 4.3. Comparing the approximated correlation values in Table 4.4 with the actual correlation values in Table 4.3, similar conclusions can be reached, in terms of the effect directions of each chemical measurement on the panel characterisics, but a slightly different conclusions in terms of their respective values. For example, from Table 4.4 (approximated values), all five chemical measurements have a positive effect on characteristics Green and Syrup, but a negative effect on Yellow, Glossy and Transp. These can also be deduced from Table 4.3 (actual values). However, the lowest correlation value of 0.084 can be observed under characteristic Brown in Table 4.3 (actual

values), while in Table 4.4 (approximated values), the lowest value of 0.108 is observed under characteristic Syrup.

Moreover, various inter-variable relationships can be observed in Figure 4.1 above, such as a relation between the chemical K270 and characteristics Glossy and Transp. Observing their (actual) correlation values (-0.529 and -0.522), shown in Table 4.3 or Figure 2.4, indicates a fair relationship between them. Also, a relation between characteristic Syrup and chemicals K232 and Peroxide (0.686 and 0.759); and between chemical DK and characteristics Green and Yellow (0.361 and -0.334) can be noted. The latter relation is not quite as strong as the former. Acidity and Brown can be seen to have no clear relation with the others.

Table 4.3 The actual correlation values between the chemicals and sensory characteristics.

	Yellow	Green	Brown	Glossy	Transp	Syrup
Acidity	-0.485	0.513	-0.197	-0.235	-0.309	0.142
Peroxide	-0.414	0.339	0.777	-0.669	-0.597	0.759
K232	-0.547	0.472	0.745	-0.695	-0.615	0.686
K270	-0.687	0.640	0.549	-0.529	-0.522	0.478
DK	-0.334	0.361	0.084	-0.480	-0.462	0.332

Table 4.4 The approximated correlation values between the chemicals and sensory characteristics.

	Yellow	Green	Brown	Glossy	Transp	Syrup
Acidity	-0.483	0.519	-0.182	-0.248	-0.312	0.108
Peroxide	-0.440	0.360	0.823	-0.660	-0.573	0.707
K232	-0.540	0.472	0.737	-0.689	-0.623	0.699
K270	-0.611	0.580	0.442	-0.609	-0.591	0.550
DK	-0.427	0.419	0.181	-0.369	-0.375	0.305

Furthermore, to illustrate how a covariance monoplot can help to reveal relationships within one set of variables, consider the monoplot of the sensory panel characteristics shown below in Figure 4.3. From this monoplot, several relationships can be deduced, such as the relation within characteristics Transp, Glossy and Syrup. Also, a relation within characteristics Yellow and Green can be noted. Brown can be seen to have no clear relation with the others.

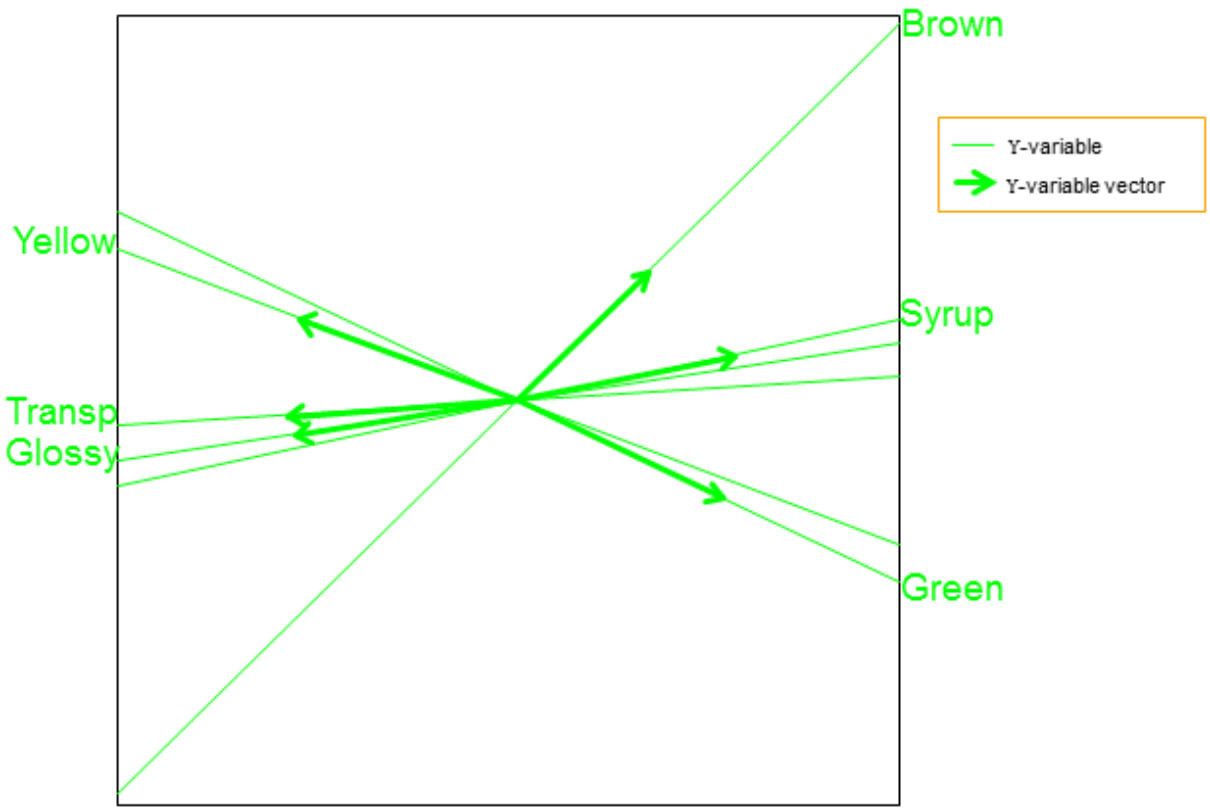

Figure 4.3 The covariance monoplot of the sensory panel characteristics.

Although not recommended, the biplot displays obtained when $\beta = 1$ and $\beta = 0$ are shown in Figures 4.4 and 4.5 respectively. Comparing these displays to that in Figure 4.1, different deductions can be made. On one hand, from Figure 4.4 ($\beta = 1$), a relation between Green, Yellow and Acidity; between DK, K270 and Transp; as well as between Peroxide, K232 and Glossy can be observed. Here, Brown and Syrup can be seen to have no clear relation with the others. On the other hand, from Figure 4.5 ($\beta = 0$), a relation between K270, Green and Yellow; between Glossy, Transp and Syrup; as well as between K232 and Brown can be deduced. Acidity, DK and Peroxide can be seen to have no clear relation with the others. Nevertheless, since there are two sets of variables, $\mathbf{X}$ and $\mathbf{Y}$, in the covariance matrix $\mathbf{S}_{XY}$ (or in Tables 4.4 and 4.5), and $\beta = 1$ as well as $\beta = 0$ only caters for one set of variables, while the other is ignored, the biplot displays obtained using these two choices of β (Figures 4.4 and 4.5) are not recommended. Hence, it is more appropriate to use $\beta = 0.5$, where both sets of variables are catered for equally.

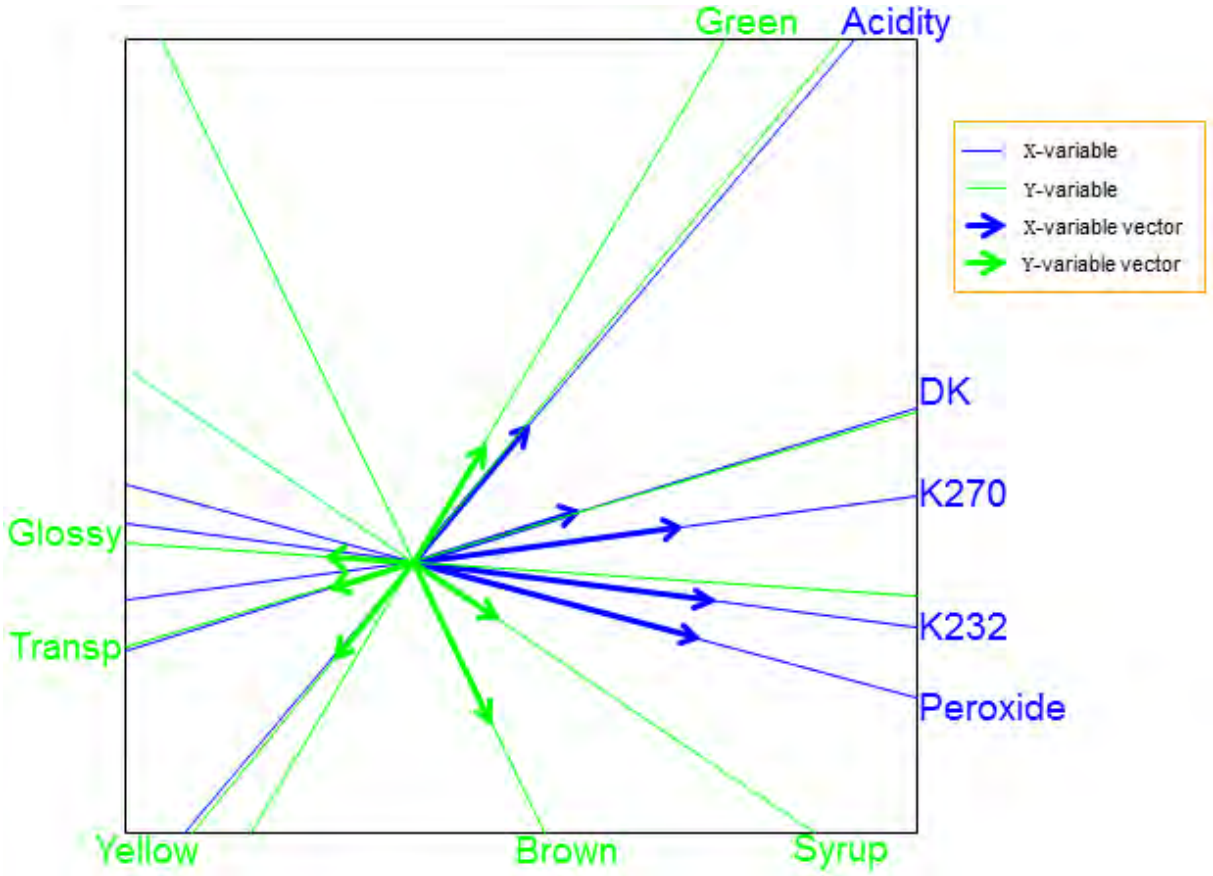

Figure 4.4 The covariance biplot of the olive oil data, $\beta = 1$.

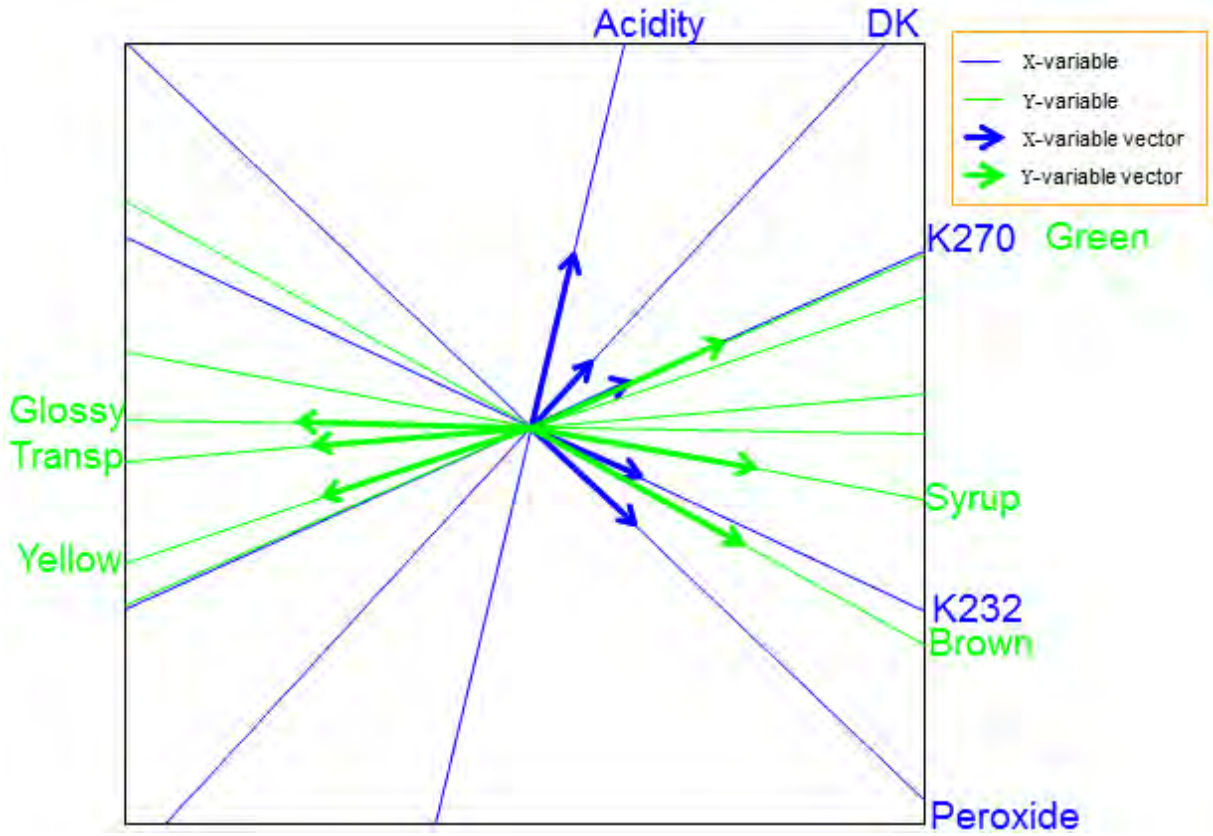

Figure 4.5 The covariance biplot of the olive oil data, $\beta = 0$.

4.5 Summary

One of the means used in analysing the relationships between different sets of variables is the covariance matrix. It describes how much two sets of variables change together. The covariance matrix of two sets of variables can be visualized graphically using the biplot. The resulting biplot is termed the covariance biplot.

If only one set of variables is considered in the covariance analysis, the resulting graphical representation is a covariance monoplot.

Advantages of a covariance biplot include the revelation of the relationships between two sets of variables as well as within each set.

In general, the covariance biplot graphically represents the covariance between a set of predictor variables and a set of response variables. Since PLS finds a set of latent variables (**T**) that maximizes the covariance between these two sets of variables, next, in Chapter 5, the biplot is employed in the form of the PLS biplot to help explore the relationships between these two sets of variables.

CHAPTER 5

PARTIAL LEAST SQUARES BIPLOTS

5.1 Introduction

Employing biplots in regression analysis yields many advantages, including demonstrating the association between samples and/or variables graphically. An example of such employment is the *PLS biplot*, which is a graphical display of a PLSR of a data set. As a new addition to the biplot family, the PLS biplot has the abovementioned advantage. It further provides a single graphical representation of the samples, together with the predictor and response variables, as well as their inter-relationships in terms of the matrix of regression coefficients. In this chapter, the fundamental idea behind the PLS biplot is discussed, before it is constructed. In addition, two ways of representing the matrix of PLSR coefficients in the PLS biplot are introduced. The first way involves using calibrated biplot axes, while the second way utilizes the area biplots introduced by Gower *et al*. (2010). Furthermore, the PLS biplot is compared with the PCA and covariance biplots.

5.2 PLS biplot

PLS can be viewed as an approximation technique, since it approximates a data matrix using only a few (A) components. For example, PLS approximates $\mathbf{X}$ (N × P) using a set of latent variables $\mathbf{T}$ (N × A) and a set of loadings $\mathbf{P}$ (P × A) such that

$$\begin{aligned} \widehat{\mathbf{X}}_0 &= \mathbf{T}(\mathbf{T}^{\mathrm{T}}\mathbf{T})^{-1}\mathbf{T}^{\mathrm{T}}\mathbf{X}_0 \\ &= \mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{X}_0 \\ &\cong \mathbf{T}\mathbf{P}^{\mathrm{T}} \end{aligned} \qquad (5.1)$$

where $(\mathbf{T}^{\mathrm{T}}\mathbf{T}) = \mathbf{I}_{\mathrm{A}}$. Matrices $\mathbf{T}$ and $\mathbf{P}$ are obtained using any of the algorithms in Section 3.5 (see Algorithms 3.1 to 3.4). Similarly, $\mathbf{Y}$ (N × M) is approximated using the same set of latent variables $\mathbf{T}$ and a set of loadings $\mathbf{Q}$ (M × A) such that

$$\begin{aligned} \widehat{\mathbf{Y}}_0 &= \mathbf{T}(\mathbf{T}^{\mathrm{T}}\mathbf{T})^{-1}\mathbf{T}^{\mathrm{T}}\mathbf{Y}_0 \\ &= \mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{Y}_0 \\ &\cong \mathbf{T}\mathbf{Q}^{\mathrm{T}} \end{aligned} \qquad (5.2)$$

where $\mathbf{Q}$ is defined in Section 3.5. In other words, PLS approximates an $\mathrm{N} \times (\mathrm{P} + \mathrm{M})$ data matrix $\mathbf{D} = [\mathbf{X} \quad \mathbf{Y}]$ as

$$\widehat{\mathbf{D}}_0 = \left[\widehat{\mathbf{X}}_0 \quad \widehat{\mathbf{Y}}_0\right] = [\mathbf{T}\mathbf{P}^{\mathrm{T}} \quad \mathbf{T}\mathbf{Q}^{\mathrm{T}}] = \mathbf{T}[\mathbf{P} \quad \mathbf{Q}]^{\mathrm{T}}.$$

Here, the matrix $\mathbf{T}$ contains information about the samples, while the loadings matrices $\mathbf{P}$ and $\mathbf{Q}$ contain information about the X- and Y-variables respectively. Corrected for their respective means,

$$\hat{\mathbf{X}} \cong \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{1}\bar{\mathbf{x}} \quad \text{and} \quad \hat{\mathbf{Y}} \cong \mathbf{T}\mathbf{Q}^{\mathrm{T}} + \mathbf{1}\bar{\mathbf{y}}$$

for $\mathbf{1}$ (N × 1) a vector of ones, $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ the column means of $\mathbf{X}$ and $\mathbf{Y}$ respectively. Thus, the PLS approximation of $\mathbf{D} = [\mathbf{X} \quad \mathbf{Y}]$ can be written as

$$\hat{\mathbf{D}} = \left[\hat{\mathbf{X}} \quad \hat{\mathbf{Y}}\right] \cong [\mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{1}\bar{\mathbf{x}} \quad \mathbf{T}\mathbf{Q}^{\mathrm{T}} + \mathbf{1}\bar{\mathbf{y}}]. \tag{5.3}$$

Analogous to the definition of a biplot in (2.2), for the PLS biplot, sample points are represented by the rows of the matrix $\mathbf{T}$, while the directions of the biplot axes are calculated by the rows of matrices $\mathbf{P}$ and $\mathbf{Q}$ for the predictor variables and response variables respectively. Although this is true for any number of components A, it is only practical to plot in two or three dimensions. For two-dimensional biplots, $A = 2$ components will be extracted and plotted in a new A-dimensional orthogonal Cartesian axes system referred to as the scaffolding axes. These scaffolding axes are not shown, but since $\mathbf{T} = \mathbf{X}_0\mathbf{R}$, they are defined by the first A columns of $\mathbf{R}$, the PLS (transformed) X-weights matrix.

5.2.1 *Interpolation of samples*

In the PLS biplot, interpolation is achieved by orthogonally projecting each biplot point onto the biplot plane. Here, the plane is the vector space generated by the columns of $\mathbf{R}$. Consider $\mathbf{x}_0$ and $\mathbf{y}_0$ as a pair of centred predictor and response samples. Since sample points are represented by the rows of the matrix $\mathbf{T} = \mathbf{X}_0\mathbf{R}$, both $\mathbf{x}_0$ and $\mathbf{y}_0$ are interpolated into the biplot plane using the equation

$$\mathbf{t}^{\mathrm{T}} = \mathbf{x}_0{}^{\mathrm{T}}\mathbf{R}. \tag{5.4}$$

5.2.2 *Prediction of samples*

Given that the biplot plane is a subspace of the full-dimensional space, any point in terms of the basis for the biplot plane is also a point in terms of the basis for the full space. Thus, a point $\mathbf{t}$ in terms of the basis for the biplot plane is also a point $\mathbf{x}_0$ in terms of the P-dimensional space of $\mathbf{X}_0$. Thus, such a point will project onto itself, yielding the prediction of $\mathbf{x}_0$ as

$$\hat{\mathbf{x}}_0^{\mathrm{T}} = \mathbf{t}^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}. \tag{5.5}$$

Likewise, a point $\mathbf{t}$ in terms of the basis for the biplot plane is also a point $\mathbf{y}_0$ in terms of the M-dimensional space of $\mathbf{Y}_0$. Hence, the prediction of sample $\mathbf{y}_0$ is achieved by

$$\hat{\mathbf{y}}_0^{\mathrm{T}} = \mathbf{t}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}. \tag{5.6}$$

5.2.3 *Prediction of axes*

To construct a prediction biplot axis, consider the k^{th} predictor and response variables respectively. From (5.1), $\mathbf{x}_0^T = \mathbf{t}^T\mathbf{P}^T$ for any point $\mathbf{t}$ in the biplot plane. For this reason, the k^{th} centred predictor variable value is given by $\mathbf{x}_0^T\mathbf{e}_k = \mathbf{t}^T\mathbf{P}^T\mathbf{e}_k$, where $\mathbf{e}_k$ is the unit vector with zeros except for a one in the k^{th} position. If this value is μ_{0X}, then

$$\mu_{0X} = \mathbf{t}^T\mathbf{P}^T\mathbf{e}_k. \tag{5.7}$$

This defines a line in the two-dimensional biplot plane and for different values, $\mu_{0X} \in (-\infty, \infty)$, parallel lines are obtained, as shown below in Figure 5.1.

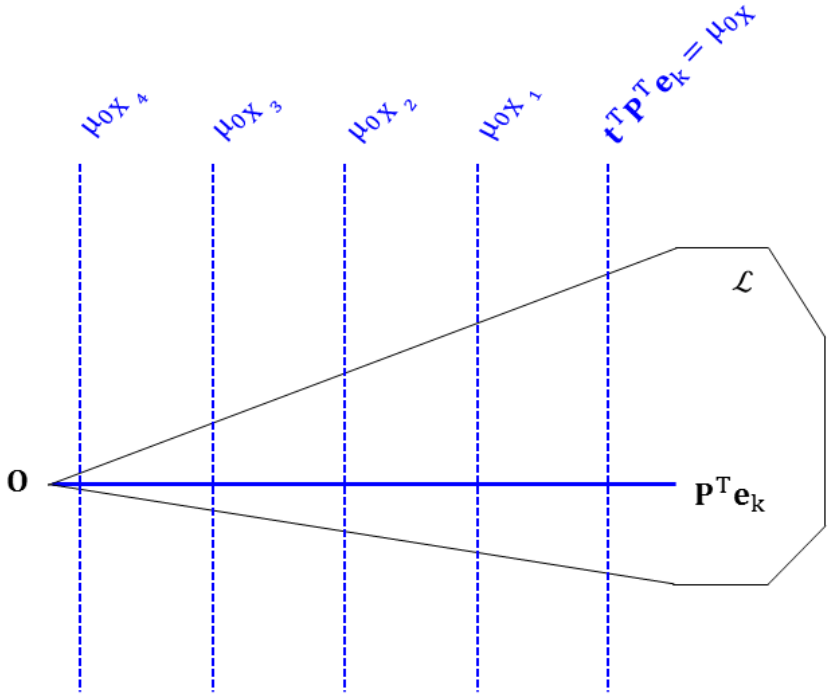

Figure 5.1 A schematic of the construction of the prediction axis for the k^{th} predictor variable in the PLS biplot plane $\mathcal{L}$.

To facilitate orthogonal projection onto the prediction biplot axes, similar to scatter plots, the line through the origin orthogonal to (5.7) is selected as the biplot axis for variable k. Any point on this biplot axis will have the form $\phi\mathbf{P}^T\mathbf{e}_k$. As a result, the point on the biplot axis predicting the value μ_{0X} for the k^{th} predictor variable will have

$$\mathbf{t}_{\mu_{0X}} = \phi\mathbf{P}^T\mathbf{e}_k. \tag{5.8}$$

Substituting (5.8) into (5.7) yields $\mu_{0X} = \phi\mathbf{e}_k^T\mathbf{P}\mathbf{P}^T\mathbf{e}_k$. Solving for ϕ then yields

$$\phi = \frac{\mu_{0X}}{\mathbf{e}_k^T\mathbf{P}\mathbf{P}^T\mathbf{e}_k}. \tag{5.9}$$

Furthermore, replacing ϕ in (5.8) by (5.9) gives the marker μ_{0X} on the k^{th} prediction biplot axis as

$$\frac{\mu_{0X}}{\mathbf{e}_k^T\mathbf{P}\mathbf{P}^T\mathbf{e}_k}\mathbf{P}^T\mathbf{e}_k. \quad (5.10)$$

Similarly, from (5.2), any point predicting $\mu_{0Y} \in (-\infty, \infty)$ for the k^{th} centred response variable will have $\mu_{0Y} = \mathbf{t}^T\mathbf{Q}^T\mathbf{e}_k$, with the biplot axis of the form $\phi\mathbf{Q}^T\mathbf{e}_k$. For this reason, the point on the biplot axis predicting the value μ_{0Y} for the k^{th} response variable will have

$$\mathbf{t}_{\mu_{0Y}} = \phi\mathbf{Q}^T\mathbf{e}_k. \quad (5.11)$$

Substituting (5.11) into $\mu_{0Y} = \mathbf{t}^T\mathbf{Q}^T\mathbf{e}_k$ yields $\mu_{0Y} = \phi\mathbf{e}_k^T\mathbf{Q}\mathbf{Q}^T\mathbf{e}_k$. Solving for ϕ then yields

$$\phi = \frac{\mu_{0Y}}{\mathbf{e}_k^T\mathbf{Q}\mathbf{Q}^T\mathbf{e}_k}. \quad (5.12)$$

Moreover, replacing ϕ in (5.11) by (5.12) gives the marker μ_{0Y} on the k^{th} response prediction biplot axis as

$$\frac{\mu_{0Y}}{\mathbf{e}_k^T\mathbf{Q}\mathbf{Q}^T\mathbf{e}_k}\mathbf{Q}^T\mathbf{e}_k. \quad (5.13)$$

The values μ_{0X} and μ_{0Y} in (5.10) and (5.13) are in terms of the centred samples. Thus, calibration markers are fitted using sensible scale values

$$\mu_X = \mu_{0X} + \bar{x}_k \text{ and } \mu_Y = \mu_{0Y} + \bar{y}_k$$

where $\bar{x}_k$ and $\bar{y}_k$ are the mean of the k^{th} predictor and response variables respectively. In other words, to trace the prediction biplot axis for the k^{th} original predictor and response variables, values of μ_X and μ_Y are substituted in the place of μ_{0X} and μ_{0Y} in (5.10) and (5.13) respectively.

With the PLSR coefficients matrix defined as $\hat{\mathbf{B}}_{PLSR} = \mathbf{R}\mathbf{Q}^T$, the i^{th} row of $\hat{\mathbf{B}}_{PLSR}$ can be written as $\hat{\mathbf{b}}_{(i)}^T = \mathbf{r}_{(i)}^T\mathbf{Q}^T$, akin to $\mathbf{y}_0^T = \mathbf{t}^T\mathbf{Q}^T$, so that the regression coefficients are predicted by the prediction biplot axes defining the response variables. Instead of predicting a sample point $\mathbf{t}^T = \mathbf{x}_{0_i}^T\mathbf{R}$, $\mathbf{e}_i^T\mathbf{R}$ are projected onto these axes. That is, projecting each of the rows of $\mathbf{R}$ onto the prediction axes defined by $\mathbf{Q}$ yields the estimated PLSR coefficients matrix $\hat{\mathbf{B}}_{PLSR}$. This is true without adjusting for centring, so here, $\mu_b = \mu_{0Y}$ and not $\mu_Y = \mu_{0Y} + \bar{y}_k$. Hence, *two different sets of prediction marker calibrations are needed on each of the response axes, a set corrected for the mean for* $\mathbf{Y}$ *(in black) and a set in terms of the centred values for* $\hat{\mathbf{B}}_{PLSR}$ *(in purple).*

Furthermore, if the RMSEP plot indicates that the PLSR should be performed in $A > 2$ components, the PLS biplot can still be constructed in two dimensions. However, the predictions obtained with the biplot axes will be the two-dimensional approximations of $\hat{\mathbf{B}}_{PLSR}$ and the two-dimensional approximations of the A-dimensional approximations of the matrices $\mathbf{X}_0$ and $\mathbf{Y}_0$. In this situation, instead of constructing the PLS biplot from the first two components, any two

components $1 \leq i, j \leq A$ can be used, but any such plot remains a two-dimensional approximation of the A-dimensional solution.

5.3 Measures of fit for the PLS biplot

To determine the adequacy of the representation of the original data matrix $\mathbf{D} = [\mathbf{X} \quad \mathbf{Y}]$, the quality of the representation provided by the PLS biplot is required. Barring $\mathbf{1}\bar{\mathbf{x}}$ and $\mathbf{1}\bar{\mathbf{y}}$ in (5.3), $\widehat{\mathbf{D}}$ can be written as

$$\begin{aligned} \widehat{\mathbf{D}} = [\mathbf{T}\mathbf{P}^{\mathrm{T}} \quad \mathbf{T}\mathbf{Q}^{\mathrm{T}}] &= [\mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{X} \quad \mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{Y}] \\ &= \mathbf{T}\mathbf{T}^{\mathrm{T}}[\mathbf{X} \quad \mathbf{Y}] \\ &= \mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D} \end{aligned}$$

where $\mathbf{D} = [\mathbf{X} \quad \mathbf{Y}]$. Suppose $\mathbf{D}$ is partitioned such that

$$\begin{aligned} \mathbf{D} &= \widehat{\mathbf{D}} + \left(\mathbf{D} - \widehat{\mathbf{D}}\right) \\ &= \mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D} + (\mathbf{D} - \mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D}) \\ &= \mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D} + (\mathbf{I}_{\mathrm{N}} - \mathbf{T}\mathbf{T}^{\mathrm{T}})\mathbf{D}. \end{aligned} \tag{5.14}$$

It follows that

$$\begin{aligned} \widehat{\mathbf{D}}^{\mathrm{T}}\left(\mathbf{D} - \widehat{\mathbf{D}}\right) = \mathbf{D}^{\mathrm{T}}\mathbf{T}\mathbf{T}^{\mathrm{T}}(\mathbf{D} - \mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D}) &= \mathbf{D}^{\mathrm{T}}\mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D} - \mathbf{D}^{\mathrm{T}}\mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D} \\ &= \mathbf{D}^{\mathrm{T}}\mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D} - \mathbf{D}^{\mathrm{T}}\mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D} \\ &= \mathbf{0} \end{aligned}$$

giving rise to Type B orthogonality:

$$\begin{aligned} \mathbf{D}^{\mathrm{T}}\mathbf{D} &= \mathbf{D}^{\mathrm{T}}[\mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D} + (\mathbf{I}_{\mathrm{N}} - \mathbf{T}\mathbf{T}^{\mathrm{T}})\mathbf{D}] \\ &= \mathbf{D}^{\mathrm{T}}\mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D} + \mathbf{D}^{\mathrm{T}}(\mathbf{I}_{\mathrm{N}} - \mathbf{T}\mathbf{T}^{\mathrm{T}})\mathbf{D} \end{aligned} \tag{5.15}$$

where $\mathbf{T}^{\mathrm{T}}\mathbf{T} = \mathbf{I}_{\mathrm{A}}$. However, there is no orthogonal decomposition of $\mathbf{D}\mathbf{D}^{\mathrm{T}}$ due to

$$\begin{aligned} \widehat{\mathbf{D}}\left(\mathbf{D} - \widehat{\mathbf{D}}\right)^{\mathrm{T}} = \mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D}(\mathbf{D} - \mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D})^{\mathrm{T}} &= \mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D}[\mathbf{D}(\mathbf{I}_{\mathrm{N}} - \mathbf{T}\mathbf{T}^{\mathrm{T}})]^{\mathrm{T}} \\ &= \mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D}\mathbf{D}^{\mathrm{T}} - \mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{T}\mathbf{T}^{\mathrm{T}} \\ &\neq \mathbf{0} \end{aligned}$$

where $(\mathbf{I}_{\mathrm{N}} - \mathbf{T}\mathbf{T}^{\mathrm{T}})^{\mathrm{T}} = (\mathbf{I}_{\mathrm{N}} - \mathbf{T}\mathbf{T}^{\mathrm{T}})$. Thus, from (5.15), only the axis predictivity measure (see Section 2.11) will be evaluated and this is done as

$$\text{Axis predictivity} = \operatorname{diag}\left(\widehat{\mathbf{D}}^{\mathrm{T}}\widehat{\mathbf{D}}\right)[\operatorname{diag}(\mathbf{D}^{\mathrm{T}}\mathbf{D})]^{-1}.$$

Analogous to the PCA biplot, the overall quality of approximation provided by the PLS biplot can be measured in terms of the percentage of variation in $\mathbf{D}$ explained by the extracted latent variables $\mathbf{T}$. More precisely,

$$\begin{aligned} \text{Overall quality} &= \operatorname{sum}\left(\operatorname{diag}\left(\widehat{\mathbf{D}}^{\mathrm{T}}\widehat{\mathbf{D}}\right)\right)\left[\operatorname{sum}\left(\operatorname{diag}(\mathbf{D}^{\mathrm{T}}\mathbf{D})\right)\right]^{-1} \\ &= \operatorname{sum}\left(\operatorname{diag}(\mathbf{D}^{\mathrm{T}}\mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D})\right)\left[\operatorname{sum}\left(\operatorname{diag}(\mathbf{D}^{\mathrm{T}}\mathbf{D})\right)\right]^{-1} \\ &= \operatorname{sum}\left(\operatorname{diag}(\mathbf{D}^{\mathrm{T}}\mathbf{T}\mathbf{T}^{\mathrm{T}}\mathbf{D})\right)\left[\operatorname{sum}\left(\operatorname{diag}(\mathbf{D}^{\mathrm{T}}\mathbf{D})\right)\right]^{-1} \end{aligned} \tag{5.16}$$

where $\mathbf{T}^{\mathrm{T}}\mathbf{T} = \mathbf{I}_{\mathrm{A}}$.

5.4 Example

The following example is an illustration of the PLS biplot, using the olive oil data discussed in Section 2.8. The five chemical quality measurements (Acidity, Peroxide, K232, K270 and DK) and the six sensory panel characteristics (Yellow, Green, Brown, Glossy, Transp and Syrup) are assigned as the predictor and response variables respectively. The original data is shown in Table 2.2. This data was later standardized for the PLS analysis. A data is standardized by first centring it and then dividing each variable by their respective standard deviation. A 2-component PLS was performed on the standardized data using the SIMPLS algorithm (Algorithm 3.4). The asymmetric PLS biplot is shown in Figure 5.2. Here, the samples of the data are given by the red points, while the purple points are for the (PLSR) coefficient points. The predictor variables of the data are represented by the blue axes, while the response variables are represented by the black axes.

In this biplot (Figure 5.2), the sets of tick markers (blue and black) on the axes have been adjusted for standardization. That is, the calibration markers are fitted using sensible scale values

$$\mu_X = (\mu_{0_X} + \bar{x}_k)s_{X_k} \text{ and } \mu_Y = (\mu_{0_Y} + \bar{y}_k)s_{Y_k} \quad (5.16)$$

where $\bar{x}_k$, $\bar{y}_k$, s_{X_k} and s_{Y_k} are the means and standard deviations of the k^{th} predictor and response variables respectively. However, for reading the coefficient values, the calibration markers (purple) on the response axes are fitted using sensible scale values $\mu_b = \mu_{0_Y}$.

Furthermore, *in the PLS biplot display* (Figure 5.2), *a representation of the variance of each variable is shown*. Here, the representation is *expressed as the thicker arrow (vector) on each axis*. These vectors correspond to one (centred) unit on the biplot axes. Observing the angles between the blue vectors, all the predictor variables can be said to be positively related to each other. Similarly, observing the angles between the black vectors, Glossy, Transp and Yellow can be said to be positively related to each other, likewise Green, Syrup and Brown. However, these two collections of relationships have a negative relation between them. For example, a negative relation exists between Green and Yellow. The actual correlation values of this data are shown in Figure 2.4 or Table 4.3.

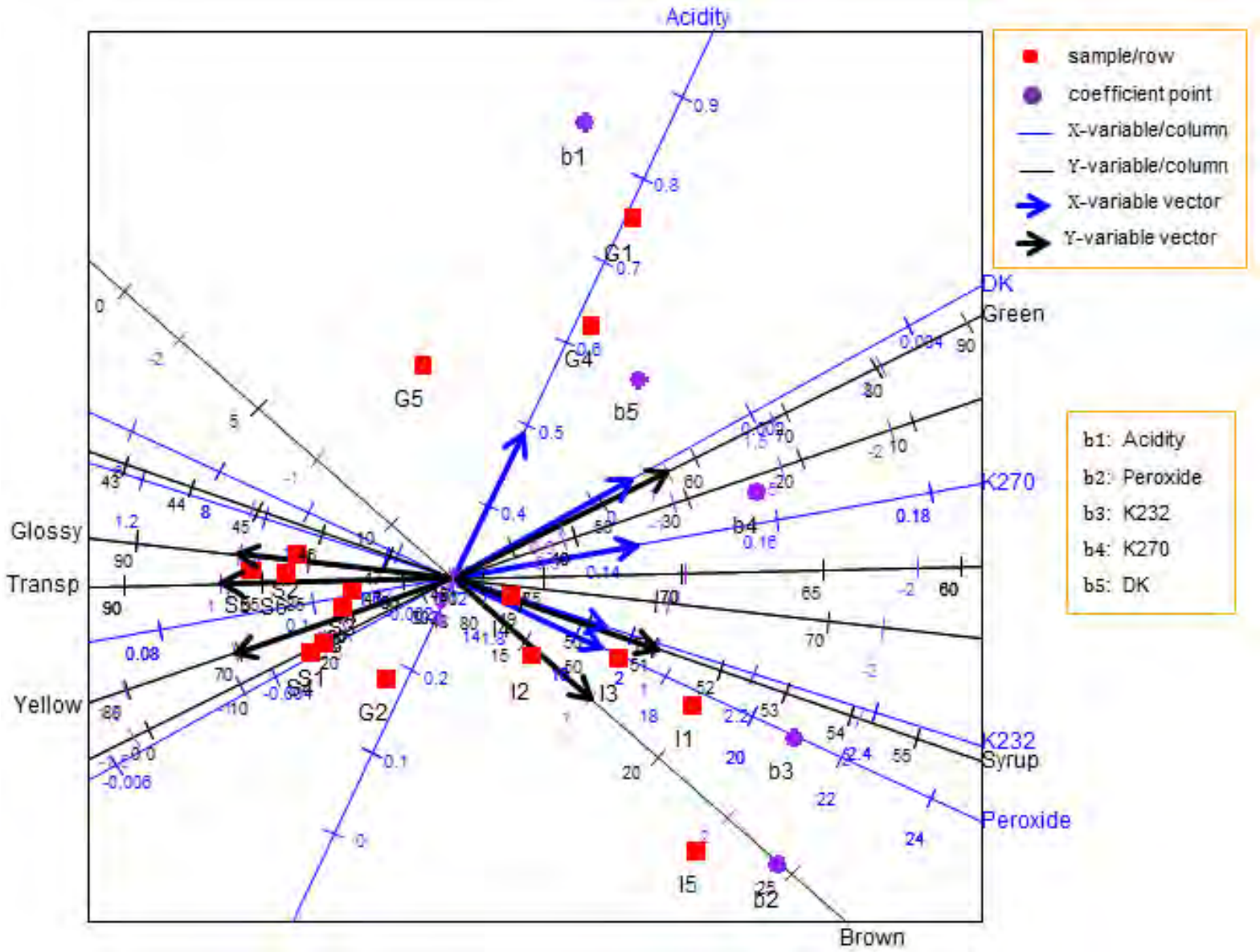

Figure 5. 2 The PLS biplot of the olive oil data, using the SIMPLS algorithm (Algorithm 3.4).

Table 5.1 The approximated olive oil values.

	Acidity	Peroxide	K232	K270	DK	Yellow	Green	Brown	Glossy	Transp	Syrup
G1	0.8	13.5	1.8	0.149	0.002	26.7	65.1	9.5	76.9	71.5	48.7
G2	0.2	12.9	1.7	0.108	-0.003	58.9	23.2	12.8	82.4	80.7	47.5
G3	0.2	11.5	1.6	0.104	-0.003	60.5	22.4	10.6	83.9	82.2	46.6
G4	0.6	13.7	1.8	0.141	0.001	32.9	56.8	10.6	77.7	73.0	48.7
G5	0.5	10.9	1.6	0.119	-0.001	47.5	40.2	7.7	82.4	79.1	46.6
I1	0.3	18.8	2.1	0.146	0.000	35.2	49.1	19.8	73.6	69.7	51.8
I2	0.3	15.4	1.9	0.126	-0.001	46.8	36.9	15.5	78.3	75.5	49.4
I3	0.3	17.0	2.0	0.137	-0.001	39.9	44.5	17.3	75.8	72.3	50.6
I4	0.3	14.5	1.8	0.125	-0.001	46.8	37.8	13.9	79.1	76.2	48.8
I5	0.2	20.1	2.2	0.143	-0.001	38.8	43.2	22.6	72.9	69.6	52.6
S1	0.2	11.4	1.6	0.100	-0.003	63.0	19.2	10.8	84.3	82.9	46.5
S2	0.3	10.2	1.5	0.099	-0.003	62.7	20.7	8.6	85.3	83.8	45.8
S3	0.3	11.5	1.6	0.105	-0.003	59.2	24.1	10.5	83.6	81.8	46.7
S4	0.2	11.3	1.5	0.099	-0.004	64.2	17.8	10.7	84.6	83.3	46.4
S5	0.2	9.4	1.4	0.093	-0.004	66.8	16.0	7.9	86.6	85.4	45.2
S6	0.2	10.1	1.5	0.097	-0.003	64.0	19.1	8.7	85.6	84.2	45.7

Table 5.2 Axis predictivity of the PLS biplot of the olive oil data.

Acidity	Peroxide	K232	K270	DK	Yellow	Green	Brown	Glossy	Transp	Syrup
0.988	0.995	0.999	0.989	0.753	0.934	0.819	0.963	0.997	0.994	0.998

With an overall quality of 0.972, various inter-variable relationships can be deduced from the biplot (Figure 5.2), such as a relation between the response Syrup and the predictors K232 and Peroxide; between predictor DK and responses Green and Yellow, as well as, between responses Glossy and Transp and predictor K270. Acidity can be seen to have no clear relation with the others.

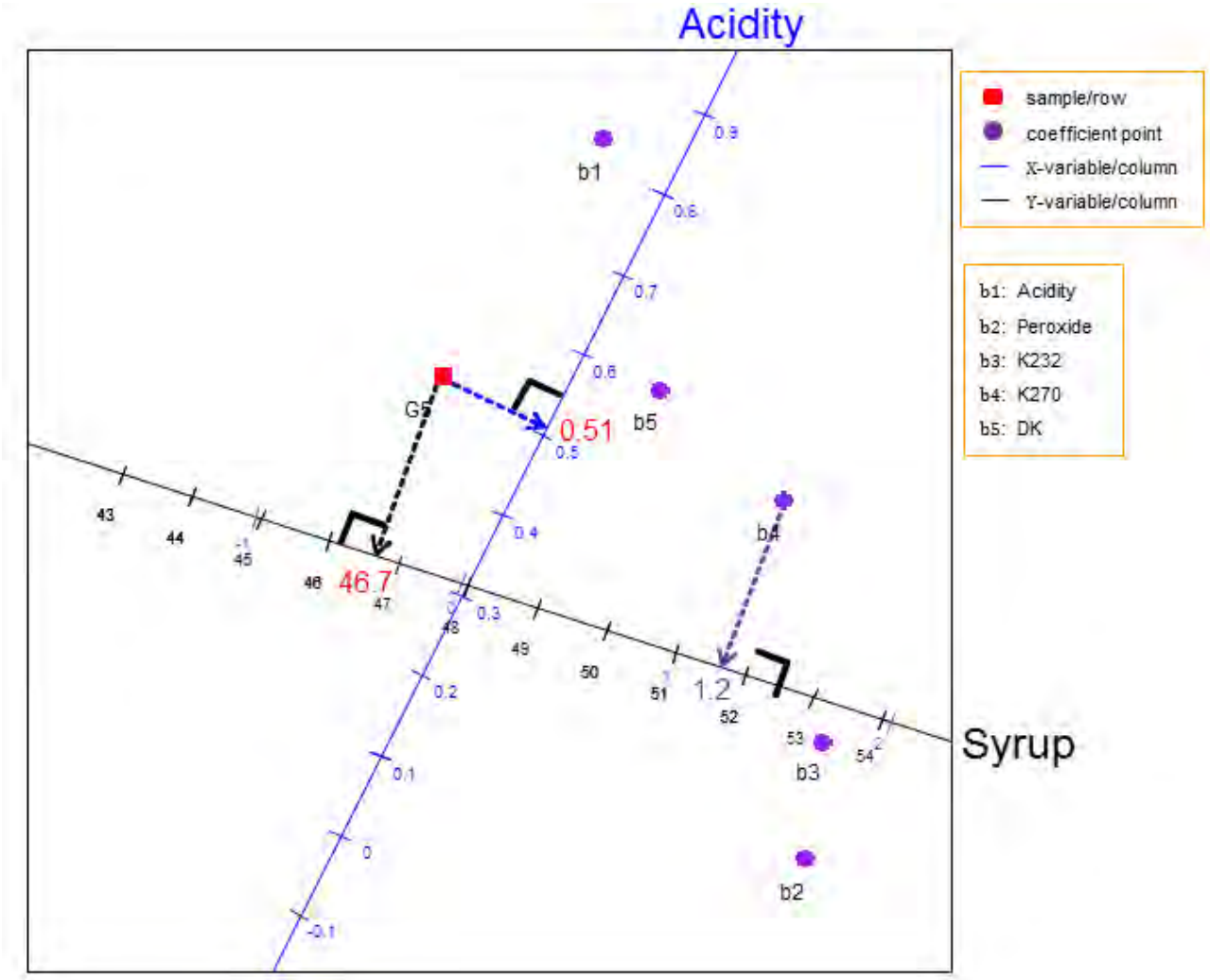

Figure 5.3 Examples of orthogonal projections in the PLS biplot of the olive oil data.

Each sample point in the PLS biplot is orthogonally projected onto the axes, and the respective values are read off to give the approximated values of the olive oil data. For example, sample point G5 projected onto the Acidity and Syrup axes yields the values 0.51 and 46.7 respectively, as shown above in Figure 5.3. Likewise, to get the approximated coefficient values from the biplot, the PLSR *coefficient points* bi, *for* $i = 1, 2, \cdots, 5$, *are projected onto the prediction axes representing the sensory panel characteristics*. However, *the purple markers on these axes are used to read off the coefficient values*. Since the coefficients matrix $\widehat{\mathbf{B}}_{\text{PLSR}} = \mathbf{R}\mathbf{Q}^{\text{T}} = \mathbf{R}\mathbf{T}^{\text{T}}\mathbf{Y}_0$ (Section 3.6), where $\mathbf{Q} = \mathbf{Y}_0^{\text{T}}\mathbf{T}$ are defined in terms of the centred Y-variables,

and the matrix **Q** used in calculating the directions of the biplot axes representing the Y-variables (responses), *two sets of calibrations marker will be needed on these response axes*. To be precise, *a set (in black) corrected for the mean for* **Y** *and a set (in purple) in terms of the centred values for* $\widehat{\mathbf{B}}_{\text{PLSR}}$. Thus, the *purple markers on the response axes will be used to read off the coefficient values.* For example, point b4 projected onto the Syrup axis gives a value of 1.2, and not 51.7, as shown above in Figure 5.3. This value, 1.2, can be seen to be the PLS regression coefficient for K270 under Syrup, in Table 5.3 below. The approximated values of the olive oil data as well as the estimated regression coefficient values are shown in Tables 5.1 and 5.3 respectively.

Moreover, the predictivity of each biplot axis is estimated, and this is shown in Table 5.2. Each of these axes quite well represents the original data, with the K232 axis having the highest axis predictivity value of 0.999, followed by the Syrup axis with 0.998. However, the DK axis has the lowest axis predictivity of 0.753. This value for the DK axis is the ratio of the predicted sum-of-squares to the total sum-of-squares for all the observations under variable DK. An axis predictivity of 1.000 means that all values can be read off the axis exactly. The lower the axis predictivity value, the less accurately the axis approximates the observed values under that variable. Thus, axis DK having the lowest axis predictivity value of 0.753 means that the axis represents the original data, but not quite as accurate as the other axes. These predictivities and the overall quality value (0.972) indicate that the PLS biplot approximates the olive oil data well.

Table 5.3 The estimated (SIMPLS) PLSR coefficient values.

	Yellow	Green	Brown	Glossy	Transp	Syrup
b1: **Acidity**	-1.175	1.343	-1.112	-0.357	-0.604	-0.119
b2: **Peroxide**	-0.921	0.675	2.379	-1.638	-1.364	1.862
b3: **K232**	-1.169	0.977	1.984	-1.646	-1.448	1.741
b4: **K270**	-1.365	1.301	0.942	-1.348	-1.314	1.206
b5: **DK**	-1.042	1.071	0.046	-0.748	-0.818	0.515

For comparison reasons, a 2-component PLS was performed using the kernel algorithm of Rännar *et al*. (1994), and the resulting PLS biplot is shown in Figure 5.4. This biplot looks quite similar to the PLS biplot in Figure 5.2. However, the positions of the coefficient points are different in both biplots. This is due to the way the PLS X-weights matrix **R** was computed in both algorithms (see Algorithms 3.2 and 3.4). The estimated regression coefficient values obtained from the kernel-PLS analysis are shown in Table 5.4.

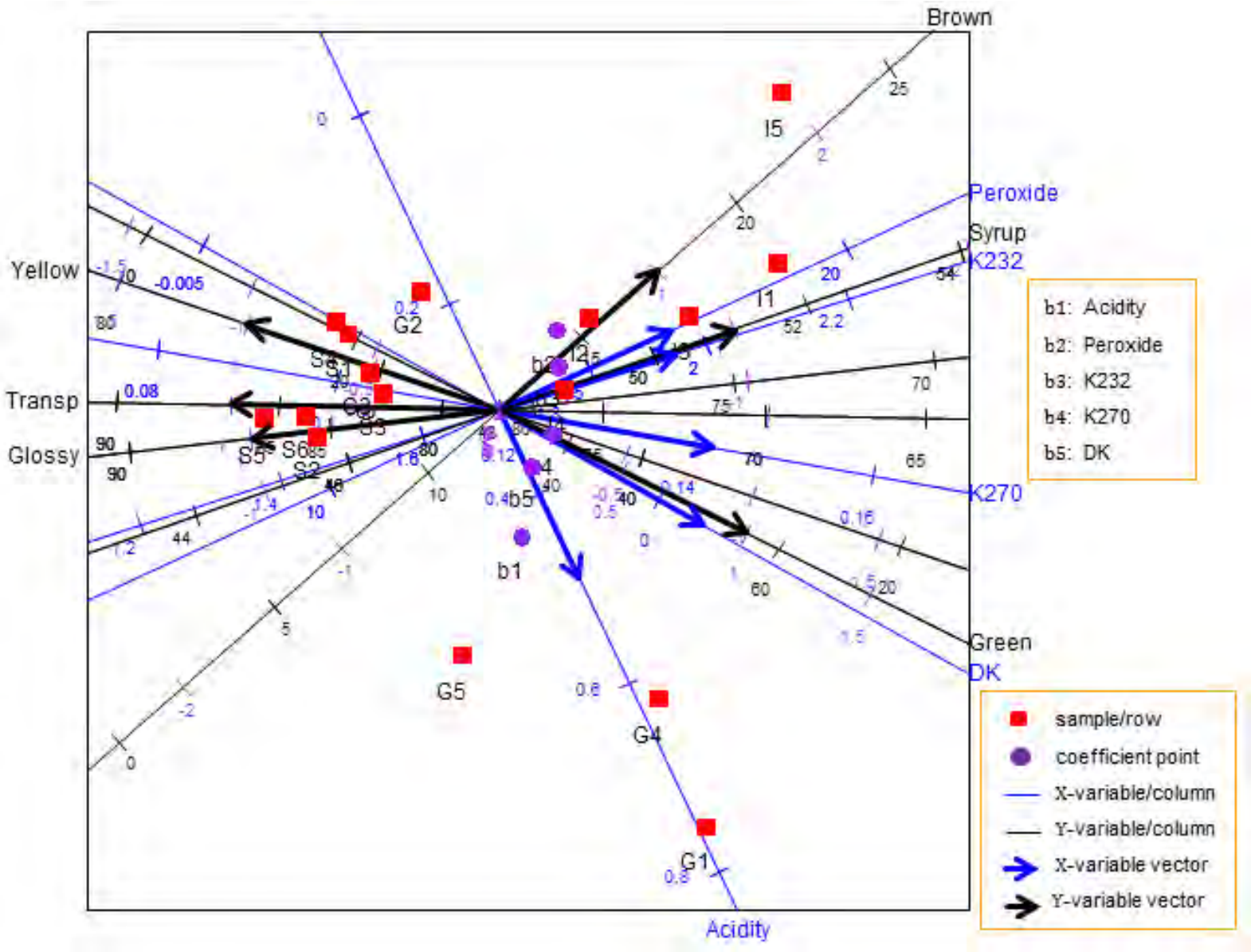

Figure 5. 4 The kernel-PLS biplot of the olive oil data (Algorithm 3.2).

Table 5.4 The estimated (kernel) PLSR coefficient values.

	Yellow	Green	Brown	Glossy	Transp	Syrup
b1: Acidity	-0.233	0.277	-0.313	-0.033	-0.096	-0.078
b2: Peroxide	-0.105	0.056	0.453	-0.262	-0.205	0.321
b3: K232	-0.159	0.123	0.352	-0.258	-0.219	0.286
b4: K270	-0.217	0.211	0.117	-0.201	-0.200	0.172
b5: DK	-0.183	0.197	-0.059	-0.104	-0.126	0.051

Generally, there are many different algorithms for performing PLS, out of which three (NIPALS, kernel and SIMPLS) were discussed in Chapter 3 of this dissertation. Refer to Section 3.5. The different algorithms produce slightly different results, especially, with regards to the X-weights (**R**) and X-scores (**T**) matrices. This is due to the way these matrices are computed in each algorithm, as shown in Table 3.1. As yet, no general rule exist as to which algorithm is best. Among the three algorithms discussed in this dissertation, *the NIPALS algorithm was develop first. It can be computationally intensive when dealing with (very) large data sets*, due to the iterative updating of the centred data matrices $\mathbf{X}_0$ and $\mathbf{Y}_0$ (step (7) of Algorithm 3.1). For this reason, the kernel algorithms, by Lindgren *et al*. (1993) and Rännar *et al*. (1994), were developed. These algorithms are based on the NIPALS algorithm, but are less computationally intensive. *The algorithm developed by Lindgren et al. (1993) handles large data with fewer variables, while the algorithm by Rännar et al. (1994) handles large data with fewer samples*. On the other hand, the *SIMPLS algorithm is slightly different and yields a different solution, but has the advantage of no iterative updating of the centred data matrices*. In addition, the X-weights (**R**) are directly obtained in terms of the original $\mathbf{X}_0$, while in the NIPALS and kernel algorithms, they are obtained in terms of the updated $\mathbf{X}_0$, thus, requiring another transformation step to convert them into **R**. See Table 3.1 and Section 3.5 for more details.

5.5 Area biplots

Besides projecting the coefficient points bi, for $\mathrm{i} = 1, 2, \cdots, \mathrm{P}$, onto the prediction axes representing the response variables to get their respective values, an alternative method is proposed. This method is based on the area biplots idea developed by Gower *et al*. (2010). In general, area biplots require no calibrated axes, just the reader's eyes to compare areas of differently shaped triangles. To be precise, it involves

(i) a 90° rotation of one set of points, in this case, the coefficient points bi, for $\mathrm{i} = 1, 2, \cdots, \mathrm{P}$,

(ii) the area spanned by a triangle of the rotated coefficient points,

(iii) one unit on the response variable axis, and

(iv) the origin

to approximate a data value. Although it is not intuitive to estimate the areas of triangles by eyes, for the exact estimated PLSR coefficient values, the coefficients matrix $\widehat{\mathbf{B}}_{\text{PLSR}} = \mathbf{RQ}^{\text{T}}$ can be examined. The area biplot method provides an easy way of comparing the relative sizes of the PLSR coefficients graphically, thereby, providing an instinctive display of the importance of the effect of the predictors on each of the response variables. Since the i^{th} row of the PLSR coefficients matrix $\widehat{\mathbf{B}}_{\text{PLSR}} = \mathbf{RQ}^{\text{T}}$ can be written as $\hat{\mathbf{b}}_{(\text{i})}^{\text{T}} = \mathbf{r}_{(\text{i})}^{\text{T}}\mathbf{Q}^{\text{T}}$, $\mathbf{r}_{(\text{i})}$ will serve as the row point defining the i^{th} coefficient point, while $\mathbf{Q}$ defines the unit points on the axes representing the response variables. Let $\boldsymbol{\vartheta}$ be a 90° rotation matrix. In the two-dimensional area biplots,

$$\boldsymbol{\vartheta} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \text{ for } \theta = \tfrac{\pi}{2} \text{ radians.}$$

Consider the row and j^{th} column points $\mathbf{r}_{(\text{i})}$ and $\mathbf{q}_{\text{j}}$ respectively. Rotating $\mathbf{r}_{(\text{i})}$ through 90° brings about $\boldsymbol{\vartheta}\mathbf{r}_{(\text{i})}$. The inner product of $\mathbf{r}_{(\text{i})}$ and $\mathbf{q}_{\text{j}}$ is defined by

$$\mathbf{r}_{(\text{i})}^{\text{T}}\mathbf{q}_{\text{j}} = \|\mathbf{r}_{(\text{i})}\| \times \|\mathbf{q}_{\text{j}}\| \times \cos(\theta_{\text{ij}}) \qquad (5.17)$$

where θ_{ij} is the angle between $\mathbf{r}_{(\text{i})}$ and $\mathbf{q}_{\text{j}}$. With the 90° rotation choice in mind,

$$\cos(\theta_{\text{ij}}) = \sin\left(\theta_{\text{ij}} + \tfrac{\pi}{2}\right) = \sin(\phi_{\text{ij}})$$

for $\phi_{\text{ij}} = \theta_{\text{ij}} + \frac{\pi}{2}$. Thus, ϕ_{ij} is the angle between the rotated row point $\boldsymbol{\vartheta}\mathbf{r}_{(\text{i})}$ and column point $\mathbf{q}_{\text{j}}$. Replacing $\cos(\theta_{\text{ij}})$ in (5.17) by $\sin(\phi_{\text{ij}})$ yields the inner product of $\boldsymbol{\vartheta}\mathbf{r}_{(\text{i})}$ and $\mathbf{q}_{\text{j}}$ as

$$\mathbf{r}_{(\text{i})}^{\text{T}}\mathbf{q}_{\text{j}} = \|\mathbf{r}_{(\text{i})}\| \times \|\mathbf{q}_{\text{j}}\| \times \sin(\phi_{\text{ij}}) = \|\mathbf{r}_{(\text{i})}\| \times \sin(\phi_{\text{ij}}) \times \|\mathbf{q}_{\text{j}}\|. \qquad (5.18)$$

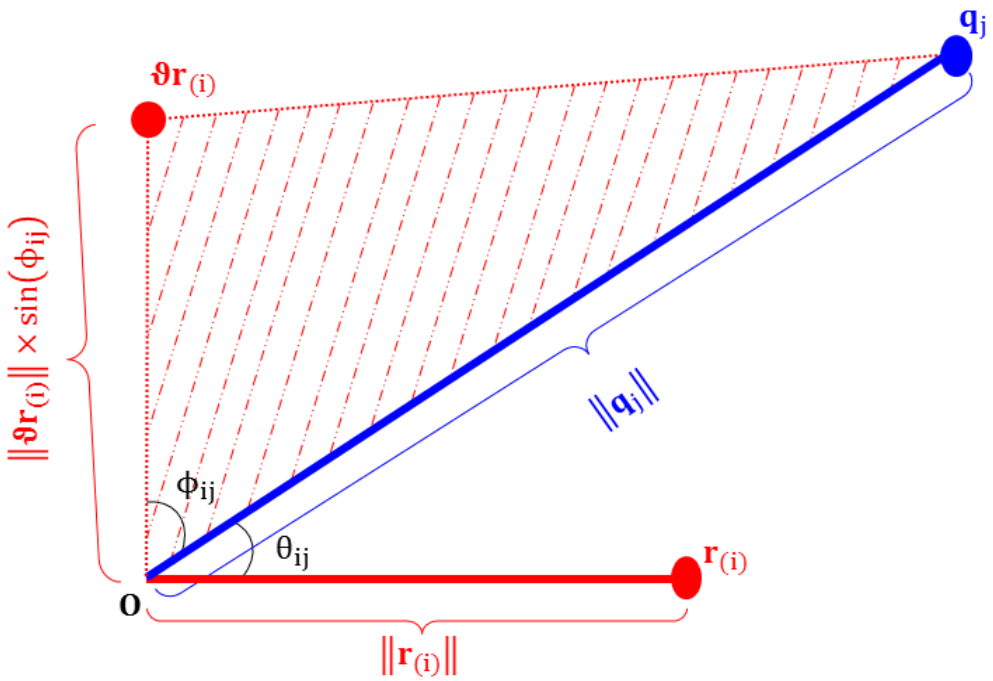

Figure 5.5 A schematic of the triangle area of $\mathbf{r}_{(\text{i})}^{\text{T}}\mathbf{q}_{\text{j}}$.

Figure 5.5 illustrates the construction of a triangle for a row point $\mathbf{r}_{(i)}$ and column point $\mathbf{q}_j$. From the origin $\mathbf{O}$, the height of the red triangle is given as $\|\vartheta\mathbf{r}_{(i)}\| \times \sin(\phi_{ij})$, while the base is defined by $\|\mathbf{q}_j\|$. With the area of a triangle defined as $\frac{1}{2}(\text{base} \times \text{height})$, the area of the red triangle is obtained as

$$\text{area of the red triangle} = \tfrac{1}{2}\left(\|\mathbf{q}_j\| \times \|\vartheta\mathbf{r}_{(i)}\| \times \sin(\phi_{ij})\right).$$

Twice this area yields

$$\begin{aligned} 2 \times \text{area of the red triangle} &= \|\mathbf{q}_j\| \times \|\vartheta\mathbf{r}_{(i)}\| \times \sin(\phi_{ij}) \\ &= \|\vartheta\mathbf{r}_{(i)}\| \times \|\mathbf{q}_j\| \times \sin(\phi_{ij}). \end{aligned} \qquad (5.19)$$

From $\widehat{\mathbf{B}}_{\text{PLSR}} = \mathbf{R}\mathbf{Q}^{\text{T}}$, $\hat{b}_{ij} = \mathbf{r}_{(i)}^{\text{T}}\mathbf{q}_j$. Since $\|\vartheta\mathbf{r}_{(i)}\| \times \|\mathbf{q}_j\| \times \sin(\phi_{ij})$ defines the inner product of $\vartheta\mathbf{r}_{(i)}$ and $\mathbf{q}_j$ (5.18), twice the area of the red triangle (5.19) approximates b_{ij}, the $(i,j)^{\text{th}}$ element of $\mathbf{B}_{\text{PLSR}}$.

Furthermore, the length of $\mathbf{r}_{(i)}$ can also be used to determine the length of one unit along axis $\mathbf{q}_j$. Recall from $\widehat{\mathbf{B}}_{\text{PLSR}} = \mathbf{R}\mathbf{Q}^{\text{T}}$ and $\hat{b}_{ij} = \mathbf{r}_{(i)}^{\text{T}}\mathbf{q}_j$, a particular coefficient value in row i of $\widehat{\mathbf{B}}_{\text{PLSR}}$ will be the product of two lengths, namely, the length of projection of $\mathbf{r}_{(i)}$ and the length of the biplot axis defined by $\mathbf{q}_j$. In other words,

$$\text{coefficient value in row i} = \begin{pmatrix}\text{length of projection}\\ \text{of}\\ \mathbf{r}_{(i)}\end{pmatrix} \times \begin{pmatrix}\text{length}\\ \text{of}\\ \text{biplot axis } \mathbf{q}_j\end{pmatrix}.$$

Fixing the coefficient value to be 1 and solving for length of projection yields the value of one unit on $\mathbf{q}_j$ as

$$\begin{pmatrix}\text{length of projection}\\ \text{of}\\ \text{one unit on } \mathbf{q}_j\end{pmatrix} = \frac{1}{\text{length of } \mathbf{q}_j}. \qquad (5.20)$$

Thus, one unit on axis $\mathbf{q}_j$ is inversely proportional to the length of $\mathbf{q}_j$. With the length of $\mathbf{q}_j$ defined by $\|\mathbf{q}_j\|$, (5.20) can be re-written as

$$\begin{pmatrix}\text{length of projection}\\ \text{of}\\ \text{one unit on } \mathbf{q}_j\end{pmatrix} = \frac{1}{\|\mathbf{q}_j\|}. \qquad (5.21)$$

Since the marker calibrations for $\widehat{\mathbf{B}}_{\text{PLSR}}$ are in terms of the centred value of $\mathbf{Y}$, see Subsection 5.2.3 above, the one unit being referred to here is a centred unit. Expression (5.21) can be referred to as the calibration factor (Section 2.6), used in determining how far apart the unit tick markers (in purple) are on axis $\mathbf{q}_j$. However, *since no calibrated axes are required in area biplots generally, these purple markers are not shown in the resulting area biplot displays, but*

can be seen in the PLS biplot displays. For example, see the black axes representing the Y-variables in the PLS biplot displays in Figures 5.2 and 5.4.

5.5.1 *Example*

Consider the PLS biplot of the olive oil data shown in Figure 5.2 above. Rotating the matrix $\widehat{\mathbf{B}}_{\text{PLSR}}$ through 90° brings about Figure 5.6 below. In order to obtain the approximated values of the PLSR coefficients b_{ij}, for $\text{i} = 1, 2, \cdots, 5$ and $\text{j} = 1, 2, \cdots, 6$, triangles spanned by the origin and each rotated bi, plus the prediction axes representing the sensory panel characteristics as bases, are added to the PLS biplot. For example, the triangle spanned by the origin and point b4, plus the Brown axis as base, is shown (in red) in Figure 5.6. This triangle has an area of $\frac{(17.2-12.3)/5.13}{2} = \frac{0.955}{2} = 0.478$. Twice this area gives the approximated value 0.955 for b4 under the Brown characteristic, as shown in Table 5.3. The values 12.3 and 5.13 are the mean and standard deviation of the Brown characteristic respectively, as shown in Table 5.5. Here, 17.2 is the value of the one unit on the Brown axis, needed for the red triangle. This value is calculated by the length of the Brown variable. In Figure 5.6, the length of the Brown variable is expressed as the thicker (black) arrow on the Brown axis.

Table 5.5 The column means and standard deviations of the Y-variables.

	Mean	Standard deviation
Yellow	50.9	19.46
Green	33.5	23.49
Brown	12.3	5.13
Glossy	80.8	6.19
Transp	78.2	8.31
Syrup	48.0	3.07

More than one triangle can be drawn at a time, on a particular axis. To illustrate, consider the points bi, for $\text{i} = 1, 2, \cdots, 5$, and the Green axis. *The triangles spanned by the origin and these points, plus the Green axis as bases, are shown in Figure* 5.7. However, caution is needed when dealing with large numbers of bi. It is not intuitive to estimate the exact area of a triangle visually, but as an exploratory tool, larger and smaller coefficients can be easily detected. *Large triangles indicate large coefficient values, while small triangles indicate small coefficient values*. From Figure 5.7, coefficient point b1 can be seen to have a larger value, followed by b4, b5 and b3. Point b2 has the smallest value. The exact coefficient values are obtained by printing out the coefficients matrix $\widehat{\mathbf{B}}_{\text{PLSR}}$, shown in Table 5.3 above.

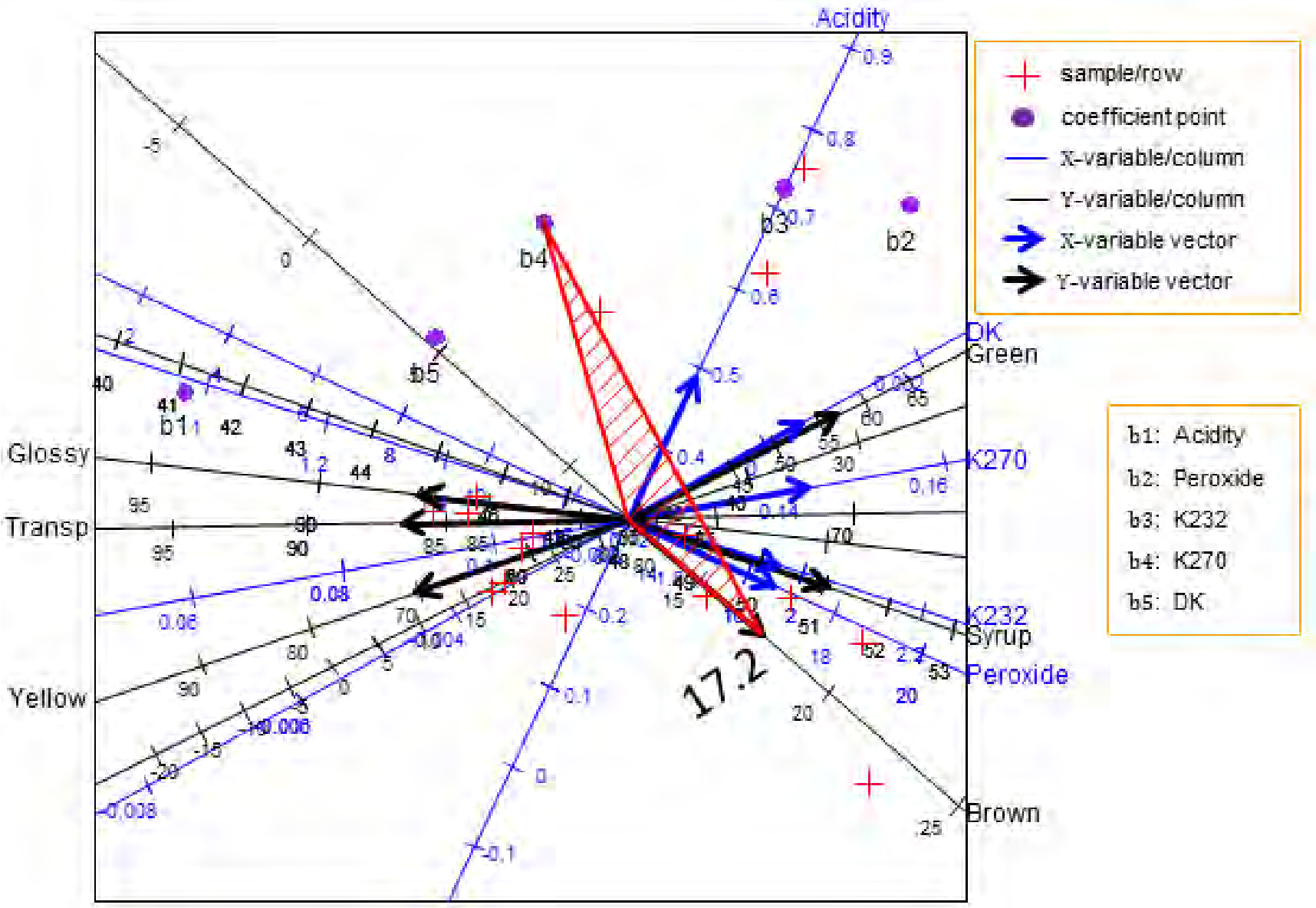

Figure 5. 6 Example of a triangle visualization in the PLS biplot of the olive oil data.

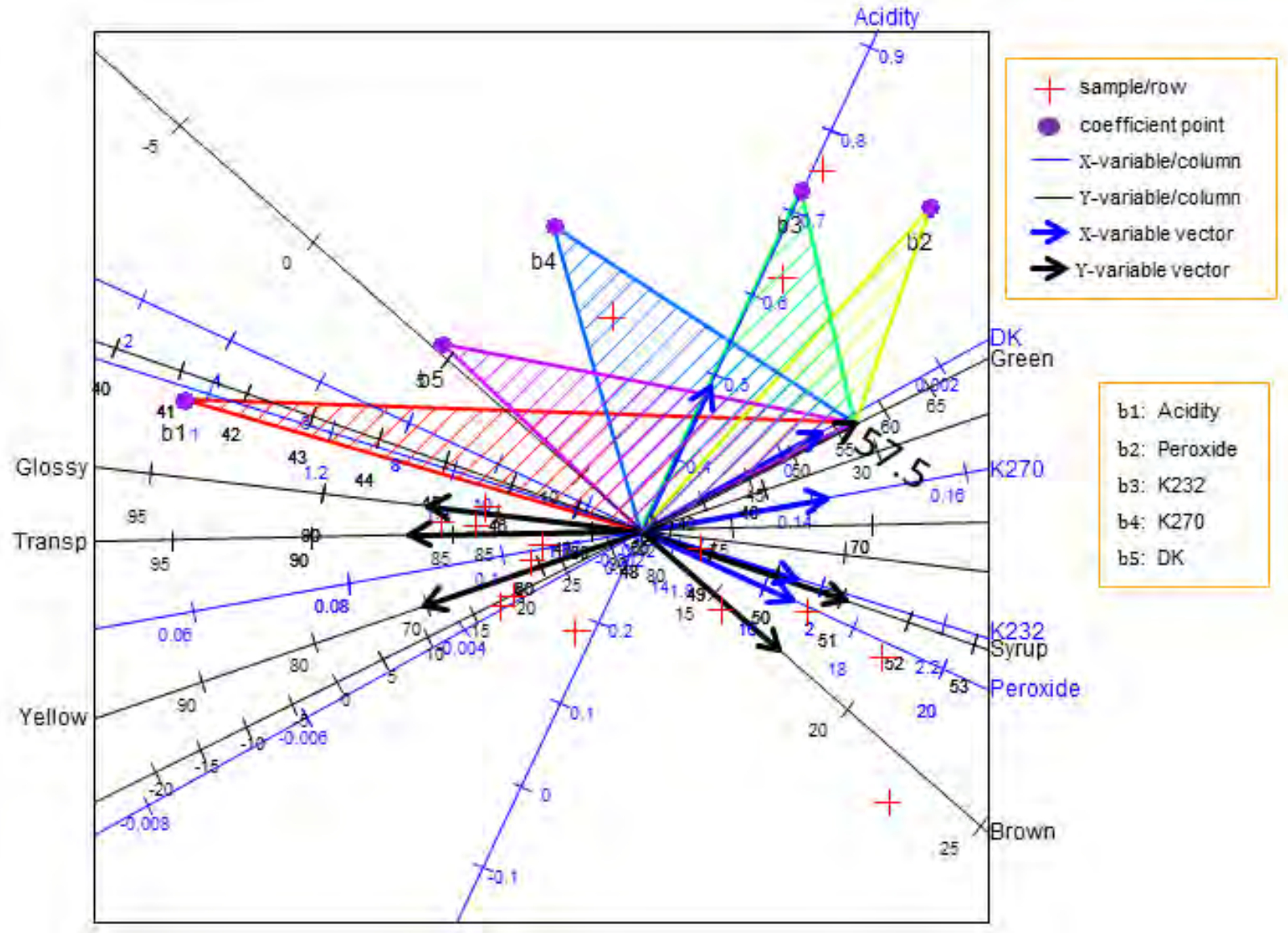

Figure 5. 7 The triangles for points bi, i = 1, 2, ⋯ , 5, with bases defined by the Green axis in the PLS biplot of the olive oil data.

5.6 Comparison with the PCA biplot

The PLS biplot and the PCA biplot are similar, in that they both aim to approximate a high-dimensional data matrix using only a few components. However, the method of approximation differs across these two biplots. Consider an $N \times (P + M)$ data matrix $\mathbf{D}$. For its PCA biplot, $\mathbf{D}$ is approximated as $\widehat{\mathbf{D}}_{PCA} = \mathbf{Z}\mathbf{V}_{[r]}^{T}$, by (2.10). As a result, the matrix $\mathbf{Z}\,(N \times r)$ gives information about the N rows of $\mathbf{D}$, while matrix $\mathbf{V}_{[r]}$ $((P + M) \times r)$ gives information about its $(P + M)$ columns. On the other hand, for PLS, $\mathbf{D}$ is approximated as $\widehat{\mathbf{D}}_{PLS} = \mathbf{T}[\mathbf{P}\ \ \mathbf{Q}]^{T}$ (5.3). Here, the matrix $\mathbf{T}\,(N \times A)$ contains information on the N rows of $\mathbf{D}$, while the matrix $\begin{bmatrix}\mathbf{P}\\ \mathbf{Q}\end{bmatrix}$ $((P + M) \times A)$ contains information on its $(P + M)$ columns. When compared to each other, the matrices $\mathbf{Z}$ and $\mathbf{T}$ differ in the way in which they are constructed, as is true for the matrices $\mathbf{V}_{[r]}$ and $\begin{bmatrix}\mathbf{P}\\ \mathbf{Q}\end{bmatrix}$. To be precise, *both* $\mathbf{Z}$ *and* $\mathbf{V}_{[r]}$ *are constructed through the utilization of PCA on* $\mathbf{D}$*, while* $\mathbf{T}$ *and* $\begin{bmatrix}\mathbf{P}\\ \mathbf{Q}\end{bmatrix}$ *are constructed through the application of PLS on* $\mathbf{D}$. To illustrate these biplot differences, consider the PCA biplot and the PLS biplot of the olive oil data, as shown in Figures 2.2 and 5.2 respectively. The displays are different as a result of their respective approximation method used in approximating the olive oil data. To be precise, there are two different sets of points (red and purple) in the PLS biplot (Figure 5.2), while only one sets of points (black) is seen in the PCA biplot (Figure 2.2). In other words, besides the sample points and biplot axes in the PLS biplot display, an additional set of points (purple) are added to the display. Furthermore, looking at their respective objective function, *PCA treats the two sets of variables* $\mathbf{X}$ *and* $\mathbf{Y}$ *as one set*, say,

$$\boldsymbol{\Upsilon} = (\mathbf{x}_1,\ \mathbf{x}_2, \cdots,\ \mathbf{x}_P,\ \mathbf{y}_1,\ \mathbf{y}_2, \cdots,\ \mathbf{y}_M) = (\boldsymbol{\upsilon}_1 \quad \boldsymbol{\upsilon}_2 \quad \cdots \quad \boldsymbol{\upsilon}_{P+M})$$

and *aims to maximize the variance within* $\boldsymbol{\Upsilon}$. On the other hand, *PLS treats the two sets of variables* $\mathbf{X}$ *and* $\mathbf{Y}$ *as two separate sets* and *aims to maximize the covariance between them* (Section 3.5). In other words,

PLS: maximize $\text{cov}(\mathbf{w}^{T}\mathbf{X}^{T}\mathbf{Y}\mathbf{Y}^{T}\mathbf{X}\mathbf{w}) = \text{cov}(\mathbf{t}^{T}\mathbf{Y}\mathbf{Y}^{T}\mathbf{t})$, subject to $\|\mathbf{t}_a^{T}\mathbf{t}_a\| = 1$

while

PCA: maximize $\text{cov}\left(\boldsymbol{\Upsilon}\mathbf{V}_{[r]}\mathbf{V}_{[r]}^{T}\right) = \text{var}\left(\boldsymbol{\Upsilon}\mathbf{V}_{[r]}\mathbf{V}_{[r]}^{T}\right)$, subject to $\mathbf{V}_{[r]}^{T}\mathbf{V}_{[r]} = \mathbf{I}_r$.

Also, the constraint imposed on the objective function differs across these two methods. To be exact, *the scores (*$\mathbf{T}$*) in PLS must be orthogonal* ($\mathbf{T}^{T}\mathbf{T} = \mathbf{I}_{[A]}$), while this is not the case in PCA. With PCA, $\widehat{\boldsymbol{\Upsilon}} = \boldsymbol{\Upsilon}\mathbf{V}_{[r]}\mathbf{V}_{[r]}^{T}$ and scores $\mathbf{Z} = \boldsymbol{\Upsilon}\mathbf{V}_{[r]}$. Here, the loadings ($\mathbf{V}_{[r]}$) must be orthogonal, i.e., $\mathbf{V}_{[r]}^{T}\mathbf{V}_{[r]} = \mathbf{I}_{[r]}$ but *the scores* ($\mathbf{Z}$*) are not orthogonal* ($\mathbf{Z}^{T}\mathbf{Z} \neq \mathbf{I}_{[r]}$). In addition, the position of the sample points and biplot axes, as well as the spreading of the sample points, in Figures 2.2

and 5.2 can be seen to be different in both biplots. Hence, different data approximation methods can result in different biplot displays.

5.7 Comparison with the covariance biplot

Generally, in covariance analysis, the roles of the two sets of variables under consideration can be interchanged without losing any information when estimating the covariance between them. In the covariance biplot construction, see Chapter 4, the covariance matrix (4.1) is used. That is,

$$\begin{aligned} \widehat{\mathbf{X}}_0^{\mathrm{T}}\widehat{\mathbf{Y}}_0 &= \mathbf{D}\boldsymbol{\Lambda}\mathbf{F}^{\mathrm{T}} \\ &= \mathbf{G}_{[\mathrm{r}]}\mathbf{H}_{[\mathrm{r}]}^{\mathrm{T}} \end{aligned} \tag{5.22}$$

where $\mathbf{G}_{[\mathrm{r}]}$ and $\mathbf{H}_{[\mathrm{r}]}$ contain the first r columns of $\mathbf{G}$ and $\mathbf{H}$ respectively. With a similar arrangement as in (2.2), only variables are involved in the covariance decompositions in (5.22). Thus, the matrix $\mathbf{G}_{[\mathrm{r}]}$ $(\mathrm{P}\times\mathrm{r})$ contains information on the X-variables, while the matrix $\mathbf{H}_{[\mathrm{r}]}$ $(\mathrm{M}\times\mathrm{r})$ contains information about the Y-variables. Seeing that PLS can be viewed as an approximation technique, it can also be used to approximate the covariance between $\mathbf{X}_0$ and $\mathbf{Y}_0$. From equations (5.1) and (5.2), the covariance matrix is estimated as

$$\underbrace{\widehat{\mathbf{X}}_0^{\mathrm{T}}\widehat{\mathbf{Y}}_0}_{(\mathrm{P}\times\mathrm{M})} = \mathbf{P}\mathbf{T}^{\mathrm{T}}\mathbf{T}\mathbf{Q}^{\mathrm{T}} = \mathbf{P}\mathbf{Q}^{\mathrm{T}} \tag{5.23}$$

with $(\mathbf{T}^{\mathrm{T}}\mathbf{T}) = \mathbf{I}_{\mathrm{A}}$. The PLS approximation of the covariance matrix (5.23) can be seen to have an arrangement similar to (2.2). However, only variables are represented in this decomposition. That is, the matrix $\mathbf{P}$ $(\mathrm{P}\times\mathrm{A})$ contains information about the X-variables, while the matrix $\mathbf{Q}$ $(\mathrm{M}\times\mathrm{A})$ contains information about the Y-variables. Furthermore, since the PLS algorithms are based on the decomposition of the covariance matrix between $\mathbf{X}_0$ and $\mathbf{Y}_0$, the PLS biplot can also be viewed as a graphical tool for displaying the approximated covariances between $\mathbf{X}_0$ and $\mathbf{Y}_0$.

Considering the main goal of PLSR, the roles of predictors $\mathbf{X}$ and responses $\mathbf{Y}$ are treated asymmetrically. That is, they are not interchangeable. Although not of interest in this dissertation, if these roles were interchangeable, the resulting PLSR analysis would be a *symmetric PLSR* (Abdi, 2010). In this analysis, the SVD is applied once on the covariance matrix between $\mathbf{X}_0$ and $\mathbf{Y}_0$ to obtain the latent variables $\mathbf{T}$. Let $(\mathrm{N}-1)\mathbf{S} = \mathbf{X}_0^{\mathrm{T}}\mathbf{Y}_0$ be the $(\mathrm{P}\times\mathrm{M})$ covariance matrix. By the SVD, $\mathbf{S} = \mathbf{D}\boldsymbol{\Lambda}\mathbf{F}^{\mathrm{T}}$. The A columns in the X-weights matrix $\mathbf{R}$ $(\mathrm{P}\times\mathrm{A})$ are defined as the first A left singular vectors of $\mathbf{S}$. That is, $\mathbf{R} = \mathbf{D}[\,,1{:}\,\mathrm{A}]$. The matrix $\mathbf{T}$ is then obtained by $\mathbf{T} = \mathbf{X}_0\mathbf{D}[\,,1{:}\,\mathrm{A}] = \mathbf{X}_0\mathbf{R}$. If the roles of $\mathbf{X}$ and $\mathbf{Y}$ are reversed, the $(\mathrm{M}\times\mathrm{P})$

covariance matrix $(N-1)\mathbf{S}^{T} = \mathbf{Y}_0^{T}\mathbf{X}_0$. By the SVD, $\mathbf{S}^{T} = \mathbf{F}\boldsymbol{\Lambda}\mathbf{D}^{T}$. In this situation, the A columns in $\mathbf{R}$ are defined as the first A right singular vectors of $\mathbf{S}^{T}$. That is, $\mathbf{R} = \mathbf{D}[\,, 1\!:\!A]$. Thus, the matrix $\mathbf{T}$ is obtained by $\mathbf{T} = \mathbf{X}_0\mathbf{D}[\,, 1\!:\!A] = \mathbf{X}_0\mathbf{R}$.

Furthermore, comparing the PLS biplot to the covariance biplot, from (5.23), the objective function of the PLS biplot, with regards to approximating the covariance between $\mathbf{X}_0$ and $\mathbf{Y}_0$, can be interpreted as

$$\text{maximize } \widehat{\mathbf{X}}_0^{T}\widehat{\mathbf{Y}}_0 = \mathbf{P}\mathbf{T}^{T}\mathbf{T}\mathbf{Q}^{T}, \text{ subject to } (\mathbf{T}^{T}\mathbf{T}) = \mathbf{I}_A,$$

while from (5.22) and Section 4.3, the objective function for the covariance biplot can be interpreted as optimally approximate the covariance matrix $\mathbf{X}_0^{T}\mathbf{Y}_0$ using the Eckart-Young theorem (Eckart & Young, 1936). Although the matrices $\mathbf{P}$ $(P \times A)$ and $\mathbf{G}_{[r]}$ $(P \times r)$ both provide the X-variables information, while $\mathbf{Q}$ $(M \times A)$ and $\mathbf{H}_{[r]}$ $(M \times r)$ provide the Y-variables information, the method used in obtaining these matrices differs. To be precise, $\mathbf{G}_{[r]}$ and $\mathbf{H}_{[r]}$ are obtained through an application of the SVD on the covariance matrix (4.1), while $\mathbf{P}$ and $\mathbf{Q}$ are obtained through the utilization of PLS on the data matrix. From Section 3.5, $\mathbf{P} = \mathbf{X}_0^{T}\mathbf{T}$ and $\mathbf{Q} = \mathbf{Y}_0^{T}\mathbf{T}$, while from Section 4.3, $\mathbf{G}_{[r]} = \mathbf{D}\boldsymbol{\Lambda}^{0.5}\mathbf{J}_r$ and $\mathbf{H}_{[r]} = \mathbf{F}\boldsymbol{\Lambda}^{0.5}\mathbf{J}_r$.

5.7.1 *Example*

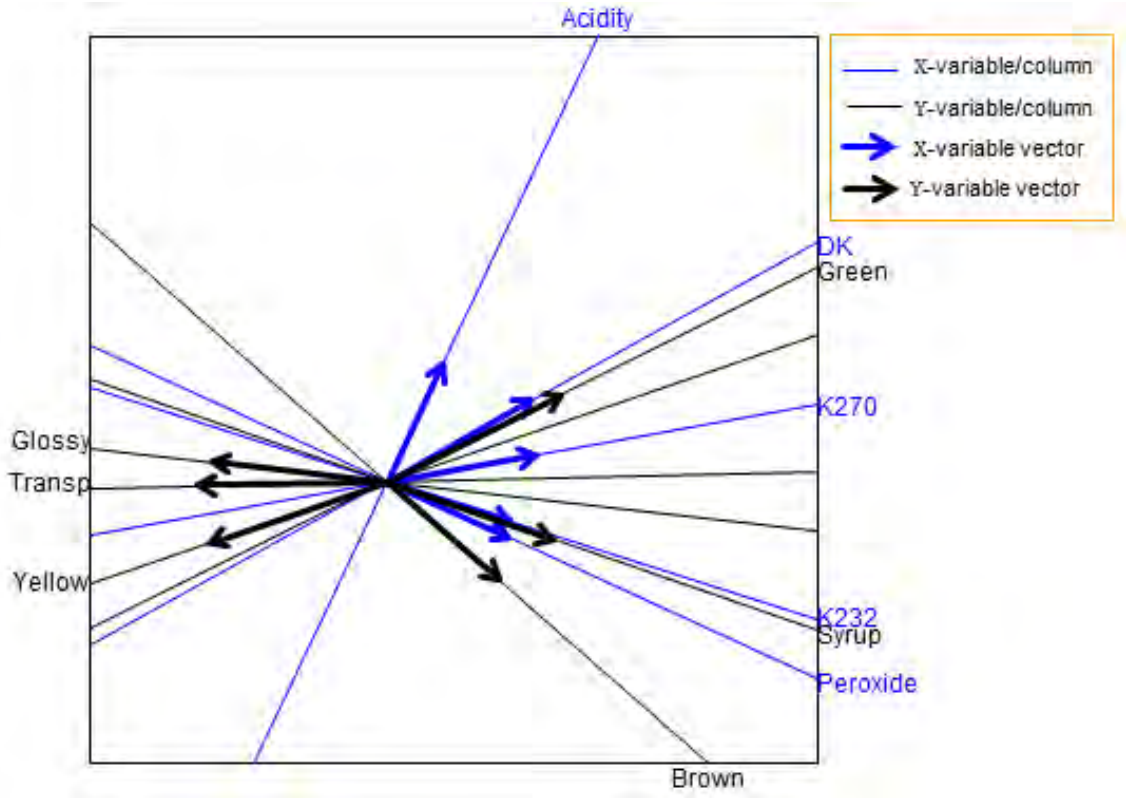

Figure 5.8 The PLS biplot of the olive oil data, without axes markers, sample and coefficient points.

Consider the PLS biplot of the olive oil data shown in Figure 5.2. *Excluding the sample (red) and coefficient (purple) points, as well as the axes calibration markers from this biplot brings about Figure* 5.8 (shown above). Now, consider the covariance biplot of the olive oil data shown in Figure 4.1. When compared with the PLS biplot in Figure 5.8, similar deductions can be reached. For example, a relation between predictor K270 and responses Glossy and Transp; as well as between response Syrup and predictors K232 and Peroxide can be observed from the PLS biplot (Figure 5.8). These deductions can also be made from the covariance biplot (Figure 4.1). Thus, it can be concluded that both biplots approximate the covariances between $\mathbf{X}_0$ and $\mathbf{Y}_0$ using different decomposition methods. However, different covariance decompositions can result in different covariance approximations and graphical displays. Since only variables are present in Figure 5.8, regardless of their roles, and with the definition of a monoplot in mind (see Section 4.2), Figure 5.8 can be viewed as a *PLS monoplot*.

5.8 Summary

Results found by the PLSR analysis of a data set can be visualized graphically using biplots, specifically, the PLS biplot. A PLS biplot provides a single graphical representation of the samples together with the predictor and response variables, as well as their inter-relationships in terms of the matrix of PLSR coefficients.

Based on the definition of the biplot, one can argue that the PLS biplot is actually a *PLS triplot*. Although not of interest in this dissertation, a *triplot*, by definition, is a joint graphical display that has three kinds of features displayed in it. These features can be specified as three sets of samples, or as two sets of samples and variables, as in the case of the PLS biplot. Observing the PLS biplot generally, three kinds of features can be identified, namely, (i) the samples, (ii) the variables, regardless of their roles, and (iii) the coefficient points. Thus, the PLS biplot can be viewed as a PLS triplot.

Analogous to the PCA biplot, the best A-dimensional plane passing through the origin is obtained first, and thereafter the orthogonal projections of the points in the high-dimensional plot of the data matrix are done onto this plane. The plane, with the projected points on it, is then extracted from the high-dimensional plot to the scaffolding axes on which the biplot is built. These scaffolding axes are not shown but are defined by the first A columns of the PLS (transformed) X-weights matrix $\mathbf{R}$. In the PLS biplot, points are defined by the extracted latent

variables (**T**), while the directions of the axes are calculated using the rows of the loadings matrices **P** and **Q** for the predictor and response variables respectively.

Furthermore, to determine the accuracy of the PLS biplot representation, the overall quality of approximation along with the axis predictivity must be evaluated. The overall quality of approximation is measured in terms of the percentage of variation in the data matrix explained by the extracted latent variables (**T**). Axis predictivity is measured by expressing the sum-of-squares of the approximated value for each variable in $\widehat{\mathbf{D}}$ as a proportion of their respective total sum-of-squares.

The constructed PLS biplot in this chapter assumes a linear relationship between the predictors and the response variables. Occasionally, the relationship between these two sets of variables may follow a non-linear function. Thus, Chapter 6 investigates the employment of the PLS biplot to explore the non-linear relationship between these two sets of variables.

CHAPTER 6

PARTIAL LEAST SQUARES BIPLOTS FOR GENERALIZED LINEAR MODELS

6.1 Introduction

Usually, in regression analysis, the modelling of one response variable $\mathbf{y}$ ($N \times 1$) by means of one or more predictor variables $\mathbf{X}$ ($N \times P$) is performed by solving the equation $\mathbf{y} = \mathbf{Xb}$, where $\mathbf{b}$ ($P \times 1$) is the unknown coefficient vector that needs to be estimated. This equation, $\mathbf{y} = \mathbf{Xb}$, is often referred to as a linear model, because it assumes that there is some linear function relating $\mathbf{y}$ to $\mathbf{X}$. It furthers assume that $\mathbf{y}$ follows a normal distribution with mean μ and constant variance σ^2. To be precise, the linear model has a general form

$$\mathrm{E}(\mathbf{y}) = \boldsymbol{\mu} = \mathbf{Xb} \tag{6.1}$$

where $\mathrm{E}(\mathbf{y})$ is the expected value of $\mathbf{y}$. However, sometimes $\mathbf{y}$ may have a distribution other than the normal distribution and the relationship between $\mathbf{y}$ and $\mathbf{X}$ may follow some non-linear function. Of course, the linear model is not a suitable tool for modelling $\mathbf{y}$ in this situation. This does not necessarily mean that the modelling of $\mathbf{y}$ is impossible, but $\mathbf{y}$ should rather be modelled using a non-linear model. An example of such a model is the Generalized Linear Model (GLM). In GLMs, $\mathbf{y}$ is assumed to be from any distribution in the exponential family. A distribution is said to be from the exponential family if its probability function can be written in the form

$$f(\mathbf{y};\theta) = \mathrm{e}^{\mathrm{a(y)b(\theta)+c(\theta)+d(y)}} \tag{6.2}$$

for a single random variable Y and a single parameter θ, where $\mathrm{a}(\cdot)$, $\mathrm{b}(\cdot)$, $\mathrm{c}(\cdot)$ and $\mathrm{d}(\cdot)$ are known functions, varying from one exponential family to another (Dobson, 2002). For a GLM, $\mathrm{a(y)} = \mathrm{y}$, making (6.2) to be written as

$$f(\mathbf{y};\theta) = \mathrm{e}^{\mathrm{yb(\theta)+c(\theta)+d(y)}}. \tag{6.3}$$

Dobson (2002) referred to (6.3) as the standard or canonical form of the exponential family. The most well-known distributions from this family are the Normal, Poisson, Binomial and Negative Binomial distributions. GLMs further assume a 'link-linear' relationship. More precisely, GLMs transform the mean of $\mathbf{y}$ non-linearly and then model the transformed mean as a linear function of $\mathbf{X}$. That is,

$$g(\boldsymbol{\mu}) = \mathbf{Xb} \tag{6.4}$$

where $\boldsymbol{\mu} = \mathrm{E}(\mathbf{y})$ and g is a monotone and differentiable link function. An example of $g(\cdot)$ is the log-link function, i.e., $\log(\boldsymbol{\mu}) = \mathbf{Xb}$, for a Poisson distributed $\mathbf{y}$. Comparing (6.4) to (6.1),

GLMs can be viewed as an extension of linear models, in that they relate $\boldsymbol{\mu}$ to $\mathbf{Xb}$ via a link function. Since $g(\cdot)$ transforms $\mathrm{E}(\mathbf{y})$ and not $\mathbf{y}$, from (6.4), $\mathrm{E}(\mathbf{y}) = \boldsymbol{\mu} = g^{-1}(\mathbf{Xb})$, where g^{-1} is the inverse of the link function $g(\cdot)$.

In this chapter, a brief overview is provided of GLMs before the extension of PLS to the GLM framework is discussed. Furthermore, the PLS biplot is proposed as a visual tool for displaying the Partial Least Squares-Generalized Linear Model (PLS-GLM) of a data set.

6.2 Generalized Linear Models

Consider $\mathbf{y}$ having N observations that are independently distributed with mean $\boldsymbol{\mu}$. The GLM for this $\mathbf{y}$ is given by (6.4), where the vector $\mathbf{b}$ needs to be estimated. To estimate this vector, the maximum likelihood estimation technique is used (McCullagh & Nelder, 1989). The maximum likelihood estimation is implemented with an Iterative Weighted Least Squares (IWLS) procedure. Suppose $\mathbf{y}$ is Poisson distributed. Let $y_1, y_2, \cdots, y_N$ denote N independent observations of $\mathbf{y}$. In the IWLS procedure, $\mathbf{y}$ is not used as the response variable, but rather a linearized form of the link function applied to $\mathbf{y}$. Denote the linearized form as $\mathbf{z}$. For a Poisson distributed $\mathbf{y}$, the link function $g(\cdot)$ is a log-link, i.e., $\log(\boldsymbol{\mu}) = \mathbf{Xb}$. Let $\eta_i = \log(\mu_i)$, where $\log(\mu_i) = \mathbf{x}_i^T\mathbf{b}$. The linearized form ($\mathbf{z}$) is calculated as

$$z_i = \eta_i + (y_i - \mu_i)\frac{\partial \eta_i}{\partial \mu_i}. \tag{6.5}$$

With $\eta_i = \log(\mu_i)$,

$$\frac{\partial \eta_i}{\partial \mu_i} = \frac{1}{\mu_i}(1) = (\mu_i)^{-1}. \tag{6.6}$$

Replacing $\frac{\partial \eta_i}{\partial \mu_i}$ in (6.5) by (6.6) yields

$$z_i = \eta_i + (y_i - \mu_i)(\mu_i)^{-1}. \tag{6.7}$$

As $\log(\mu_i) = \mathbf{x}_i^T\mathbf{b}$, it follows that $\mu_i = e^{\mathbf{x}_i^T\mathbf{b}}$. Then

$$\eta_i = \log(\mu_i) = \log\left(e^{\mathbf{x}_i^T\mathbf{b}}\right) = \mathbf{x}_i^T\mathbf{b}. \tag{6.8}$$

and

$$z_i = \mathbf{x}_i^T\mathbf{b} + \left(y_i - e^{\mathbf{x}_i^T\mathbf{b}}\right)\left(e^{\mathbf{x}_i^T\mathbf{b}}\right)^{-1}. \tag{6.9}$$

Furthermore, the weight v_i in the IWLS procedure is defined as a function of μ_i. More precisely,

$$v_i = 1/\left[\mathrm{var}(y_i)\left(\frac{\partial \eta_i}{\partial \mu_i}\right)^2\right]. \tag{6.10}$$

Since $\mathrm{var}(y_i) = \mu_i$ for a Poisson distributed $\mathbf{y}$, from (6.6), $v_i = 1/\left[\mu_i\left(\frac{1}{\mu_i}\right)^2\right] = 1/\left[\frac{1}{\mu_i}\right]$ which simplifies to

$$v_i = \mu_i = e^{\mathbf{x}_i^T \mathbf{b}}. \quad (6.11)$$

The estimated coefficient is then obtained by regressing z_i on $\mathbf{X}$ using the weights v_i. That is,

$$\hat{\mathbf{b}} = (\mathbf{X}^T\mathbf{V}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{V}\mathbf{z} \quad (6.12)$$

where $\mathbf{V}$ $(N \times N)$ is a diagonal matrix of weights with elements v_i and $\mathbf{z}$ $(N \times 1)$ a vector with elements z_i. Seeing that z_i (6.9) and v_i (6.11) depend on the unknown coefficients in $\mathbf{b}$, the Weighted Least Squares (WLS) procedure is iterated until convergence (Dobson, 2002). This algorithm can be summarized as follows.

Algorithm 6.1:

(1) Start with an initial estimate for $\mathbf{b}$. Let $\mathbf{b}^{(0)}$ denote this estimate.

(2) At each iteration l, compute z_i and v_i as

$$z_i^{(l)} = \mathbf{x}_i^T\mathbf{b}^{(l)} + \left(y_i - e^{\mathbf{x}_i^T\mathbf{b}^{(l)}}\right)\left(e^{\mathbf{x}_i^T\mathbf{b}^{(l)}}\right)^{-1}$$

$$v_i^{(l)} = e^{\mathbf{x}_i^T\mathbf{b}^{(l)}}.$$

(3) Update $\mathbf{b}$, for better approximations for z_i and v_i, as

$$\mathbf{b}^{(l+1)} = \left(\mathbf{X}^T\mathbf{V}^{(l)}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{V}^{(l)}\mathbf{z}^{(l)}$$

where $\mathbf{V}^{(l)} = \text{diag}\{v_i^{(l)}\}$ and $\mathbf{z}^{(l)}$ is a vector of element $z_i^{(l)}$.

(4) Repeat steps (2) and (3) until convergence of $\mathbf{b}$ is reached.

The general linearized form z_i (6.5) and the weight v_i (6.10) are defined for any distribution of $\mathbf{y}$ in the exponential family, besides the Poisson. For example, suppose $\mathbf{y}$ is Binomial distributed with N samples and a probability of success π. Here, the link function $g(\cdot)$ can be a logit-link. That is,

$$\eta_i = \text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right) = \mathbf{x}_i^T\mathbf{b}. \quad (6.13)$$

Since $E(y_i) = \mu_i = N_i\pi_i$ for a Binomial distributed $\mathbf{y}$, $\pi_i = \frac{\mu_i}{N_i}$. Substitute $\pi_i = \frac{\mu_i}{N_i}$ into (6.13) yields

$$\eta_i = \log\left(\frac{\mu_i}{N_i-\mu_i}\right) = \mathbf{x}_i^T\mathbf{b}$$

which can be written as

$$\eta_i = \log(\mu_i) - \log(N_i - \mu_i) = \mathbf{x}_i^T\mathbf{b}. \quad (6.14)$$

From (6.14),

$$\begin{aligned}\frac{\partial\eta_i}{\partial\mu_i} &= \frac{1}{\mu_i}(1) - \frac{1}{N_i-\mu_i}(-1)\\ &= \frac{N_i-\mu_i+\mu_i}{\mu_i(N_i-\mu_i)}\\ &= \frac{N_i}{\mu_i(N_i-\mu_i)}.\end{aligned}$$

With $\mu_i = N_i\pi_i$,

$$\begin{aligned}\frac{\partial \eta_i}{\partial \mu_i} &= \frac{N_i}{N_i\pi_i(N_i - N_i\pi_i)} \\ &= \frac{1}{\pi_i(N_i - N_i\pi_i)} \\ &= \frac{1}{N_i\pi_i(1-\pi_i)}.\end{aligned} \tag{6.15}$$

Replacing $\frac{\partial \eta_i}{\partial \mu_i}$ and μ_i in z_i (6.5) by (6.15) and $\mu_i = N_i\pi_i$ respectively yields

$$\begin{aligned}z_i &= \eta_i + (y_i - N_i\pi_i)\left(\frac{1}{N_i\pi_i(1-\pi_i)}\right) \\ &= \eta_i + \frac{y_i - N_i\pi_i}{N_i\pi_i(1-\pi_i)}.\end{aligned} \tag{6.16}$$

Furthermore, the weight v_i (6.10) turns out to be

$$v_i = \frac{1}{\text{var}(y_i)\left(\frac{1}{N_i\pi_i(1-\pi_i)}\right)^2}. \tag{6.17}$$

Since $\text{var}(y_i) = N_i\pi_i(1-\pi_i)$ for a Binomial distributed $\mathbf{y}$, (6.17) becomes

$$\begin{aligned}v_i &= \frac{1}{N_i\pi_i(1-\pi_i)\left(\frac{1}{N_i\pi_i(1-\pi_i)}\right)^2} \\ &= 1 \times \frac{(N_i\pi_i(1-\pi_i))^2}{N_i\pi_i(1-\pi_i)}\end{aligned}$$

which simplifies to

$$v_i = N_i\pi_i(1-\pi_i). \tag{6.18}$$

As $\log\left(\frac{\pi_i}{1-\pi_i}\right) = \mathbf{x}_i^T\mathbf{b}$, it follows that $\pi_i = \frac{e^{\mathbf{x}_i^T\mathbf{b}}}{1+e^{\mathbf{x}_i^T\mathbf{b}}}$.

Now, consider a scenario where there are more than one response variable, i.e., $M > 1$ and $\mathbf{Y}$ $(N \times M)$. Obviously (6.4) cannot be used to model this response matrix $\mathbf{Y}$, but rather, a multivariate version of (6.4). With $\mathbf{Y} = (\mathbf{y}_1 \;\; \mathbf{y}_2 \;\; \cdots \;\; \mathbf{y}_M)$, difficulties can be encountered when estimating the coefficients (6.12). To explain, consider the weight matrix $\mathbf{V}$ in the IWLS. In this procedure, for one Y-variable, $\mathbf{V}$ is defined as a $(N \times N)$ diagonal matrix of elements v_i, where $v_i = 1/\left[\text{var}(y_i)\left(\frac{\partial \eta_i}{\partial \mu_i}\right)^2\right]$ and $\mu_i = e^{\mathbf{x}_i^T\mathbf{b}}$ (6.10) and (6.11). Here, $\text{var}(y_i)$ is a scalar. On the other hand, for $M > 1$ Y-variables, $\text{var}(y_i)$ would become $\text{var}(\mathbf{Y})$ and $\text{var}(\mathbf{Y})$ is not a scalar, but rather, a $(M \times 1)$ vector. This, together with taking the derivative of a vector, poses a (computational) problem for v_i in (6.10). As a result, a multivariate version of v_i is needed, for estimating the coefficients in (6.12). To the knowledge of the author, no general framework tackling this problem has been constructed yet. Although several texts, papers and publications have been written on GLMs and their extensions to other multivariate techniques, such as DA and PLS, none of them have given a (proper) general framework for multivariate $(M > 1)$ GLMs. The recently developed PLS-GLM R package **plsRglm**, by Bertrand *et al*. (2013), does PLS and PLS-GLM regression, works with complete and incomplete data, and uses several

criteria for choosing the number of components among others, but it is still limited to $M = 1$ Y-variable. Multivariate analysis texts on GLMs, like Fahrmeir and Tutz (2001) and Lattin *et al*. (2003), also only consider $M = 1$ Y-variable. Lattin *et al*. (2003) concentrated on a binary response variable as well as a response variable having more than two categories in their logit and probit GLM applications, see pages 474-519. They called their frameworks the *logit and probit choice models*. They also compared the difference between these two models, using a simple example. Even so, they still focused on a $M = 1$ Y-variable. Fahrmeir and Tutz (2001) further focused on $\mathbf{y}$ having more than two categories and termed such $\mathbf{y}$ the *multivariate response variable* in their GLM applications. The resulting GLM cannot be called the multivariate GLM, but rather, a *multicategorical response GLM*. In addition, papers such as Bastien *et al*. (2005), Ding & Gentleman (2004), Marx (1996), Meyer *et al*. (2010) and Park *et al*. (2002) only deal with a single response variable in their PLS-GLM frameworks. Although Gersende & Lambert-Lacroix (2004) focused on binary responses, they also still use $M = 1$ Y-variable in their discussed PLS-GLM framework. Seeing as more work is still needed on the development of a general framework for multivariate ($M > 1$) GLMs, *alternatively*, *one can take each of the response variables in* $\mathbf{Y}$ *and model it using* (6.4), thereby having M univariate GLMs. However, the results obtained using this method cannot be taken as the (multivariate) GLM analysis for $\mathbf{Y}$.

6.3 Partial Least Squares for Generalized Linear Models

For a response variable $\mathbf{y}\,(N \times 1)$, the PLSR model is expressed as $\mathbf{y}_0 = \mathbf{X}_0\hat{\mathbf{b}}_{PLS}$, where $\hat{\mathbf{b}}_{PLS}\,(P \times 1) = \mathbf{Rq}$ from (3.17). Although $\mathbf{y}_0 = \mathbf{X}_0\hat{\mathbf{b}}_{PLS}$ looks exactly like the ordinary GLM (6.4), performing PLS-GLM has the dimension reduction advantage as well as finding a (smaller) set of orthogonal latent variables $\mathbf{T}_*$ for the non-linear modelling of $\mathbf{y}$. In this chapter, the orthogonal latent variables are denoted by $\mathbf{T}_*$, to differentiate them from the orthogonal latent variables ($\mathbf{T}$) obtained under the standard PLS (Chapter 3). The PLS model can be extended to a GLM such that

$$g(\boldsymbol{\mu}) = \mathbf{X}_0\hat{\mathbf{b}}_{PLS-GLM} \tag{6.19}$$

where $\hat{\mathbf{b}}_{PLS-GLM} = \mathbf{R}_*\mathbf{q}_*$ is the PLS-GLM coefficient vector and both $\mathbf{R}_*$ and $\mathbf{q}_*\,(A \times 1)$ are computed from the PLS-GLM. With $\mathbf{z}$, a linearized form of the link function applied to $\mathbf{y}$, being used as the response variable in the IWLS procedure for the GLM, instead of $\mathbf{y}$ (see

Section 6.2), the objective function of PLS-GLM can be interpreted as follows. For $a = 1, 2, \ldots, A$,

$$\text{maximize } \operatorname{cov}\left(\mathbf{w}_{a*}^{T}\mathbf{X}^{T}\mathbf{z}\mathbf{z}^{T}\mathbf{X}\mathbf{w}_{a*}\right) = \operatorname{cov}\left(\mathbf{t}_{a*}^{T}\mathbf{z}\mathbf{z}^{T}\mathbf{t}_{a*}\right)$$
$$\text{subject to } \mathbf{t}_{a*}^{T}\mathbf{V}\mathbf{t}_{a*} = \mathbf{w}_{a*}^{T}\mathbf{X}^{T}\mathbf{V}\mathbf{X}\mathbf{w}_{a*} = 1 \text{ and } \mathbf{t}_{a*}^{T}\mathbf{t}_{b*} = \mathbf{w}_{a*}^{T}\mathbf{X}^{T}\mathbf{V}\mathbf{X}\mathbf{w}_{b*} = 0,$$

where $\mathbf{t}_{a*} = \mathbf{X}\mathbf{w}_{a*}$, $b = 1, 2, \cdots, A-1$ and $b \neq a$. This function is solved using the PLS-GLM algorithm. Unlike the PLS objective function, see Section 3.5, the constraint imposed here involves the PLS scores $\mathbf{t}_{a*}$ and the IWLS weight matrix $\mathbf{V}$, i.e., $\left\|\mathbf{t}_{a*}^{T}\mathbf{V}\mathbf{t}_{a*}\right\| = 1$ and not $\left\|\mathbf{t}_{a*}^{T}\mathbf{t}_{a*}\right\| = 1$. Park *et al*. (2002) proposed an algorithm for a PLS-GLM. This algorithm is summarized below. Let $\boldsymbol{\eta} = \log(\boldsymbol{\mu}) = \mathbf{X}\mathbf{b}$.

Algorithm 6.2:

(1) Start with an initial estimate for $\mathbf{b}$. Let $\mathbf{b}^{(0)}$ denote this estimate. Then initialize

$$l = 0$$
$$\mathbf{V}^{(l)} = \operatorname{diag}\left\{v_i^{(l)}\right\}, \text{ for } i = 1, 2, \cdots, N \text{ and } v_i \text{ defined in (6.10)}$$
$$\mathbf{X}_0 = \mathbf{X} - \frac{1}{N}\mathbf{1}\mathbf{1}^{T}\mathbf{V}^{(l)}\mathbf{X}$$
$$\mathbf{z}_0^{(l)} = \boldsymbol{\eta}^{(l)} - \frac{1}{N}\mathbf{1}\mathbf{1}^{T}\mathbf{V}^{(l)}\boldsymbol{\eta}^{(l)} = \mathbf{X}_0\mathbf{b}^{(l)} - \frac{1}{N}\mathbf{1}\mathbf{1}^{T}\mathbf{V}^{(l)}\mathbf{X}_0\mathbf{b}^{(l)}.$$

(2) For $a = 1, 2, \cdots, A$, compute the PLS parameters (see Algorithm 3.1) as

a. $\mathbf{w}_a = \mathbf{X}_{a-1}^{T}\mathbf{V}^{(l)}\,\mathbf{z}_{a-1}^{(l)} \,/\, \left\|\mathbf{X}_{a-1}^{T}\mathbf{V}^{(l)}\,\mathbf{z}_{a-1}^{(l)}\right\|$
b. $\mathbf{p}_a = \mathbf{X}_{a-1}^{T}\mathbf{V}^{(l)}\,\mathbf{X}_{a-1}\mathbf{w}_a$
c. $q_a = \mathbf{z}_{a-1}^{T\,(l)}\,\mathbf{V}^{(l)}\,\mathbf{X}_{a-1}\mathbf{w}_a$
d. $\mathbf{X}_a = \mathbf{X}_{a-1} - \mathbf{X}_{a-1}\mathbf{w}_a\mathbf{p}_a^{T}$
e. $\mathbf{z}_a^{(l)} = \mathbf{z}_{a-1}^{(l)} - q_a\mathbf{X}_{a-1}\mathbf{w}_a$.
f. Store $\mathbf{w}_a$ $(P \times 1)$, $\mathbf{p}_a$ $(P \times 1)$ and q_a into the successive columns of $\mathbf{W}_*$, $\mathbf{P}_*$ and $\mathbf{q}_*^{T}$.
g. Compute $\mathbf{T}_* = \mathbf{X}_0\mathbf{R}_*$, where $\mathbf{R}_* = \mathbf{W}_*(\mathbf{P}_*^{T}\mathbf{W}_*)^{-1}$.
h. Get $\hat{\mathbf{b}}_{\mathrm{PLS-GLM}} = \mathbf{R}_*\mathbf{q}_*$.

(3) Set $\mathbf{b}^{(l+1)} = \hat{\mathbf{b}}_{\mathrm{PLS-GLM}}$ and update $\boldsymbol{\eta}$, weight matrix $\mathbf{V}$ and $\mathbf{z}_0$ as

$$\boldsymbol{\eta}^{(l+1)} = \mathbf{X}_0\mathbf{b}^{(l+1)} + \frac{1}{N}\mathbf{1}\mathbf{1}^{T}\mathbf{V}^{(l)}\boldsymbol{\eta}^{(l)}$$
$$\mathbf{V}^{(l+1)} = \operatorname{diag}\left\{v_i^{(l+1)}\right\}, \text{ for } v_i \text{ defined in (6.10)}$$
$$\mathbf{z}_0^{(l+1)} = \mathbf{X}_0\mathbf{b}^{(l+1)} + \frac{\partial\boldsymbol{\eta}}{\partial\boldsymbol{\mu}}(\mathbf{y} - \boldsymbol{\mu}).$$

(4) Check whether the change in successive estimates is sufficiently small. If not, return to step (2), with $l = l + 1$.

(5) Once convergence is reached, select the final number of components A, call this A^*, and take A^* columns of $\mathbf{T}_*$, $\mathbf{R}_*$ and $\mathbf{q}_*^{T}$ to fit (6.19).

This algorithm can also be written using the SIMPLS algorithm (see Algorithm 3.4). The only difference is how the PLS parameters in step (2) are computed. Here, the objective function can be expressed as

$$\text{maximize } \operatorname{cov}(\mathbf{r}_*^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{z}\mathbf{z}^{\mathrm{T}}\mathbf{X}\mathbf{r}_*) = \operatorname{cov}(\mathbf{t}_*^{\mathrm{T}}\mathbf{z}\mathbf{z}^{\mathrm{T}}\mathbf{t}_*)$$
$$\text{subject to } \mathbf{t}_{\mathrm{a}*}^{\mathrm{T}}\mathbf{V}\mathbf{t}_{\mathrm{a}*} = \mathbf{r}_{\mathrm{a}*}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{V}\mathbf{X}\mathbf{r}_{\mathrm{a}*} = 1 \text{ and } \mathbf{t}_{\mathrm{a}*}^{\mathrm{T}}\mathbf{t}_{\mathrm{b}*} = \mathbf{r}_{\mathrm{a}*}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{V}\mathbf{X}\mathbf{r}_{\mathrm{b}*} = 0,$$

where $\mathbf{t}_* = \mathbf{X}\mathbf{r}_*$.

Algorithm 6.3:

(1) Start with an initial estimate for $\mathbf{b}$. Let $\mathbf{b}^{(0)}$ denote this estimate. Then initialize

$$l = 0$$
$$\mathbf{V}^{(l)} = \operatorname{diag}\left\{\mathrm{v}_i^{(l)}\right\}, \text{ for } \mathrm{i} = 1, 2, \cdots, \mathrm{N} \text{ and } \mathrm{v_i} \text{ defined in (6.10)}$$
$$\mathbf{X}_0 = \mathbf{X} - \frac{1}{\mathrm{N}}\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{V}^{(l)}\mathbf{X}$$
$$\mathbf{z}_0^{(l)} = \boldsymbol{\eta}^{(l)} - \frac{1}{\mathrm{N}}\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{V}^{(l)}\boldsymbol{\eta}^{(l)} = \mathbf{X}_0\mathbf{b}^{(l)} - \frac{1}{\mathrm{N}}\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{V}^{(l)}\mathbf{X}_0\mathbf{b}^{(l)}.$$

(2) For $\mathrm{a} = 1, 2, \cdots, \mathrm{A}$, compute the PLS parameters (see Algorithm 3.4) as

a. $\mathbf{S}_{\mathrm{a}} = \mathbf{D}\boldsymbol{\Lambda}\mathbf{F}^{\mathrm{T}}$, by the SVD, for $\mathbf{S}_1 = \mathbf{X}_0^{\mathrm{T}}\mathbf{V}^{(l)}\mathbf{z}_0^{(l)}$
b. $\mathbf{r}_{\mathrm{a}} = \mathbf{D}[,\ 1]$
c. $\mathbf{t}_{\mathrm{a}} = \mathbf{X}_0\mathbf{r}_{\mathrm{a}}/\|\mathbf{X}_0\mathbf{r}_{\mathrm{a}}\|$
d. $\mathbf{p}_{\mathrm{a}} = \mathbf{X}_0^{\mathrm{T}}\mathbf{V}^{(l)}\mathbf{t}_{\mathrm{a}}$
e. $\mathrm{q}_{\mathrm{a}} = \mathbf{z}_{\mathrm{a}-1}^{\mathrm{T}\ (l)}\mathbf{V}^{(l)}\mathbf{t}_{\mathrm{a}}$
f. $\mathbf{S}_{\mathrm{a}+1} = (\mathbf{I}_{\mathrm{P}} - \mathbf{p}_{\mathrm{a}}(\mathbf{p}_{\mathrm{a}}^{\mathrm{T}}\mathbf{p}_{\mathrm{a}})^{-1}\mathbf{p}_{\mathrm{a}}^{\mathrm{T}})\mathbf{S}_{\mathrm{a}}$.
g. Store $\mathbf{r}_{\mathrm{a}}$ $(\mathrm{P} \times 1)$, $\mathbf{t}_{\mathrm{a}}$ $(\mathrm{N} \times 1)$, $\mathbf{p}_{\mathrm{a}}$ $(\mathrm{P} \times 1)$ and q_{a} into the successive columns of $\mathbf{R}_*$, $\mathbf{T}_*$, $\mathbf{P}_*$ and $\mathbf{q}_*^{\mathrm{T}}$.
h. Get $\hat{\mathbf{b}}_{\mathrm{PLS-GLM}} = \mathbf{R}_*\mathbf{q}_*$.

(3) Set $\mathbf{b}^{(l+1)} = \hat{\mathbf{b}}_{\mathrm{PLS-GLM}}$ and update $\boldsymbol{\eta}$, weight matrix $\mathbf{V}$ and $\mathbf{z}_0$ as

$$\boldsymbol{\eta}^{(l+1)} = \mathbf{X}_0\mathbf{b}^{(l+1)} + \frac{1}{\mathrm{N}}\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{V}^{(l)}\boldsymbol{\eta}^{(l)}$$
$$\mathbf{V}^{(l+1)} = \operatorname{diag}\left\{\mathrm{v}_i^{(l+1)}\right\}, \text{ for } \mathrm{v_i} \text{ defined in (6.10)}$$
$$\mathbf{z}_0^{(l+1)} = \mathbf{X}_0\mathbf{b}^{(l+1)} + \frac{\partial\boldsymbol{\eta}}{\partial\boldsymbol{\mu}}(\mathbf{y} - \boldsymbol{\mu}).$$

(4) Check whether the change in successive estimates is sufficiently small. If not, return to step (2), with $l = l + 1$.

(5) Once convergence is reached, select the final number of components A, call this A*, and take A* columns of $\mathbf{T}_*$, $\mathbf{R}_*$ and $\mathbf{q}_*^{\mathrm{T}}$ to fit (6.19).

6.4 PLS biplot for Generalized Linear Models

The PLS biplot, see Chapter 5, can be used as a graphical tool for displaying the PLS-GLM of a data set. For a pair of predictor and response samples $\mathbf{x}$ and $\mathbf{y}$, both samples will be

interpolated into the biplot space using the equation $\mathbf{t}_*^{\mathrm{T}} = (\mathbf{x} - \bar{\mathbf{x}})^{\mathrm{T}}\mathbf{R}_*$. To trace the prediction biplot axes for the k^{th} original predictor variable, values of $\mu_{0_X} \in (-\infty, \infty)$ are substituted into the expression $\frac{\mu_{0X}}{\mathbf{e}_k^{\mathrm{T}}\mathbf{P}_*\mathbf{P}_*^{\mathrm{T}}\mathbf{e}_k}\mathbf{P}_*^{\mathrm{T}}\mathbf{e}_k$, where $\mathbf{e}_k$ is the unit vector with zeros except for a one in the k^{th} position. Here, with $\mathbf{y}\,(N \times 1)$, $g(\boldsymbol{\mu}) = \mathbf{T}_*\mathbf{q}_*$. Any point predicting μ_{0_Y} for the response variable will have $\mu_{0_Y} = \mathbf{t}_*^{\mathrm{T}}\mathbf{q}_*$, with the biplot axis of the form $\phi\mathbf{q}_*$. If the value μ_{0_Y} is predicted, $\mathbf{t}_{*_{\mu_{0_Y}}} = \phi\mathbf{q}_*$ and $\mu_{0_Y} = \phi\mathbf{q}_*^{\mathrm{T}}\mathbf{q}_*$ so that the marker μ_{0_Y} on the response prediction biplot axis is given by $\frac{\mu_{0_Y}}{\mathbf{q}_*^{\mathrm{T}}\mathbf{q}_*}\mathbf{q}_*$. The values μ_{0_X} and μ_{0_Y} are in terms of the centred samples. Hence, calibration markers are fitted, where $\mu_X = \mu_{0_X} + \bar{x}_k$ and $\mu_Y = \mu_{0_Y} + \bar{y}$ are sensible scale marker values. Furthermore, the estimated PLS-GLM coefficient vector $\hat{\mathbf{b}}_{\mathrm{PLS-GLM}} = \mathbf{R}_*\mathbf{q}_*$ is obtained by projecting $\mathbf{e}_i^{\mathrm{T}}\mathbf{R}_*$ onto the prediction axis defined by $\mathbf{q}_*^{\mathrm{T}}$.

6.5 Example

The following example is an illustration of the PLS biplot to a Poisson GLM, using the possum diversity data from Lindenmayer *et al*. (1991). This data shows a study done on the diversity of arboreal marsupials (possum) in the montane ash forest in Australia. Different species of possum were observed on one hundred and one (different) sites. For each site, nine variable measures (Diversity, Shrubs, Stumps, Stags, Bark, Habitat, BAcacia, Eucalyptus and Aspect) were recorded. With the last two measures being categorical variables, they were further splited up, using dummy variables, to form additional variables. As a result, five additional variables were created. Thus, the variable measures are Diversity, Shrubs, Stumps, Stags, Bark, Habitat, BAcacia, E.regnans, E.delegatensis, E.nitens, NW-NE, NW-SE, SE-SW and SW-NW. The aim of this study is to model the relationship between Diversity and the other variables. The one hundred and one sites are assigned as the samples. Diversity is assigned as the response variable, while the remaining variables are assigned as the predictors. Hence, $\mathbf{X}\,(151 \times 13)$ and $\mathbf{y}\,(151 \times 1)$ in this example. This data can be obtained from the **robustbase** package in R, downloaded freely from CRAN, http://cran.r-project.org/. Since the Diversity measure is a count variable, a Poisson PLS-GLM is fitted. Here, $\boldsymbol{\eta} = \log(\boldsymbol{\mu}) = \mathbf{Xb}$ and $\boldsymbol{\mu} = e^{\mathbf{Xb}}$. Using Algorithm 6.2, the resulting PLS biplot is shown in Figure 6.1, *along with a representation of the variance of each variable*. In this display, the samples of the data are given by the red points, while the purple points are for the (PLS-GLM) coefficient points. The predictor variables of the data are represented by the blue axes, while the response variable is represented by the black axis.

Comparing the length of the thicker arrows (vectors) to each other, Shrubs, Stumps, NW-NE and E.delegatensis can be said to have a large standard deviations.

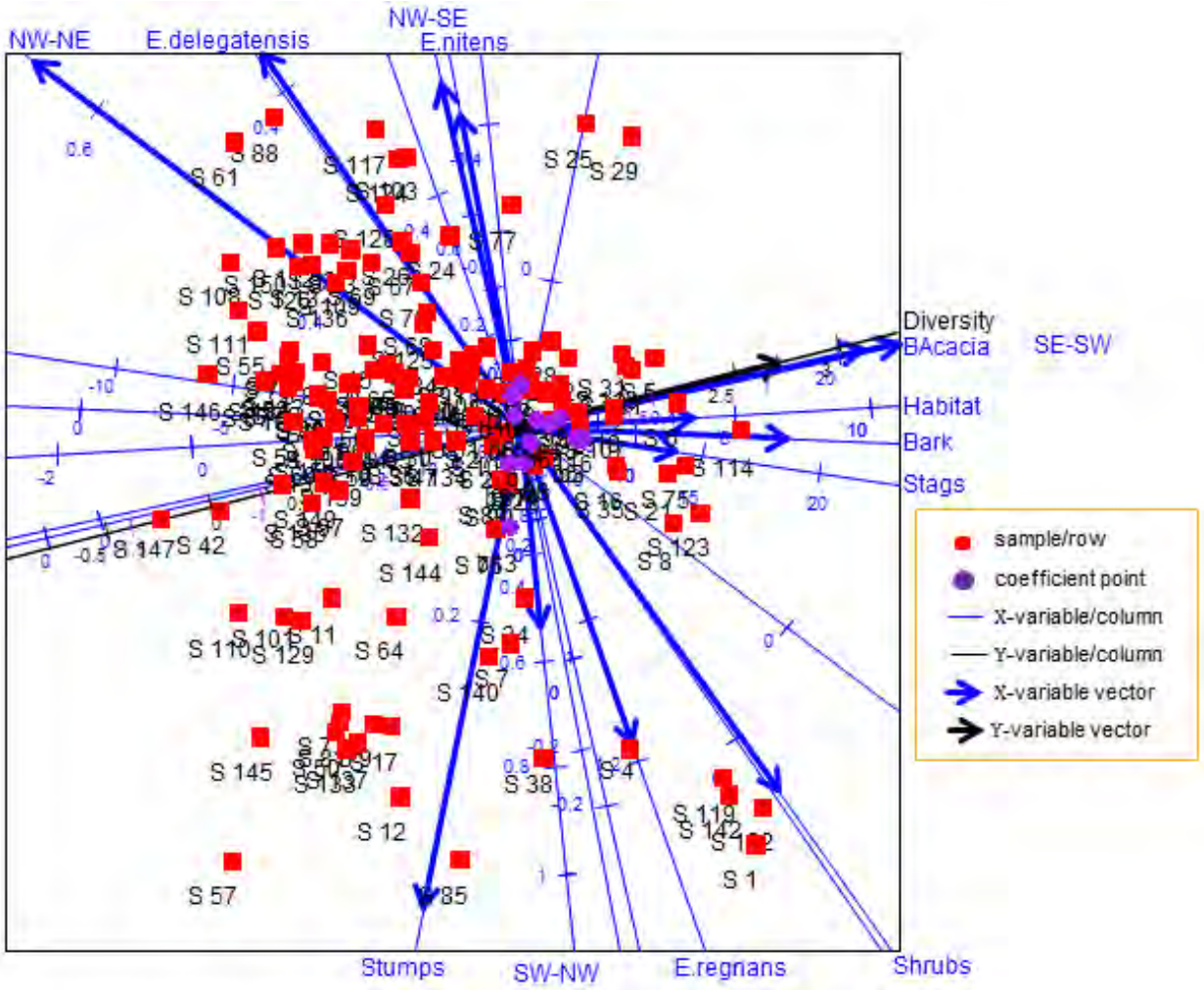

Figure 6.1 The PLS biplot of a Poisson PLS-GLM of the possum diversity data.

In addition, several relationships can be deduced from this biplot. To start with, the relation between Diversity and predictors BAcacia, SE-SW, Habitat, Bark and Stags can be seen. Also, the relation within predictors E.regnans, E.nitens, NW-SE and SW-NW; and within predictors Shrubs and E.delegatensis can be seen. Another display of Figure 6.1, where the sample point names have been excluded, is shown in Figure 6.3.

Orthogonally projecting each of the coefficient points bi, for $i = 1, 2, \cdots, 13$, in Figure 6.1 onto the Diversity axis yields the coefficient values. As discussed in Subsection 5.2.3 and Section 6.4, *the purple markers on the Diversity axis* are used to read off these values. A zoomed-in display of the coefficient points, shown below in Figure 6.2, can be used for easier orthogonal

projections. For example, coefficient points b10 and b2 projected orthogonally onto the Diversity axis yields 0.01 and −0.07 respectively, as shown in Figure 6.2 below.

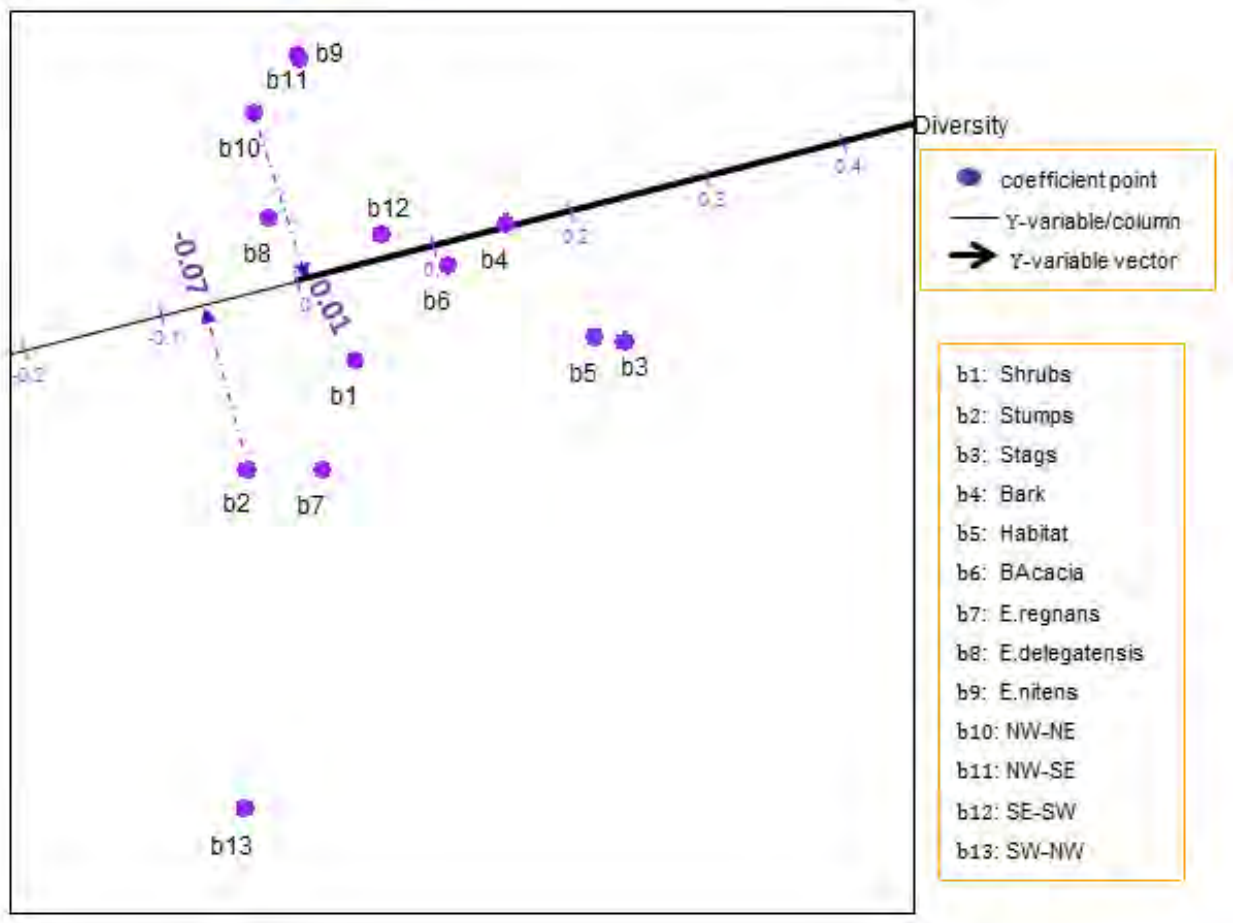

Figure 6.2 A zoomed-in display of the coefficient points in the PLS biplot of a Poisson PLS-GLM of the possum diversity data.

With a single response variable ($M = 1$), such dotted lines in Figure 6.2 can easily be used to read off the values for the regression coefficients. In most PLS biplots, where there are $M > 1$ response variables, dotted lines connecting all the coefficient points to the $M > 1$ response axes would be confusing and the plot would become too messy. The obtained coefficient values of the possum diversity data are shown in Table 6.1, under the PLS-GLM column. From Table 6.1, variable Stags can be said to have a high effect on Diversity, followed by Habitat, Bark, SW-NW and BAcacia. However, the other variables can be seen to have a low to no effect, with NW-NE having the lowest effect.

Table 6.1 The predicted coefficient values.

		PLS-GLM	SIMPLS fitted PLS-GLM
b1:	Shrubs	0.026	0.026
b2:	Stumps	-0.067	-0.067
b3:	Stags	0.214	0.214
b4:	Bark	0.153	0.153
b5:	Habitat	0.195	0.195
b6:	BAcacia	0.106	0.106
b7:	E.regnans	-0.016	-0.016
b8:	E.delegatensis	-0.009	-0.009
b9:	E.nitens	0.039	0.039
b10:	NW-NE	-0.001	-0.001
b11:	NW-SE	0.039	0.039
b12:	SE-SW	0.066	0.066
b13:	SW-NW	-0.128	-0.128

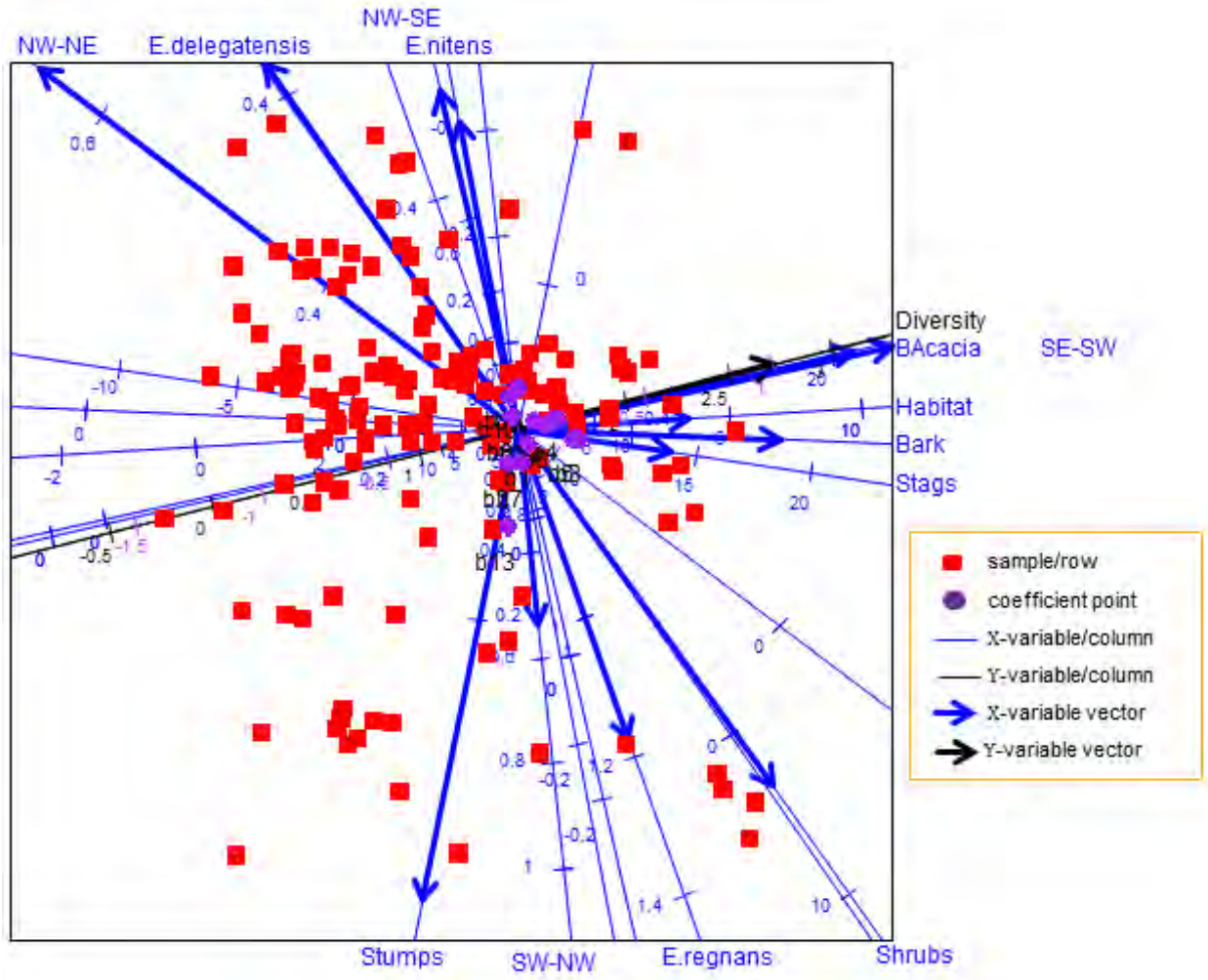

Figure 6.3 The PLS biplot of a Poisson PLS-GLM of the possum diversity data, without sample names.

Furthermore, for comparison reasons, a separate PLS-GLM using the SIMPLS algorithm discussed in Algorithm 6.3 was fitted and the resulting PLS biplot is shown in Figure 6.4.

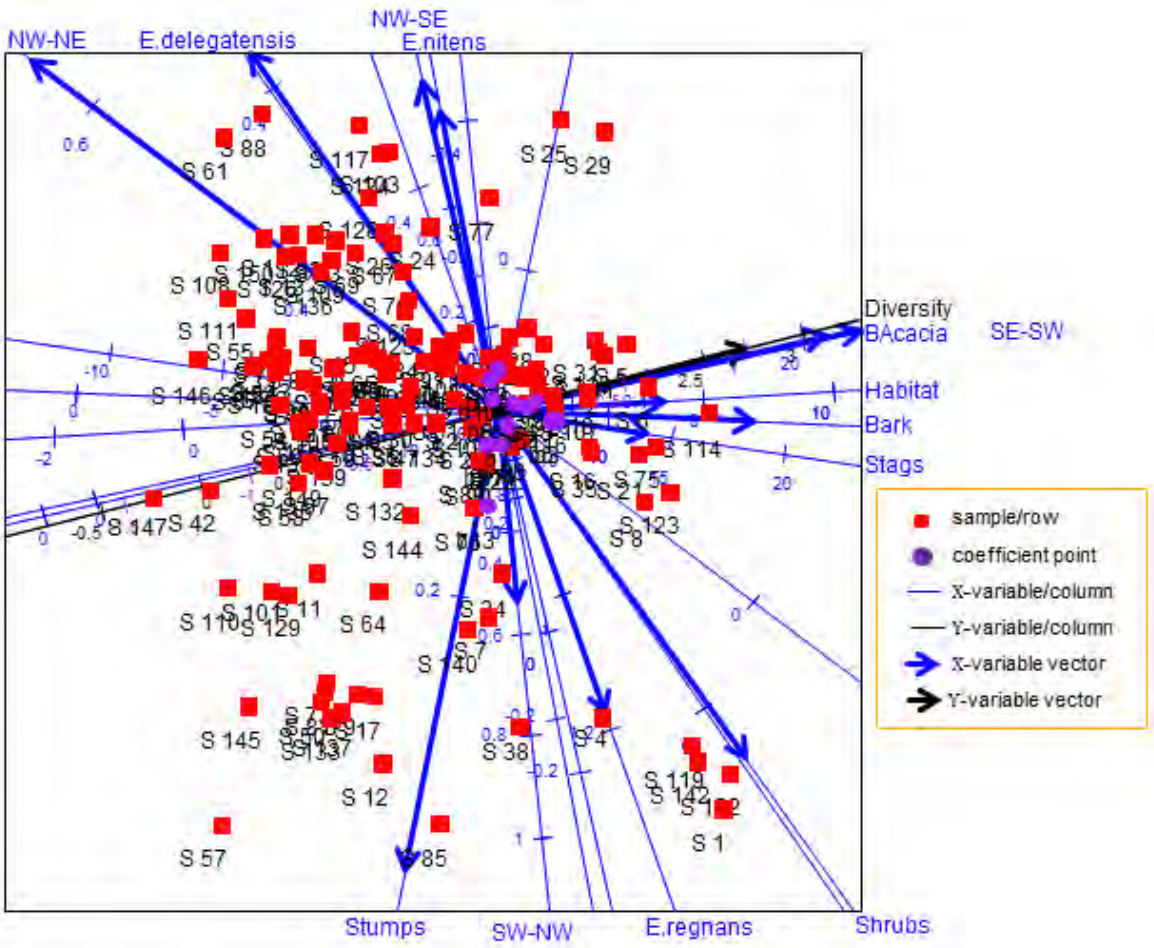

Figure 6.4 The PLS biplot of a Poisson PLS-GLM of the possum diversity data, fitted using the SIMPLS algorithm (Algorithm 6.3).

Comparing Figure 6.4 to Figure 6.1, similar deductions can be observed. This is obvious, seeing as both biplot displays are identical. However, this is not always the case for all data sets. For example, consider Figures 8.25 and 8.27 in Chapter 8. Although the same data was used for the PLS-GLM fitted using Algorithm 6.2 and the SIMPLS algorithm (Algorithm 6.3), different biplot displays were obtained.

6.6 Example with $M > 1$ Y-variables

The following example is an illustration of the PLS biplot to five separate Poisson GLMs, using the bio-env data from Greenacre (2010). A copy of this data can be found on the dropbox link

https://www.dropbox.com/sh/wr66u07t1vjm9da/AACg_E4h8MvgOHuCXk69yDIya

under the "*Data Sets*" folder or at

http://www.multivariatestatistics.org/data.html.

This data shows the biological species and environmental variables observed at a particular location on a sea-bed. There were five groups of species (A, B, C, D and E) considered, and four environmental variables (Pollution, Depth, Temperature and Sediment) were used. The first three environmental variables are measured on a continuous scale, while the sediment variable is a categorical variable that classifies the substrate of the sample into three groups - Sand, Clay and Grave. For this reason, the sediment variable was coded as a set of dummy variables with Clay left out as the reference category, thereby increasing the number of environmental variables used to five (Pollution, Depth, Temperature, Sediment_S and Sediment_G). The thirty sites are assigned as the samples. The species and the environmental variables are the response and predictor variables respectively. Thus, the bio-env data can be viewed as a data matrix $\mathbf{X}$ (30×5) of predictors and a matrix $\mathbf{Y}$ (30×5) of responses. As the response variables $\mathbf{Y}$ are counts, the GLM methodology discussed in Section 6.2 applies. Each single response variable $\mathbf{y}$ is analyzed separately since the discussed GLM methodology is only developed for single response models. Thus, five separate (Poisson) PLS-GLMs, using Algorithm 6.2, will be fitted for the five separate response variables. The resulting PLS biplots are shown in Figures 6.5 to 6.9, *along with a representation of the variance of each variable*. In these biplot displays, comparing the length of the thicker arrows (vectors) to each other, Temperature can be said to have a large standard deviation, while Pollution has a small deviations. Although the same environmental variables were used in the PLS-GLMs, the approximated values of the environmental variables obtained from one PLS-GLM differ from those obtained in another PLS-GLM. This can be observed by the different biplot displays in Figures 6.5 to 6.9. The predicted coefficient values from the five PLS-GLMs are shown below in Table 6.2.

Table 6.2 The predicted coefficient values from the five Poisson PLS-GLMs.

	A	B	C	D	E
b1:Pollution	-0.537	-0.339	0.274	-0.274	-0.379
b2:Depth	-0.108	0.067	0.027	0.177	-0.392
b3:Temperature	-0.111	0.019	-0.058	0.065	0.102
b4:Sediment_S*	0.000	0.000	0.000	0.000	0.000
b5:Sediment_G*	0.000	0.000	0.000	0.000	0.000

* 0 indicating Clay

From the five PLS biplots, Figures 6.5 to 6.9, a negative relation exists between environmental variable Pollution and Species A, B, and D. On the other hand, a positive relation exists between

Species E and environmental variable Temperature, but a negative relation between Species C and Temperature.

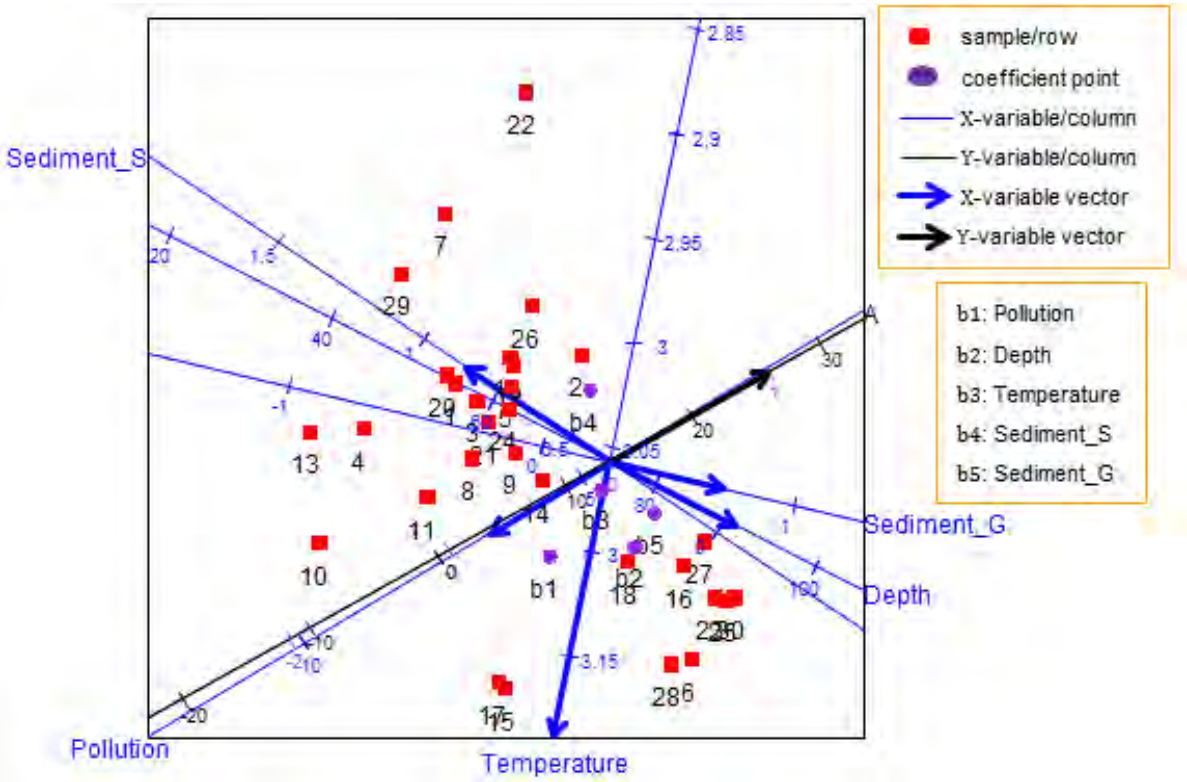

Figure 6.5 The PLS biplot of a Poisson PLS-GLM for species A of the bio-env data.

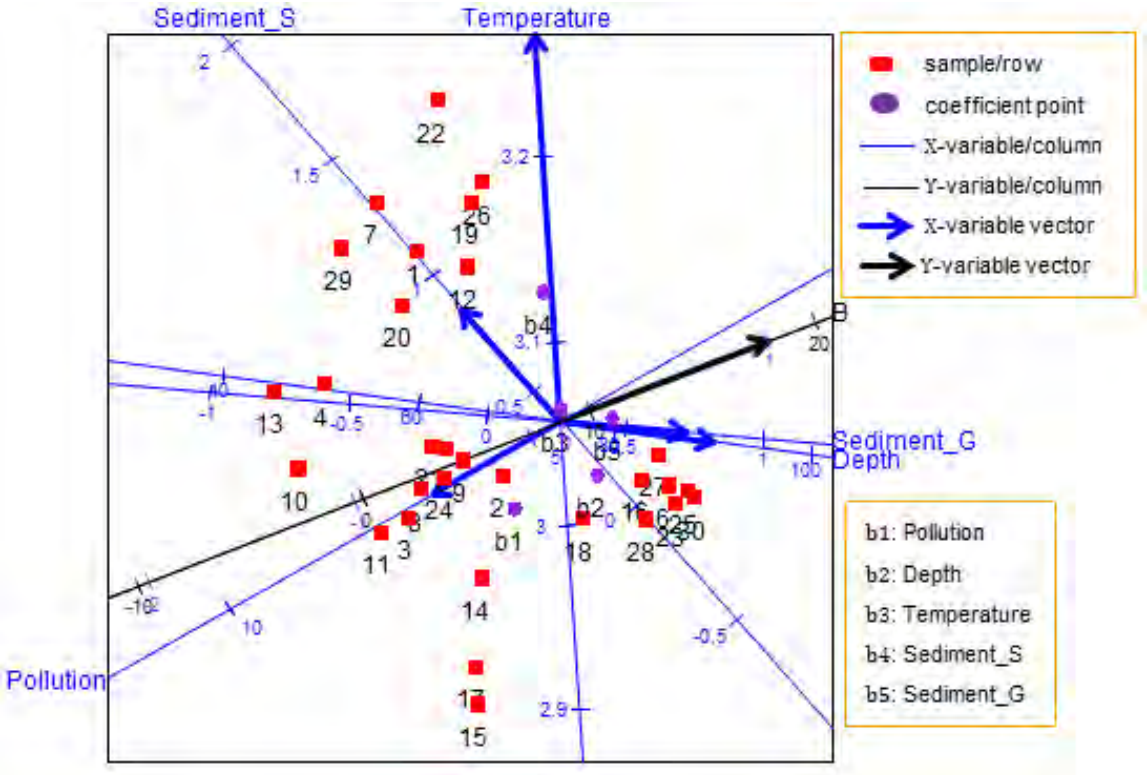

Figure 6.6 The PLS biplot of a Poisson PLS-GLM for species B of the bio-env data.

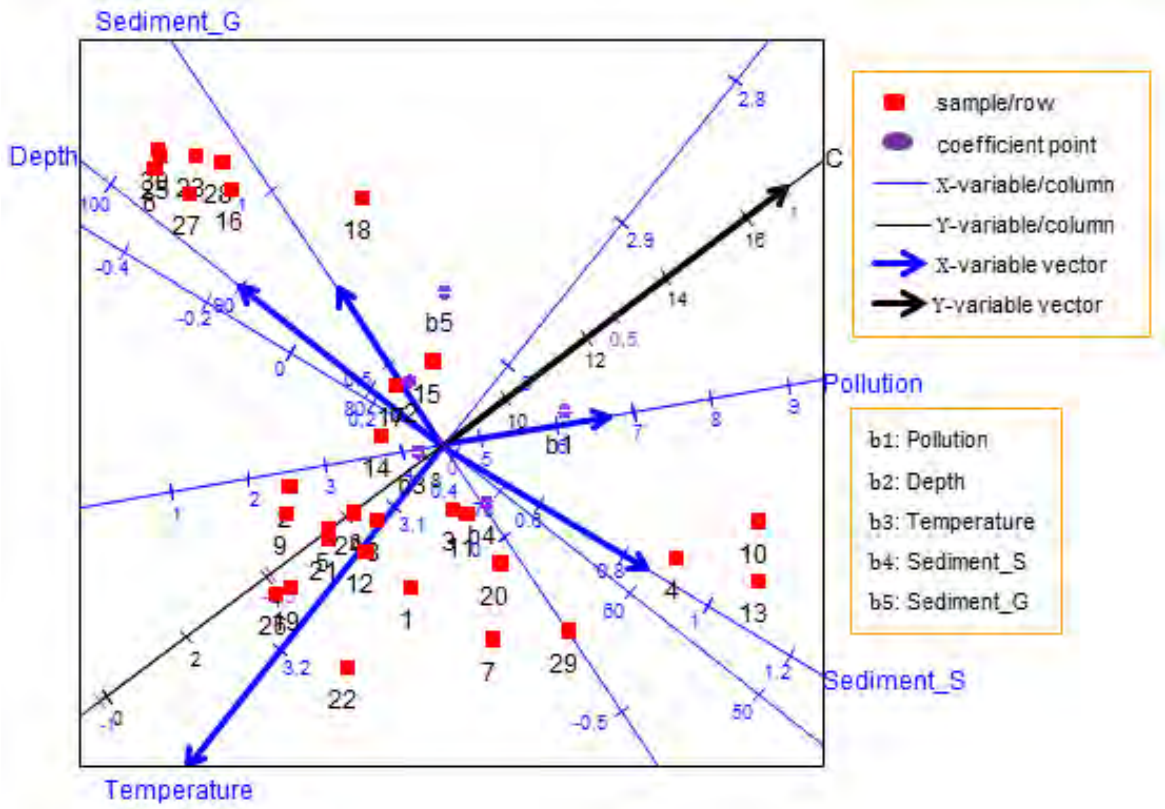

Figure 6.7 The PLS biplot of a Poisson PLS-GLM for species C of the bio-env data.

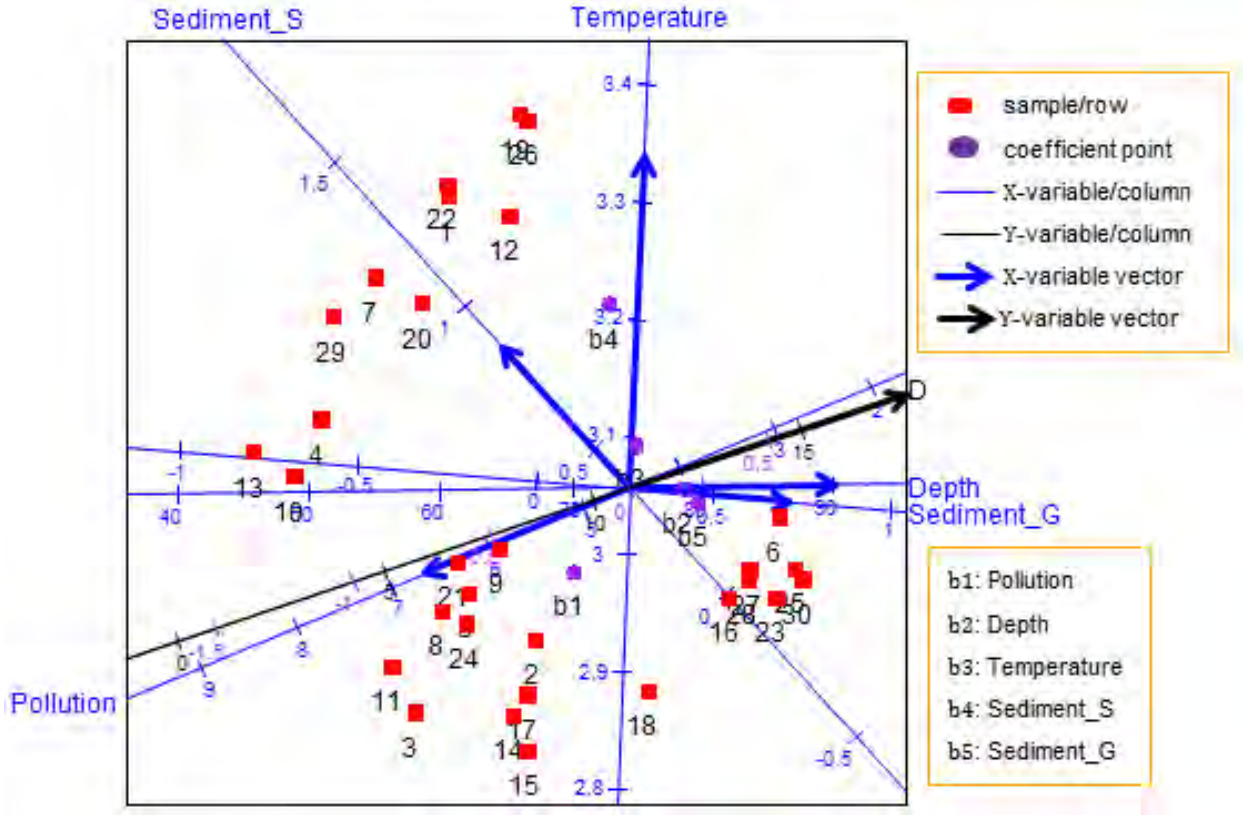

Figure 6.8 The PLS biplot of a Poisson PLS-GLM for species D of the bio-env data.

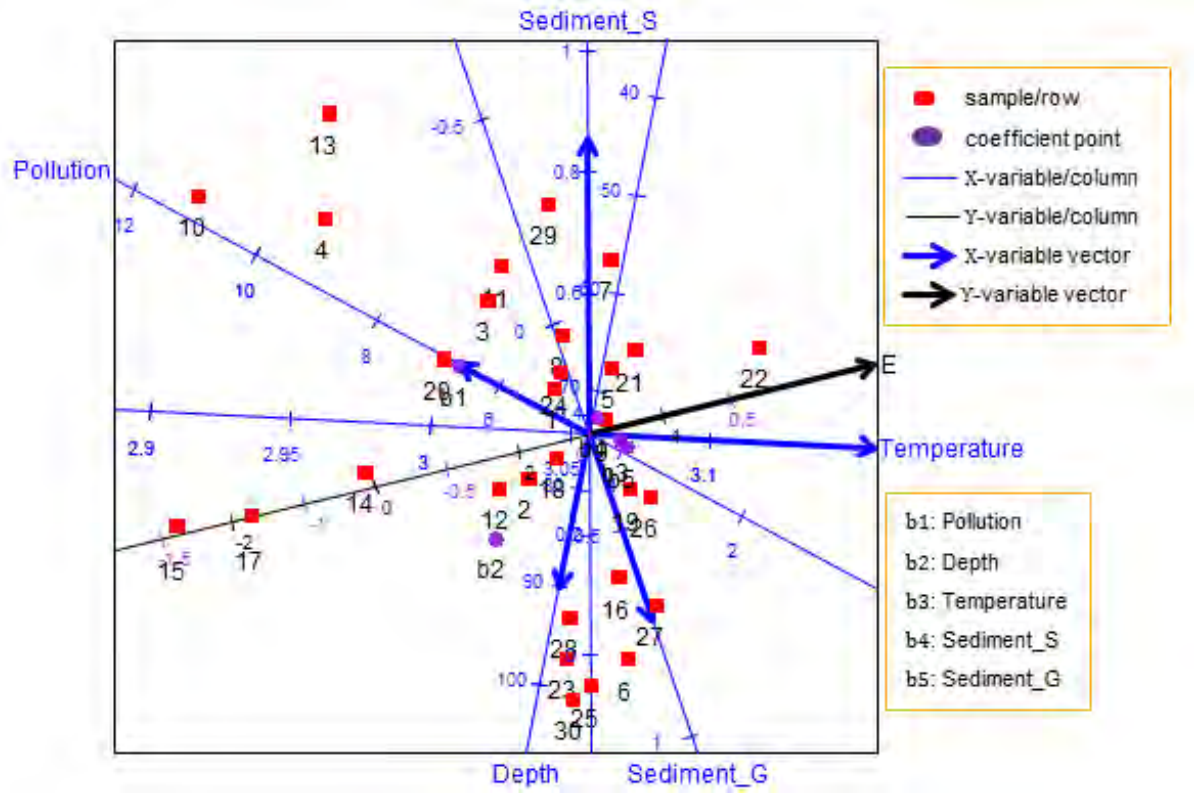

Figure 6.9 The PLS biplot of a Poisson PLS-GLM for species E of the bio-env data.

It is possible to merge these biplot displays together, to give an impression of what the PLS biplot would look like if a multivariate PLS-GLM framework was available and done on the data. This merging can be done using the Procrustes Analysis (PA) by Gower & Dijksterhuis (2004). The general idea of PA is to superimpose many configurations into one configuration. Consider a matrix $\mathbf{M}_1$ $(\mathrm{N} \times \mathrm{C1})$ and a matrix $\mathbf{M}_2$ $(\mathrm{N} \times \mathrm{C2})$, where $\mathrm{C1} = \mathrm{C2} = \mathrm{C}$. PA, specifically, Orthogonal Procrustes Analysis (OPA) in its simplest form transforms $\mathbf{M}_1$ to best fit $\mathbf{M}_2$. This transformation is done using an orthogonal matrix $\mathbf{O}$ $(\mathrm{C1} \times \mathrm{C2})$. To be precise, OPA seeks to find an orthogonal matrix $\mathbf{O}$ such that $\|\mathbf{M}_1\mathbf{O} - \mathbf{M}_2\|$ is minimized. Here,

$$\begin{aligned} \|\mathbf{M}_1\mathbf{O} - \mathbf{M}_2\| &= \mathrm{tr}\{(\mathbf{M}_1\mathbf{O} - \mathbf{M}_2)(\mathbf{M}_1\mathbf{O} - \mathbf{M}_2)^{\mathrm{T}}\} \\ &= \mathrm{tr}\{\mathbf{O}^{\mathrm{T}}\mathbf{M}_1^{\mathrm{T}}\mathbf{M}_1\mathbf{O}\} + \mathrm{tr}\{\mathbf{M}_2^{\mathrm{T}}\mathbf{M}_2\} - 2\mathrm{tr}\{\mathbf{M}_2^{\mathrm{T}}\mathbf{M}_1\mathbf{O}\}. \end{aligned} \quad (6.20)$$

Since $\mathbf{O}\mathbf{O}^{\mathrm{T}} = \mathbf{I}_{\mathrm{C1}}$,

$$\begin{aligned} \mathrm{tr}\{\mathbf{O}^{\mathrm{T}}\mathbf{M}_1^{\mathrm{T}}\mathbf{M}_1\mathbf{O}\} &= \mathrm{tr}\{\mathbf{O}\mathbf{O}^{\mathrm{T}}\mathbf{M}_1^{\mathrm{T}}\mathbf{M}_1\mathbf{O}\mathbf{O}^{\mathrm{T}}\} \\ &= \mathrm{tr}\{\mathbf{M}_1^{\mathrm{T}}\mathbf{M}_1\}. \end{aligned}$$

Thus, from (6.20), $\|\mathbf{M}_1\mathbf{O} - \mathbf{M}_2\| = \mathrm{tr}\{\mathbf{M}_1^{\mathrm{T}}\mathbf{M}_1\} + \mathrm{tr}\{\mathbf{M}_2^{\mathrm{T}}\mathbf{M}_2\} - 2\mathrm{tr}\{\mathbf{M}_2^{\mathrm{T}}\mathbf{M}_1\mathbf{O}\}$. By the SVD, $\mathbf{M}_2^{\mathrm{T}}\mathbf{M}_1 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$. Now,

$$\begin{aligned} \mathrm{tr}\{\mathbf{M}_2^{\mathrm{T}}\mathbf{M}_1\mathbf{O}\} = \mathrm{tr}\{\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}\mathbf{O}\} &= \mathrm{tr}\{\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}\mathbf{O}\mathbf{U}\} \\ &= \mathrm{tr}\{\mathbf{\Sigma}\mathbf{H}\} \\ &= \textstyle\sum_{\mathrm{i}=1}^{\mathrm{C}} \sigma_{\mathrm{ii}} \mathrm{h}_{\mathrm{ii}} \end{aligned}$$

where $\mathbf{H} = \mathbf{V}^{\mathrm{T}}\mathbf{O}\mathbf{U}$. Since $\mathbf{V}$, $\mathbf{O}$ and $\mathbf{U}$ are orthogonal, this implies that $\mathbf{H} = \mathbf{V}^{\mathrm{T}}\mathbf{O}\mathbf{U} = \mathbf{I}_{\mathrm{C}}$. Therefore, $\mathbf{I}_{\mathrm{C}} = \mathbf{V}^{\mathrm{T}}\mathbf{O}\mathbf{U}$, implies that

$$\mathbf{O} = \mathbf{V}\mathbf{U}^{\mathrm{T}}. \tag{6.21}$$

Figure 6.10 below illustrates the OPA, using two different triangles. In this figure, $\mathbf{M}_1$ (blue triangle) has been rotated anti-clockwise to best fit $\mathbf{M}_2$ (black triangle). As a result, triangle $\mathbf{M}_1{}^{\mathrm{OPA}}$ is obtained.

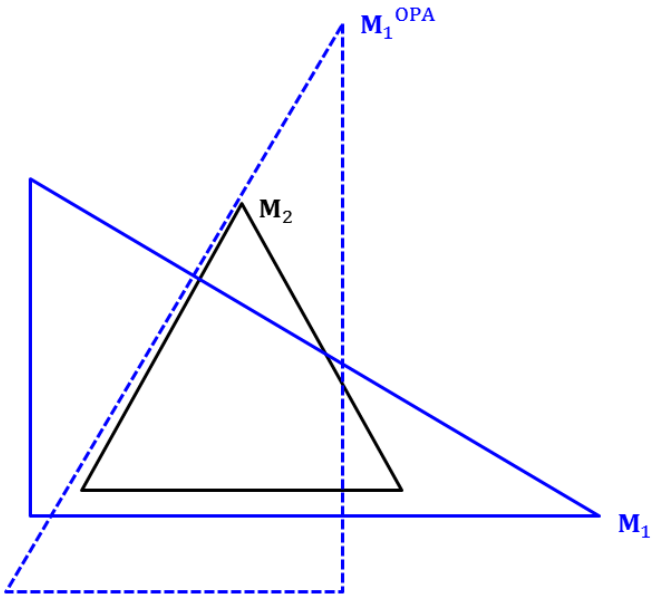

Figure 6.10 A schematic of OPA on two dissimilar triangles.

It is possible to apply OPA to more than two matrices. Let $\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_{\mathrm{K}}$ denote these matrices. The general idea of OPA is still the same, except here, matrices $\mathbf{O}_1, \mathbf{O}_2, \cdots, \mathbf{O}_{\mathrm{K}}$ is found such that

$$\mathrm{K}\sum_{\mathrm{k}=1}^{\mathrm{K}}\|\mathbf{M}_{\mathrm{k}}\mathbf{O}_{\mathrm{k}} - \mathbf{G}\| = \left(\frac{\mathrm{K}-1}{\mathrm{K}}\right)^2 \sum_{\mathrm{k}=1}^{\mathrm{K}}\|\mathbf{M}_{\mathrm{k}}\mathbf{O}_{\mathrm{k}} - \mathbf{G}_{\mathrm{k}}\| \tag{6.22}$$

is minimized over all $\mathbf{O}_{\mathrm{k}}$, where

$$\mathbf{G} = \frac{1}{\mathrm{K}}\sum_{\mathrm{k}=1}^{\mathrm{K}}\mathbf{M}_{\mathrm{k}}\mathbf{O}_{\mathrm{k}} \text{ and } \mathbf{G}_{\mathrm{k}} = \frac{1}{\mathrm{K}-1}\sum_{\mathrm{i}\neq\mathrm{k}}^{\mathrm{K}}\mathbf{M}_{\mathrm{i}}\mathbf{O}_{\mathrm{i}}$$

are the group-average and k -excluded group-average configurations. In other words, $\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_{\mathrm{K}}$ is transformed to best fit $\mathbf{G}$, using $\mathbf{O}_1, \mathbf{O}_2, \cdots, \mathbf{O}_{\mathrm{K}}$. This situation is referred to as the Generalized Orthogonal Procrustes Analysis (GOPA) and the solution is based on an iterative application of OPA. For further in-dept details on GOPA, see Gower & Dijksterhuis (2004). With the idea of superimposing the five PLS biplot displays shown in Figures 6.5 to 6.9 into one, the problem can be interpreted as finding $\mathbf{O}_1, \mathbf{O}_2, \cdots, \mathbf{O}_5$ such that

$$5\sum_{\mathrm{k}=1}^{5}\left\|\mathbf{P}_{*\mathrm{k}}\mathbf{O}_{\mathrm{k}} - \mathbf{G}\right\| \tag{6.23}$$

is minimized, where $K = 5$, $k = 1, 2, \cdots, 5$ and $\mathbf{G} = \frac{1}{5}\sum_{k=1}^{5}\mathbf{P}_{*k}\mathbf{O}_k$. Note, $\mathbf{P}_{*k}$ is the X-loadings matrix of each of the (five) PLS biplot displays shown in Figures 6.5 to 6.9. Since in these figures (Figures 6.5 to 6.9),

samples points (red) are represented by the rows of $\mathbf{T}_*$,

coefficient points (purple) are represented by the rows of $\mathbf{R}_*$ and

axes (blue and black) are represented by the rows of $\begin{bmatrix}\mathbf{P}_* \\ \mathbf{q}_*^T\end{bmatrix}$,

in the forthcoming combined display,

samples points (red) will be represented by the rows of $\mathbf{T}_{*k}\mathbf{O}_k$,

coefficient points (purple) will be represented by the rows of $\mathbf{R}_{*k}\mathbf{O}_k$ and

axes (blue and black) will be represented by the rows of $\begin{bmatrix}\mathbf{P}_{*k} \\ \mathbf{q}_{*k}^T\end{bmatrix}\mathbf{O}_k$

for $k = 1, 2, \cdots, 5$.

Applying GOPA to Figures 6.5 to 6.9 brings about Figure 6.11, without the coefficient points. In this figure, there are five (different) Pollution, Depth, Temperature, Sediment_S and Sediment_G axes, resulting from the five (different) PLS biplots in Figures 6.5 to 6.9. Likewise, five different sets of sample points are shown in this figure, due to the five different PLS biplots in Figures 6.5 to 6.9. However, since there is only one specie variable (black axis) per PLS biplot display in Figures 6.5 to 6.9, only one A, B, C, D and E axes will be in the combined display, as seen in Figure 6.11. To differentiate between these sets of environmental variables and sample points, an addition of .a, .b, .c, .d and .e have been added to their respective names in Figure 6.11. Here, .a, .b, .c, .d and .e means that the respective entities are from Figures 6.5, 6.6, 6.7, 6.8 and 6.9 respectively. From the combined display (Figure 6.11), one can conclude that there could be a relation between Pollution and species A, B and D, as well as between Temperature and species E and C. However, since Figure 6.11 is not the PLS biplot display of a multivariate PLS-GLM done on the bio-env data, these conclusions cannot be authenticated. Another presentation of the combined display (Figure 6.11) is shown in Figure 6.12, where only the sample group average points are shown, along with the combined axes. These group average points are defined by the N rows of $\frac{1}{5}\sum_{k=1}^{5}\mathbf{T}_{*k}\mathbf{O}_k$. Here, $N = 30$, hence, there are 30 different group average points in Figure 6.12.

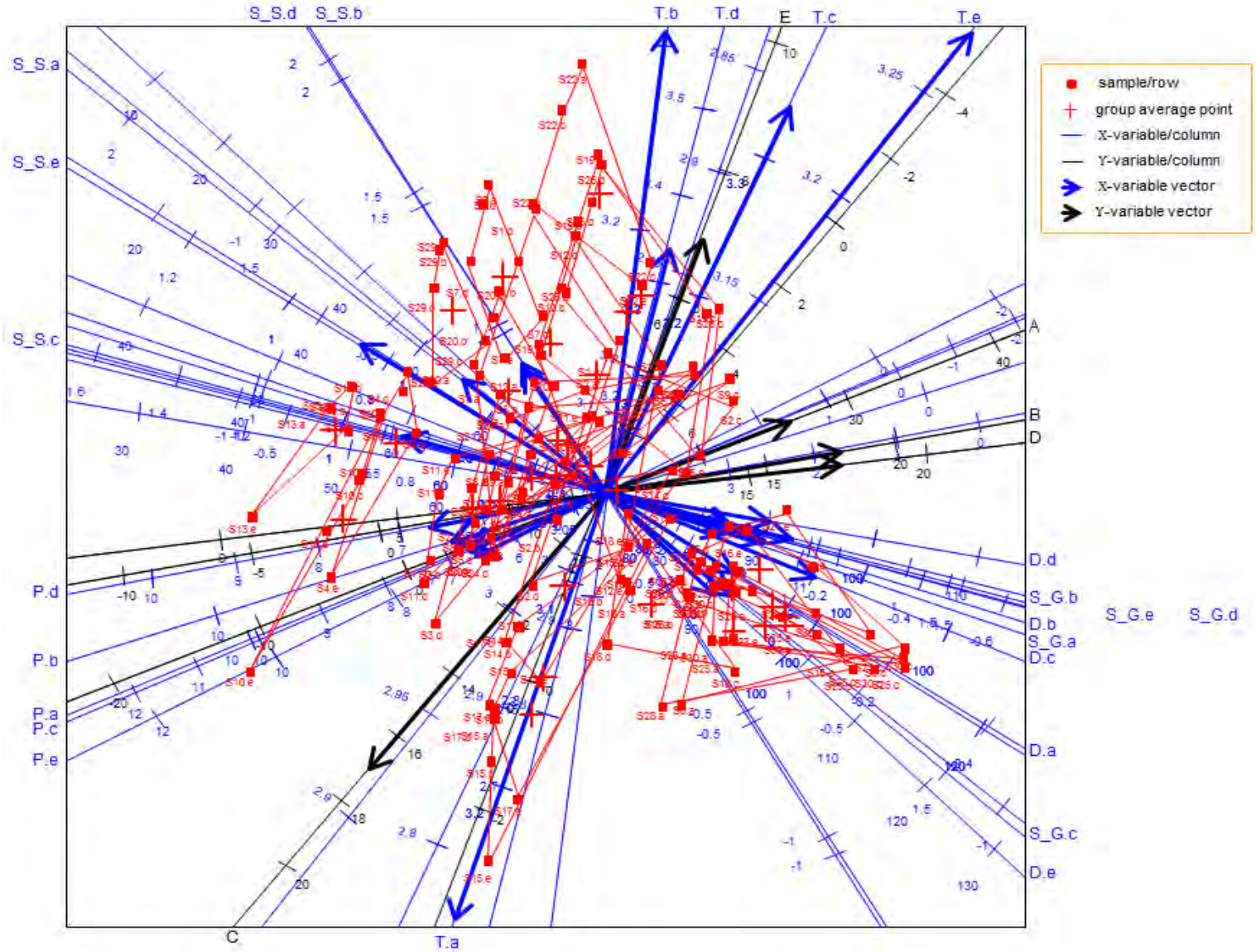

Figure 6. 11 The GOPA display of the PLS biplots in Figures 6.5 to 6.9, without the coefficient points.

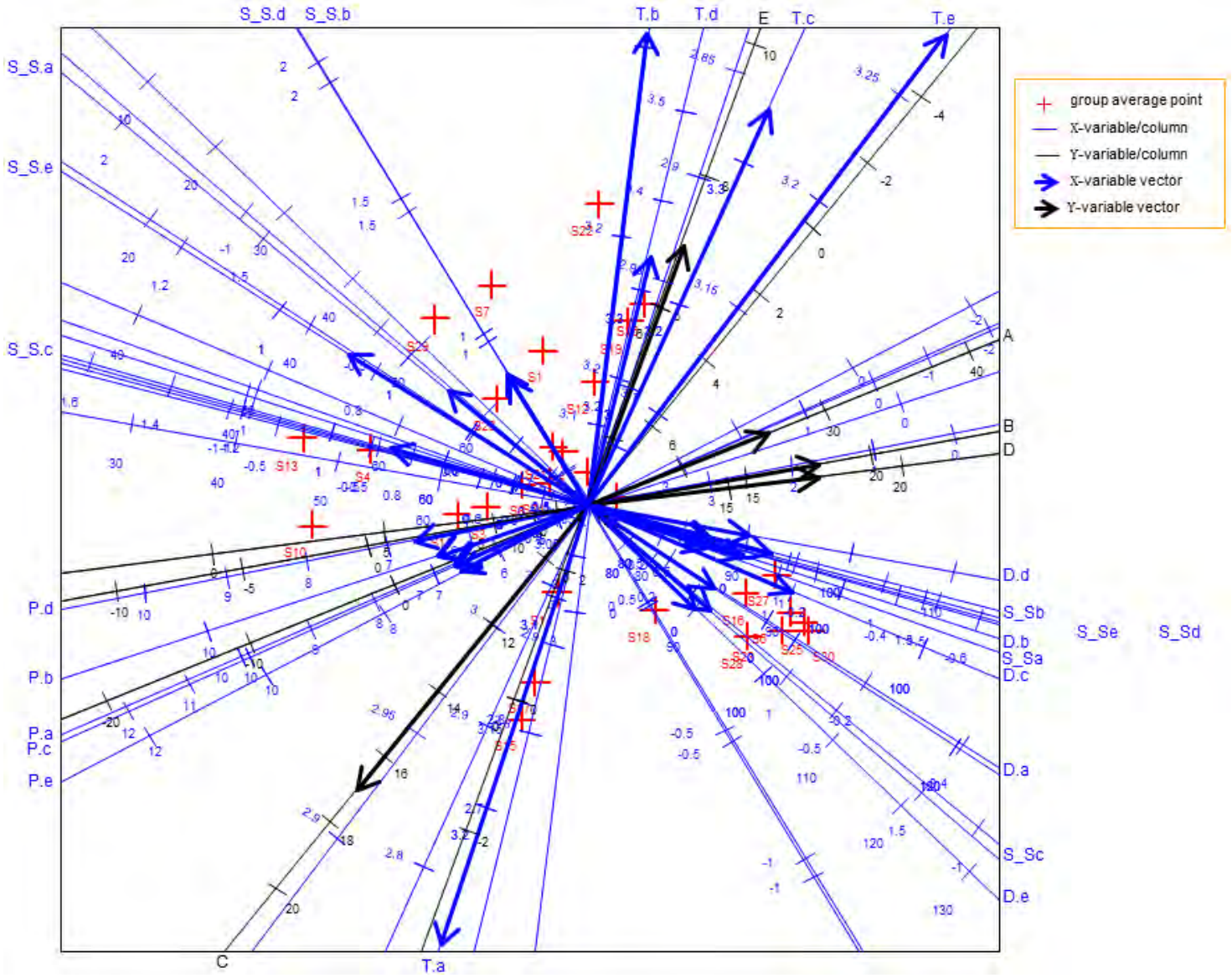

Figure 6. 12 The sample group average points of the GOPA display of the PLS biplots in Figures 6.5 to 6.9, without the coefficient points.

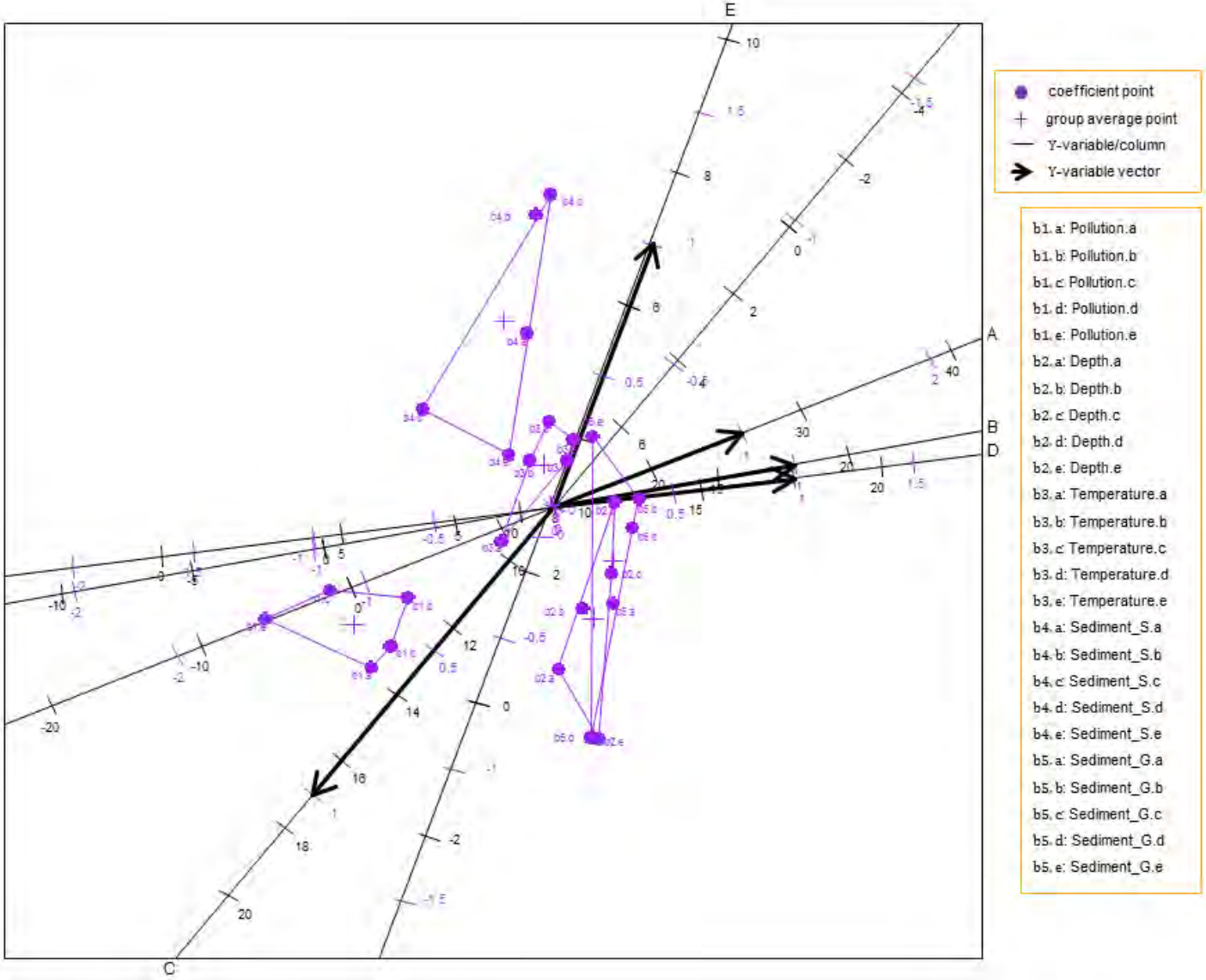

Figure 6. 13 The coefficient points of the GOPA display of the PLS biplots in Figures 6.5 to 6.9.

Furthermore, there will be five different sets of coefficient points in the combined display (Figure 6.11) as a result of the five different PLS biplot displays in Figures 6.5 to 6.9. These coefficient points are not shown in the combined display (Figure 6.11), for clear visibility of the sample points, but are shown above in Figure 6.13. A representation of the coefficient group average points are shown in Figure 6.13 as well. These group average points are defined by the P rows of $\frac{1}{5}\sum_{k=1}^{5}\mathbf{R}_{*k}\mathbf{O}_k$, thus, there will be $P = 5$ different coefficient group average points in Figure 6.13. To get an impression of the multivariate PLS-GLM coefficient values, each of the purple points in Figure 6.13 is projected onto their respective axes and their value read off using the *purple markers on these axes*. The respective axis name on which to project on, for a particular coefficient point, is given alongside the coefficient's name. *For example, coefficient points* $b4.d$ *and* $b1.a$ *are to be projected only onto axes D and A respectively, and not onto axes B, C or E*. However, since the univariate PLS-GLM (Algorithm 6.2) was used in fitting the five species separately in Figures 6.5 to 6.9, after which their resulting PLS biplot displays were combined together in Figure 6.11 (and Figure 6.13), the predicted coefficient values obtained from Figure 6.13 are still the univariate PLS-GLM coefficients. Therefore, the coefficient values obtained from Figure 6.13 are still the values shown in Table 6.2.

6.7 Summary

When there is the need to model a set of response variables from a (very) large set of predictor variables and the relationship between the response variables and predictors follows a non-linear function, the PLS-GLM can be a useful tool. Results found by the PLS-GLM of a data set can be visualized graphically using biplots, specifically, the PLS biplot.

Furthermore, this chapter only considered a PLS-GLM with $M = 1$ response variables. It can be expanded to $M > 1$ response variables, but more work is needed on the development of a general framework for multivariate ($M > 1$) GLMs and PLS-GLMs.

CHAPTER 7

BIPLOTS FOR SPARSE PARTIAL LEAST SQUARES

7.1 Introduction

In biological data, such as genomics, the set of predictor variables can be very large and sometimes strongly correlated. As a result, for the modelling of the response variables, a careful variable selection of the predictors is necessary. Methods such as PLSR can be useful for modelling such data. For the application of PLS on biological data, Lê Cao *et al*. (2008) proposed a sparse version of PLS to help facilitate biological interpretations for biologists. They termed this method *Sparse Partial Least Squares (SPLS)*. This method is based on the *SVD application to PLS and the addition of a sparsity constraint with a soft-thresholding penalization*. More precisely, given a set of predictor variables $\mathbf{X}$ (N × P) and a set of response variables $\mathbf{Y}$ (N × M), the SVD is applied on $\mathbf{S} = \mathbf{X}^{\mathrm{T}}\mathbf{Y}$ and then the PLS X-weight vector $\mathbf{w}$ (P × 1) or $\mathbf{r}$ (P × 1) and Y-weight vector $\mathbf{c}$ (M × 1) are subjected to the specified sparsity constraint.

In this chapter, a brief introduction to the principle of the SVD application to PLS approach and soft-thresholding penalization is provided, before the SPLS method is discussed. In addition, the extension of SPLS to the GLM framework is discussed. This is useful when a non-linear relationship exist between the variables. As a visual tool for displaying the SPLS and/or the Sparse Partial Least Squares-Generalized Linear Model (SPLS-GLM) of a data set, the PLS biplot (Chapter 5) is proposed.

7.2 Singular Value Decomposition application to Partial Least Squares

Consider a set of predictor variables $\mathbf{X}$ (N × P) and a set of response variables $\mathbf{Y}$ (N × M). In this approach, the SVD of $\mathbf{S} = \mathbf{X}^{\mathrm{T}}\mathbf{Y}$ is performed once (Lê Cao *et al*., 2008). By the SVD,

$$\mathbf{S} = \mathbf{D}\boldsymbol{\Lambda}\mathbf{F}^{\mathrm{T}}. \tag{7.1}$$

The (P × 1) and (M × 1) PLS weight vectors $\mathbf{r}_1$ and $\mathbf{c}_1$ are then obtained as the first left and right singular vectors of $\mathbf{S}$ respectively. Thereafter, the first set of score vectors $\mathbf{t}_1$ (N × 1) and $\mathbf{u}_1$ (N × 1) and loading vectors $\mathbf{p}_1$ (P × 1) and $\mathbf{q}_1$ (M × 1) are computed, as in Algorithm 3.4. For the next sets of PLS vectors, the matrix $\mathbf{S}$ in (7.1) is replaced by

$$\mathbf{S}_{\mathrm{a}} = \mathbf{S}_{\mathrm{a}-1} - \delta_{\mathrm{a}-1}\mathbf{r}_{\mathrm{a}-1}\mathbf{c}_{\mathrm{a}-1}^{\mathrm{T}} \tag{7.2}$$

where $\delta_{\mathrm{a}-1}$ is the $(\mathrm{a}-1)^{\mathrm{th}}$ singular value of $\mathbf{S}_{\mathrm{a}-1}$, $\mathrm{a} = 2, 3, \cdots, \mathrm{A}$ and $\mathbf{S}_1 = \mathbf{S}$ in (7.1).

7.3 Soft-thresholding penalization

Consider a penalty parameter λ and a set of numbers $Ƌ \in (-\infty, \infty)$. In general, a soft-thresholding penalization assigns every number in $Ƌ$ closer to zero than λ a zero value, while a number farther from zero than λ is assigned the value obtained after subtracting the number from λ. To illustrate, consider $Ƌ = (2,\ 5,\ -2,\ 0,\ -5,\ 0.7)$ and $\lambda = 1$. Applying the soft-thresholding penalization on $Ƌ$ yields $(1,\ 4,\ -1,\ 0,\ -4,\ 0)$.

7.4 Sparse Partial Least Squares

Keeping the objective of PLS in mind (Section 3.3), the objective function for the SPLS can be expressed as follows. For $a = 1, 2, \ldots, A$,

$$\begin{aligned} &\text{maximize } \operatorname{cov}(\mathbf{r}_a^T \mathbf{X}^T \mathbf{Y}\mathbf{Y}^T \mathbf{X}\mathbf{r}_a) = \operatorname{cov}(\mathbf{t}_a^T \mathbf{Y}\mathbf{Y}^T \mathbf{t}_a) \\ &\text{subject to } \mathbf{r}_a = g_{\lambda_X}(\mathbf{X}^T \mathbf{Y}\mathbf{c}_a) \le \lambda_X \text{ and } \mathbf{c}_a = g_{\lambda_Y}(\mathbf{Y}^T \mathbf{X}\mathbf{r}_a) \le \lambda_Y \end{aligned}$$

where

$$\begin{aligned} g_{\lambda_X}(\mathbf{X}^T \mathbf{Y}\mathbf{c}_a) &= \operatorname{sign}(\mathbf{X}^T \mathbf{Y}\mathbf{c}_a)(|\mathbf{X}^T \mathbf{Y}\mathbf{c}_a| - \lambda_X)_+ \\ g_{\lambda_Y}(\mathbf{Y}^T \mathbf{X}\mathbf{r}_a) &= \operatorname{sign}(\mathbf{Y}^T \mathbf{X}\mathbf{r}_a)(|\mathbf{Y}^T \mathbf{X}\mathbf{r}_a| - \lambda_Y)_+, \end{aligned} \tag{7.3}$$

$\mathbf{t}_a = \mathbf{X}\mathbf{r}_a$, $b = 1,\ 2, \cdots, A-1$ and $b \neq a$. The SPLS objective function can be solved by using the SPLS algorithm (Lê Cao *et al*., 2008).

The expressions in (7.3) are the Least Absolute Shrinkage and Selection Operator (LASSO) penalization function for the weight vectors $\mathbf{r}_a$ and $\mathbf{c}_a$ (Tibshirani, 1996). Hastie *et al*. (2009) and Yoshida *et al*. (2013) termed these functions the *soft-thresholding penalization functions*, due to the LASSO performing a 'soft' version of the Best Subset penalization (another penalization function). In the Best Subset penalization, which is termed the *hard-thresholding penalization*, all variables with coefficients smaller than the K^{th} largest coefficient are removed from the analysis, whereas in the LASSO penalization, only variables having non-zero coefficients are selected and used. Thus, (7.3) can be referred to as the *soft-thresholding LASSO penalty function* for the weight vectors $\mathbf{r}_a$ and $\mathbf{c}_a$. The notation discussed in Chapter 1 is applicable in this section. The SPLS algorithm by Lê Cao *et al*. (2008) is summarized as follows.

Algorithm 7.1:

(1) $\mathbf{S}_1 = \mathbf{X}_0^T \mathbf{Y}_0$ and $a = 1$.

(2) Compute $\mathbf{r}_a$ $(P \times 1)$ and $\mathbf{c}_a$ $(M \times 1)$ as

$$\begin{aligned} \mathbf{S}_a &= \mathbf{D}\mathbf{\Lambda}\mathbf{F}^T, \text{ by the SVD} \\ \mathbf{r}_a &= \mathbf{D}[,\ 1] \\ \mathbf{c}_a &= \mathbf{F}[,\ 1]. \end{aligned}$$

(3) Penalize $\mathbf{r}_a$ $(P \times 1)$ and $\mathbf{c}_a$ $(M \times 1)$ with a soft-thresholding penalization (7.3) as

$$\mathbf{r}_{a_{new}} = g_{\lambda_X}(\mathbf{S}_a\mathbf{c}_a) / \left\| g_{\lambda_X}(\mathbf{S}_a\mathbf{c}_a) \right\|$$
$$\mathbf{c}_{a_{new}} = g_{\lambda_Y}(\mathbf{S}_a^T\mathbf{r}_a) / \left\| g_{\lambda_Y}(\mathbf{S}_a^T\mathbf{r}_a) \right\|$$

where

$$g_{\lambda_X}(\mathbf{S}_a\mathbf{c}_a) = \text{sign}(\mathbf{S}_a\mathbf{c}_a)(|\mathbf{S}_a\mathbf{c}_a| - \lambda_X)_+ \text{ and}$$
$$g_{\lambda_Y}(\mathbf{S}_a^T\mathbf{r}_a) = \text{sign}(\mathbf{S}_a^T\mathbf{r}_a)(|\mathbf{S}_a^T\mathbf{r}_a| - \lambda_Y)_+$$

are the soft-thresholding penalization functions, with penalty parameters λ_X and λ_Y.

(4) Check whether $\mathbf{r}_{a_{new}}$ and $\mathbf{c}_{a_{new}}$ converges. If not, return to step (3), with $\mathbf{r}_a = \mathbf{r}_{a_{new}}$ and $\mathbf{c}_a = \mathbf{c}_{a_{new}}$. Here, convergence is reached when $\left\|\mathbf{r}_{a_{new}} - \mathbf{r}_a\right\| / \|\mathbf{r}_a\|$ and $\left\|\mathbf{c}_{a_{new}} - \mathbf{c}_a\right\| / \|\mathbf{c}_a\|$ are both small, say, less than 10^{-5}.

(5) Compute $\mathbf{t}_a$ $(N \times 1)$, $\mathbf{u}_a$ $(N \times 1)$, $\mathbf{p}_a$ $(P \times 1)$ and $\mathbf{q}_a$ $(M \times 1)$ as

$$\mathbf{t}_a = \mathbf{X}_a\mathbf{r}_{a_{new}} / \left\|\mathbf{X}_a\mathbf{r}_{a_{new}}\right\|$$
$$\mathbf{u}_a = \mathbf{Y}_a\mathbf{c}_{a_{new}} / \left\|\mathbf{Y}_a\mathbf{c}_{a_{new}}\right\|$$
$$\mathbf{p}_a = \mathbf{X}_a^T\mathbf{t}_a$$
$$\mathbf{q}_a = \mathbf{Y}_a^T\mathbf{t}_a.$$

(6) Update $\mathbf{S}_a$ $(P \times M)$, for the next latent variable, as

$$\mathbf{S}_{a+1} = (\mathbf{I}_P - \mathbf{p}_a(\mathbf{p}_a^T\mathbf{p}_a)^{-1}\mathbf{p}_a^T)\mathbf{S}_a.$$

(7) Return to step (2), with $a = a + 1$, to compute the next latent variable until $a = A$ latent variables have been reached.

(8) Store $\mathbf{r}_{a_{new}}$, $\mathbf{c}_{a_{new}}$, $\mathbf{t}_a$, $\mathbf{u}_a$, $\mathbf{p}_a$ and $\mathbf{q}_a$ into the successive columns of $\mathbf{R}_\#$, $\mathbf{C}_\#$, $\mathbf{T}_\#$, $\mathbf{U}_\#$, $\mathbf{P}_\#$ and $\mathbf{Q}_\#$.

Compared to the standard PLS (Chapter 3), SPLS has an additional advantage of performing variables selection, besides components extraction, in its analysis. In the standard PLS, using the NIPALS, kernel or SIMPLS algorithms, only components extraction is done in the analysis. One has to perform variables selection separately, using techniques such as the VIP (Section 3.9), prior to the PLS analysis. However, this is not the case in the SPLS analysis. Both components extraction and variables selection are done in the SPLS analysis. In the SPLS analysis, the variables selection is achieved by introducing the soft-thresholding LASSO penalization on the pair of X- and Y-weight vectors $\mathbf{r}_a$ and $\mathbf{c}_a$ (7.3).

7.4.1 *Choosing a value for the penalty parameters* λ_X *and* λ_Y

Although publications such as Chun & Keles (2010) and Tibshirani (1996) arbitrarily chose the values of the penalty parameters λ_X and λ_Y in their applications, Hastie *et al*. (2009) and Lê Cao *et al*. (2008) suggested that these parameters should be chosen so that a minimum error of prediction is obtained. With this suggestion in mind, the following experiment is proposed. To choose a value for the penalty parameters λ_X and λ_Y, the SPLS analysis of the data is performed

using various pairs (λ_X, λ_Y) values, where $\lambda_X \in \mathbb{R}$ and $\lambda_Y \in \mathbb{R}$. Let L be the (desired) number of (λ_X, λ_Y) pairings. The RMSEP value (Section 3.7) per SPLS analysis is recorded. Since there are L number of pairs (λ_X, λ_Y), there will be L different SPLS analyses performed and L different RMSEP values obtained. Afterwards, the pairs (λ_X, λ_Y) with the lowest RMSEP value are recorded. Let $(\lambda_X^*, \lambda_Y^*)$ denote a pair having the lowest RMSEP value. Thereafter, the value for λ_X and λ_Y to be used in the final SPLS analysis of the data is chosen as the value of λ_X^* and λ_Y^*, with the lowest RMSEP value.

The main purpose of λ_X and λ_Y is in the variables selection part of SPLS. With the penalty functions

$$\mathbf{r}_{a_{new}} = \text{sign}(\mathbf{S}_a\mathbf{c}_a)(|\mathbf{S}_a\mathbf{c}_a| - \lambda_X)_+ \text{ and}$$
$$\mathbf{c}_{a_{new}} = \text{sign}(\mathbf{S}_a^T\mathbf{r}_a)(|\mathbf{S}_a^T\mathbf{r}_a| - \lambda_Y)_+,$$

if the number of Y-variables (M) is small, then there is no need to perform variables selection on the Y-variables. Thus, $\lambda_Y = 0$ and the search in the abovementioned experiment would be for the value of λ_X that gives the minimum RMSEP value.

For $\lambda_X = 0 = \lambda_Y$,

$$g_0(\mathbf{S}_a\mathbf{c}_a) = \text{sign}(\mathbf{S}_a\mathbf{c}_a)(|\mathbf{S}_a\mathbf{c}_a| - 0)_+ = \mathbf{S}_a\mathbf{c}_a \text{ and}$$
$$g_0(\mathbf{S}_a^T\mathbf{r}_a) = \text{sign}(\mathbf{S}_a^T\mathbf{r}_a)(|\mathbf{S}_a^T\mathbf{r}_a| - 0)_+ = \mathbf{S}_a^T\mathbf{r}_a.$$

Then $\mathbf{r}_{a_{new}}$ and $\mathbf{c}_{a_{new}}$ in step (3) becomes

$$\mathbf{r}_{a_{new}} = g_0(\mathbf{S}_a\mathbf{c}_a)\,/\|g_0(\mathbf{S}_a\mathbf{c}_a)\| = \mathbf{S}_a\mathbf{c}_a\,/\|\mathbf{S}_a\mathbf{c}_a\| \text{ and}$$
$$\mathbf{c}_{a_{new}} = g_0(\mathbf{S}_a^T\mathbf{r}_a)\,/\|g_0(\mathbf{S}_a^T\mathbf{r}_a)\| = \mathbf{S}_a^T\mathbf{r}_a\,/\|\mathbf{S}_a^T\mathbf{r}_a\|,$$

which is equivalent to the SIMPLS algorithm (Algorithm 3.4), with $\mathbf{S}_a = \mathbf{D}\boldsymbol{\Lambda}\mathbf{F}^T$ and $\mathbf{c}_a = \mathbf{F}[,1]$ (step (2)),

$$\begin{aligned}\mathbf{r}_{a_{new}} &= \mathbf{D}\boldsymbol{\Lambda}\mathbf{F}^T\mathbf{F}[,1]\,/\|\mathbf{D}\boldsymbol{\Lambda}\mathbf{F}^T\mathbf{F}[,1]\| \\ &= \mathbf{D}[,1]\lambda_1\,/\|\mathbf{D}[,1]\lambda_1\| \\ &= \mathbf{r}_a\lambda_1\,/\|\mathbf{r}_a\lambda_1\| \\ &= \mathbf{r}_a\,/\|\mathbf{r}_a\|\end{aligned}$$

where $\mathbf{r}_a = \mathbf{D}[,1]$. Likewise,

$$\begin{aligned}\mathbf{c}_{a_{new}} &= \mathbf{F}\boldsymbol{\Lambda}\mathbf{D}^T\mathbf{D}[,1]\,/\|\mathbf{F}\boldsymbol{\Lambda}\mathbf{D}^T\mathbf{D}[,1]\| \\ &= \mathbf{F}[,1]\lambda_1\,/\|\mathbf{F}[,1]\lambda_1\| \\ &= \mathbf{c}_a\lambda_1\,/\|\mathbf{c}_a\lambda_1\| \\ &= \mathbf{c}_a\,/\|\mathbf{c}_a\|\end{aligned}$$

for $\mathbf{c}_a = \mathbf{F}[,1]$.

7.5 Sparse Partial Least Squares for Generalized Linear Models

For a response variable $\mathbf{y}$ (N × 1), the SPLS regression model is expressed as $\mathbf{y}_0 = \mathbf{X}_0\hat{\mathbf{b}}_{\text{SPLS}}$, where $\hat{\mathbf{b}}_{\text{SPLS}}$ (P × 1) $= \mathbf{R}_{\#}\mathbf{q}_{\#}$ from Algorithm 7.1. This model can be extended to a GLM such that

$$g(\boldsymbol{\mu}) = \mathbf{X}_0\hat{\mathbf{b}}_{\text{SPLS-GLM}} \tag{7.4}$$

where $\hat{\mathbf{b}}_{\text{SPLS-GLM}} = \mathbf{R}_+\mathbf{q}_+$ is the SPLS-GLM coefficient vector and both $\mathbf{R}_+$ (P × A) and $\mathbf{q}_+$ (A × 1) are computed from the SPLS-GLM. With $\mathbf{z}$, a linearized form of the link function applied to $\mathbf{y}$, being used as the response variable in the IWLS procedure for the GLM, instead of $\mathbf{y}$ (see Section 6.2), the objective function for the SPLS-GLM can be expressed as follows. For $\mathrm{a} = 1, 2, \ldots, \mathrm{A}$,

$$\begin{aligned}&\text{maximize } \operatorname{cov}\left(\mathbf{w}_{\mathrm{a}*}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{z}\mathbf{z}^{\mathrm{T}}\mathbf{X}\mathbf{w}_{\mathrm{a}*}\right) = \operatorname{cov}\left(\mathbf{t}_{\mathrm{a}*}^{\mathrm{T}}\mathbf{z}\mathbf{z}^{\mathrm{T}}\mathbf{t}_{\mathrm{a}*}\right)\\ &\text{subject to } \mathbf{w}_{\mathrm{a}*} = \mathrm{g}_{\lambda_{\mathrm{X}}}\left(\mathbf{X}^{\mathrm{T}}\mathbf{z}\mathbf{c}_{\mathrm{a}*}\right) \le \lambda_{\mathrm{X}} \text{ and } \mathbf{c}_{\mathrm{a}*} = \mathrm{g}_{\lambda_{\mathrm{Y}}}\left(\mathbf{z}^{\mathrm{T}}\mathbf{X}\mathbf{w}_{\mathrm{a}*}\right) \le \lambda_{\mathrm{Y}}\end{aligned}$$

where

$$\begin{aligned}\mathrm{g}_{\lambda_{\mathrm{X}}}\left(\mathbf{X}^{\mathrm{T}}\mathbf{z}\mathbf{c}_{\mathrm{a}*}\right) &= \operatorname{sign}\left(\mathbf{X}^{\mathrm{T}}\mathbf{z}\mathbf{c}_{\mathrm{a}*}\right)\left(\left|\mathbf{X}^{\mathrm{T}}\mathbf{z}\mathbf{c}_{\mathrm{a}*}\right| - \lambda_{\mathrm{X}}\right)_+\\ \mathrm{g}_{\lambda_{\mathrm{Y}}}\left(\mathbf{z}^{\mathrm{T}}\mathbf{X}\mathbf{w}_{\mathrm{a}*}\right) &= \operatorname{sign}\left(\mathbf{z}^{\mathrm{T}}\mathbf{X}\mathbf{w}_{\mathrm{a}*}\right)\left(\left|\mathbf{z}^{\mathrm{T}}\mathbf{X}\mathbf{w}_{\mathrm{a}*}\right| - \lambda_{\mathrm{Y}}\right)_+,\end{aligned}$$

$\mathbf{t}_{\mathrm{a}*} = \mathbf{X}\mathbf{w}_{\mathrm{a}*}$, $\mathrm{b} = 1, 2, \cdots, \mathrm{A}-1$ and $\mathrm{b} \neq \mathrm{a}$. This function is solved using the SPLS-GLM algorithm. The SPLS algorithm (Algorithm 7.1) is employed into a GLM framework, to form the SPLS-GLM algorithm, as follows. Let $\boldsymbol{\eta} = \log(\boldsymbol{\mu}) = \mathbf{X}\mathbf{b}$. The notation discussed in Chapter 1 is applicable in this section.

Algorithm 7.2:

(1) Start with an initial estimate for $\mathbf{b}$. Let $\mathbf{b}^{(0)}$ denote this estimate. Then initialize

$l = 0$

$\mathbf{V}^{(l)} = \operatorname{diag}\left\{\mathrm{v}_i^{(l)}\right\}$, for $\mathrm{i} = 1, 2, \cdots, \mathrm{N}$ and $\mathrm{v_i}$ defined in (6.10)

$\mathbf{X}_0 = \mathbf{X} - \frac{1}{\mathrm{N}}\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{V}^{(l)}\mathbf{X}$

$\mathbf{z}_0^{(l)} = \boldsymbol{\eta}^{(l)} - \frac{1}{\mathrm{N}}\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{V}^{(l)}\boldsymbol{\eta}^{(l)} = \mathbf{X}_0\mathbf{b}^{(l)} - \frac{1}{\mathrm{N}}\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{V}^{(l)}\mathbf{X}_0\mathbf{b}^{(l)}$

$\mathbf{S}_1 = \mathbf{X}_0^{\mathrm{T}}\mathbf{V}^{(l)}\mathbf{z}_0^{(l)}$.

(2) For $\mathrm{a} = 1, 2, \cdots, \mathrm{A}$, compute the SPLS parameters as

a. $\mathbf{S}_{\mathrm{a}} = \mathbf{D}\boldsymbol{\Lambda}\mathbf{F}^{\mathrm{T}}$, by the SVD

b. $\mathbf{w}_{\mathrm{a}} = \mathbf{D}[,\ 1]$

c. $\mathbf{c}_{\mathrm{a}} = \mathbf{F}[,\ 1]$

d. $\mathbf{w}_{\mathrm{a_{new}}} = \mathrm{g}_{\lambda_{\mathrm{X}}}(\mathbf{S}_{\mathrm{a}}\mathbf{c}_{\mathrm{a}}) \,/\left\|\mathrm{g}_{\lambda_{\mathrm{X}}}(\mathbf{S}_{\mathrm{a}}\mathbf{c}_{\mathrm{a}})\right\|$ and

$\mathbf{c}_{\mathrm{a_{new}}} = \mathrm{g}_{\lambda_{\mathrm{Y}}}(\mathbf{S}_{\mathrm{a}}^{\mathrm{T}}\mathbf{w}_{\mathrm{a}}) \,/\left\|\mathrm{g}_{\lambda_{\mathrm{Y}}}(\mathbf{S}_{\mathrm{a}}^{\mathrm{T}}\mathbf{w}_{\mathrm{a}})\right\|$,

where

$$\mathrm{g}_{\lambda_{\mathrm{X}}}(\mathbf{S}_{\mathrm{a}}\mathbf{c}_{\mathrm{a}}) = \operatorname{sign}(\mathbf{S}_{\mathrm{a}}\mathbf{c}_{\mathrm{a}})(|\mathbf{S}_{\mathrm{a}}\mathbf{c}_{\mathrm{a}}| - \lambda_{\mathrm{X}})_+ \text{ and}$$

$$g_{\lambda_Y}(\mathbf{S}_a^T\mathbf{w}_a) = \text{sign}(\mathbf{S}_a^T\mathbf{w}_a)(|\mathbf{S}_a^T\mathbf{w}_a| - \lambda_Y)_+$$

are the soft-thresholding penalization functions, with penalty parameters λ_X and λ_Y.

e. Check whether $\mathbf{w}_{a_{new}}$ and $\mathbf{c}_{a_{new}}$ converges. If not, return to step (e), with $\mathbf{w}_a = \mathbf{w}_{a_{new}}$ and $\mathbf{c}_a = \mathbf{c}_{a_{new}}$.

f. $\mathbf{t}_a = \mathbf{X}_{a-1}\mathbf{w}_{a_{new}} / \left\|\mathbf{X}_{a-1}\mathbf{w}_{a_{new}}\right\|$

g. $\mathbf{p}_a = \mathbf{X}_{a-1}^T\mathbf{V}^{(l)}\mathbf{X}_{a-1}\mathbf{w}_a$

h. $q_a = \mathbf{z}_{a-1}^{T\,(l)}\mathbf{V}^{(l)}\mathbf{X}_{a-1}\mathbf{w}_a$

i. $\mathbf{X}_a = \mathbf{X}_{a-1} - \mathbf{X}_{a-1}\mathbf{w}_a\mathbf{p}_a^T$

j. $\mathbf{z}_a^{(l)} = \mathbf{z}_{a-1}^{(l)} - q_a\mathbf{X}_{a-1}\mathbf{w}_a$

k. $\mathbf{S}_{a+1} = \mathbf{X}_{a-1}^T\mathbf{V}^{(l)}\mathbf{z}_{a-1}^{(l)}$.

l. Store $\mathbf{w}_{a_{new}}$ ($P \times 1$), $\mathbf{p}_a$ ($P \times 1$) and q_a into the successive columns of $\mathbf{W}_+$, $\mathbf{P}_+$ and $\mathbf{q}_+^T$.

m. Compute $\mathbf{T}_+ = \mathbf{X}_0\mathbf{R}_+$, where $\mathbf{R}_+ = \mathbf{W}_+(\mathbf{P}_+^T\mathbf{W}_+)^{-1}$

n. Get $\hat{\mathbf{b}}_{SPLS-GLM} = \mathbf{R}_+\mathbf{q}_+$

(3) Set $\mathbf{b}^{(l+1)} = \hat{\mathbf{b}}_{SPLS-GLM}$ and update $\boldsymbol{\eta}$, weight matrix $\mathbf{V}$ and $\mathbf{z}_0$ as

$$\boldsymbol{\eta}^{(l+1)} = \mathbf{X}_0\mathbf{b}^{(l+1)} + \frac{1}{N}\mathbf{1}\mathbf{1}^T\mathbf{V}^{(l)}\boldsymbol{\eta}^{(l)}$$

$$\mathbf{V}^{(l+1)} = \text{diag}\left\{v_i^{(l+1)}\right\}, \text{ for } v_i \text{ defined in (6.10)}$$

$$\mathbf{z}_0^{(l+1)} = \mathbf{X}_0\mathbf{b}^{(l+1)} + \frac{\partial\boldsymbol{\eta}}{\partial\boldsymbol{\mu}}(\mathbf{y} - \boldsymbol{\mu}).$$

(4) Check whether the change in successive estimates is sufficiently small. If not, return to step (2), with $l = l + 1$.

(5) Once convergence is reached, select the final number of components A, call this A^+, and take A^+ columns of $\mathbf{T}_+$, $\mathbf{R}_+$ and $\mathbf{q}_+^T$ to fit (7.4).

7.6 PLS biplot for Sparse Partial Least Squares

Proposing the PLS biplot as a graphical tool for displaying the SPLS of a data set, the PLS biplot can be implemented in a SPLS framework as follows. Given a pair of predictor and response samples $\mathbf{x}$ and $\mathbf{y}$, these samples will be interpolated into the biplot space using the equation $\mathbf{t}_\#^T = (\mathbf{x} - \bar{\mathbf{x}})^T\mathbf{R}_\#$. To trace the (prediction) biplot axes for the k^{th} original predictor and response variables, values of $\mu_{0_X} \in (-\infty, \infty)$ and $\mu_{0_Y} \in (-\infty, \infty)$ are substituted into the expressions $\frac{\mu_{0_X}}{\mathbf{e}_k^T\mathbf{P}_\#\mathbf{P}_\#^T\mathbf{e}_k}\mathbf{P}_\#^T\mathbf{e}_k$ and $\frac{\mu_{0_Y}}{\mathbf{e}_k^T\mathbf{Q}_\#\mathbf{Q}_\#^T\mathbf{e}_k}\mathbf{Q}_\#^T\mathbf{e}_k$ respectively. Here, calibration markers are fitted, where $\mu_X = \mu_{0_X} + \bar{x}_k$ and $\mu_Y = \mu_{0_Y} + \bar{y}_k$ are sensible scale marker values. Furthermore, the estimated SPLS coefficients matrix, defined as $\widehat{\mathbf{B}}_{SPLS} = \mathbf{R}_\#\mathbf{Q}_\#^T$, is obtained by projecting $\mathbf{e}_i^T\mathbf{R}_\#$ onto the prediction axes defined by $\mathbf{Q}_\#$.

Likewise, with $\mathbf{y}$ $(N \times 1)$, $g(\boldsymbol{\mu}) = \mathbf{T}_+\mathbf{q}_+$. For the SPLS-GLM framework, a pair of predictor and response samples $\mathbf{x}$ and $\mathbf{y}$ are both interpolated into the biplot space using the equation $\mathbf{t}_+^T = (\mathbf{x} - \bar{\mathbf{x}})^T\mathbf{R}_+$. Values of $\mu_{0_Y} \in (-\infty, \infty)$ and $\mu_{0_X} \in (-\infty, \infty)$ are substituted into the expressions $\frac{\mu_{0_Y}}{\mathbf{q}_+^T\mathbf{q}_+}\mathbf{q}_+$ and $\frac{\mu_{0_X}}{\mathbf{e}_k^T\mathbf{P}_+\mathbf{P}_+^T\mathbf{e}_k}\mathbf{P}_+^T\mathbf{e}_k$, to trace the prediction biplot axes for the original response variable and k^{th} original predictor variable respectively. The values μ_{0_X} and μ_{0_Y} are in terms of the centred samples. Therefore, calibration markers are fitted, where $\mu_X = \mu_{0_X} + \bar{x}_k$ and $\mu_Y = \mu_{0_Y} + \bar{y}$ are sensible scale marker values. In addition, the estimated SPLS-GLM coefficient vector $\hat{\mathbf{b}}_{SPLS-GLM} = \mathbf{R}_+\mathbf{q}_+$ is obtained by projecting $\mathbf{e}_i^T\mathbf{R}_+$ onto the prediction axis defined by $\mathbf{q}_+^T$.

7.7 Example

The following example is an illustration of the PLS biplot to a SPLS and a SPLS-GLM, using the cereal data from Varmuza & Filzmoser (2009) and the possum diversity data discussed in Section 6.5 respectively. The cereal data shows the infrared spectra and chemical properties measurements of fifteen cereals. There were one hundred and forty five infrared spectra taken, and six chemical properties (Heating.value, C, H, N, Starch and Ash) used. The fifteen cereals are assigned as samples, while the infrared spectra measurements and chemical properties are the predictors (**X**) and responses (**Y**) respectively. Thus, the cereal data can be viewed as a (15×151) data matrix, comprising of an $\mathbf{X}$ (15×145) matrix and a $\mathbf{Y}$ (15×6) matrix. This data can be obtained from the **chemometrics** package in R, downloaded freely from CRAN, http://cran.r-project.org/.

7.7.1 *SPLS biplot of the cereal data*

Before performing an SPLS analysis on this data, the experiment described in Subsection 7.4.1 was performed, using Algorithm 7.1. Different pairs of $\lambda_X \in (0, 100)$ and $\lambda_Y \in (0, 100)$ are used in this experiment. When pair $(\lambda_X = 0,\ \lambda_Y = 0)$ is used, it means that no thresholding is applied in the SPLS analysis. A plot of the obtained RMSEP values is shown below in Figure 7.1, along with their respective λ_X and λ_Y value. When evaluating the RMSEP for different combination of λ_X and λ_Y values (Figure 7.1) obtained from this experiment, it appears that different λ_Y does not have much of an effect, if any at all. This is not surprising, since $M = 6$. Thus, $\lambda_Y = 0$ will be used in the reminder of this example and much attention is given to the value of λ_X that gives the lowest RMSEP value. For this reason, the experiment was re-ran

using $\lambda_X \in (0, 500)$ and $\lambda_Y = 0$. A plot of the obtained RMSEP values is shown below in Figure 7.2. Since the RMSEP value, which is based on the differences between the actual and the fitted Y-values

$$\text{RMSEP} = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{M}\left(y_{ij} - \hat{y}_{ij}\right)^2 / N)}$$

(see Section 3.7), measures the variability of the differences between the **Y**- and $\hat{\mathbf{Y}}$-values, these differences have to be as small as possible, to get a 'good' prediction of **Y**. A 'good' prediction can be described as the prediction obtained when the $\hat{\mathbf{Y}}$-values are similar to those in **Y**. The smaller the RMSEP value, the better the prediction, with a zero value indicating a perfect prediction.

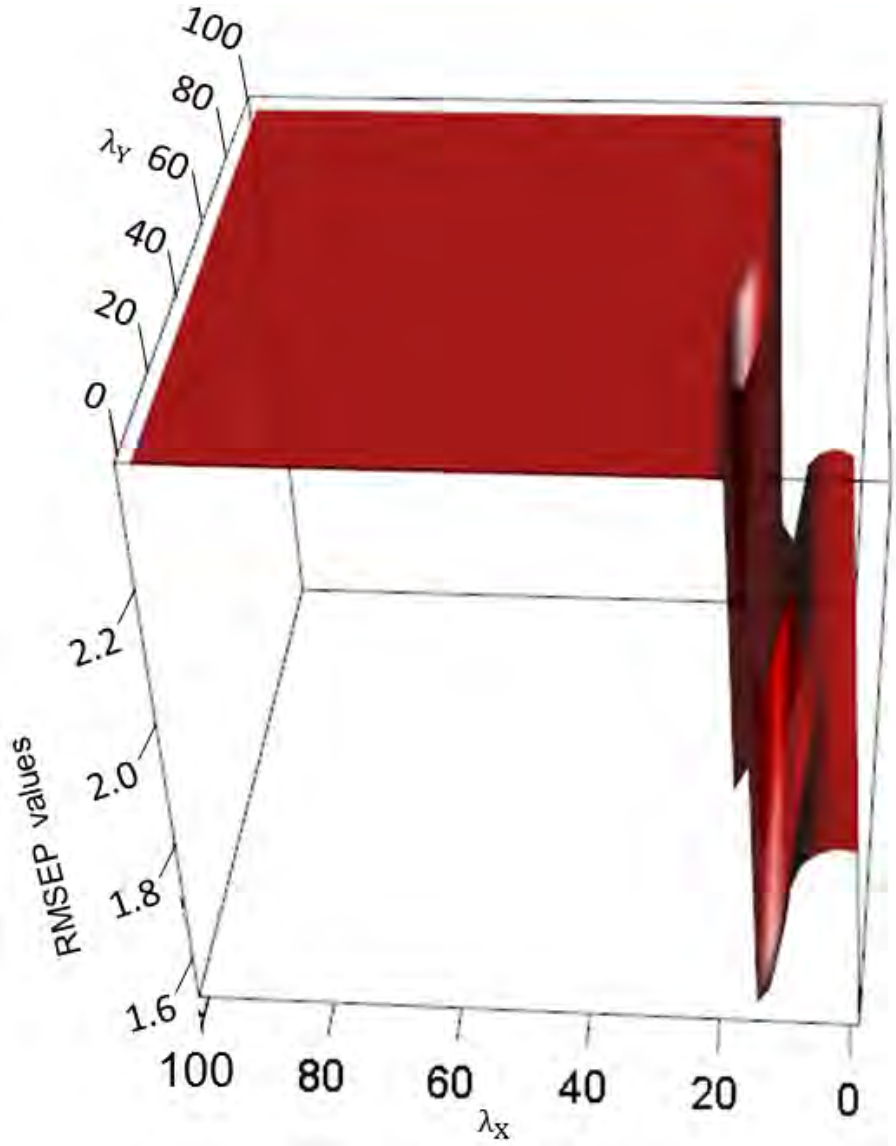

Figure 7.1 A 3D plot of $\lambda_X \in (0, 100)$, $\lambda_Y \in (0, 100)$ and their respective RMSEP value, for the cereal data.

If a smaller RMSEP value is obtained, it means that the values in $\widehat{\mathbf{Y}}$ are similar to their counterparts in $\mathbf{Y}$. However, if the values in $\widehat{\mathbf{Y}}$ are (very) different from those in $\mathbf{Y}$, the difference between $\mathbf{Y}$ and $\widehat{\mathbf{Y}}$ will be large. As a result, the RMSEP value will be large, indicating that the prediction is not a good prediction. Hence, one desires the RMSEP value to be as small as possible, to have a better prediction. The closer to 0 the value is, the better, where 0 indicates a perfect prediction. The 'worst' RMSEP value that can be obtained is when the $\widehat{\mathbf{Y}}$-values are all 0, for $\mathbf{Y} \neq 0$. In this situation, RMSEP $= \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{M}(y_{ij})^2/N}$, and has a value far from 0. An example of this can be seen in Figure 7.2 below, at $\lambda_X \geq 20$. This is because the values of the X-weights matrix $\mathbf{R}_{\#}$ are set to zero, due to the soft-thresholding penalization imposed on it. In this plot, the RMSEP value starts off at a value of 1.81, when $\lambda_X = 0$, and decreases to 1.73, at $\lambda_X = 10$. As the value for λ_X further increases, so does the RMSEP value, but stays constant after $\lambda_X = 20$. At this point, an RMSEP value of 2.37 is obtained. From Figure 7.2, the minimum RMSEP value of 1.73 is observed at $\lambda_X = 10$. Thus, for the SPLS analysis of the cereal data, $\lambda_X = 10$ and $\lambda_Y = 0$.

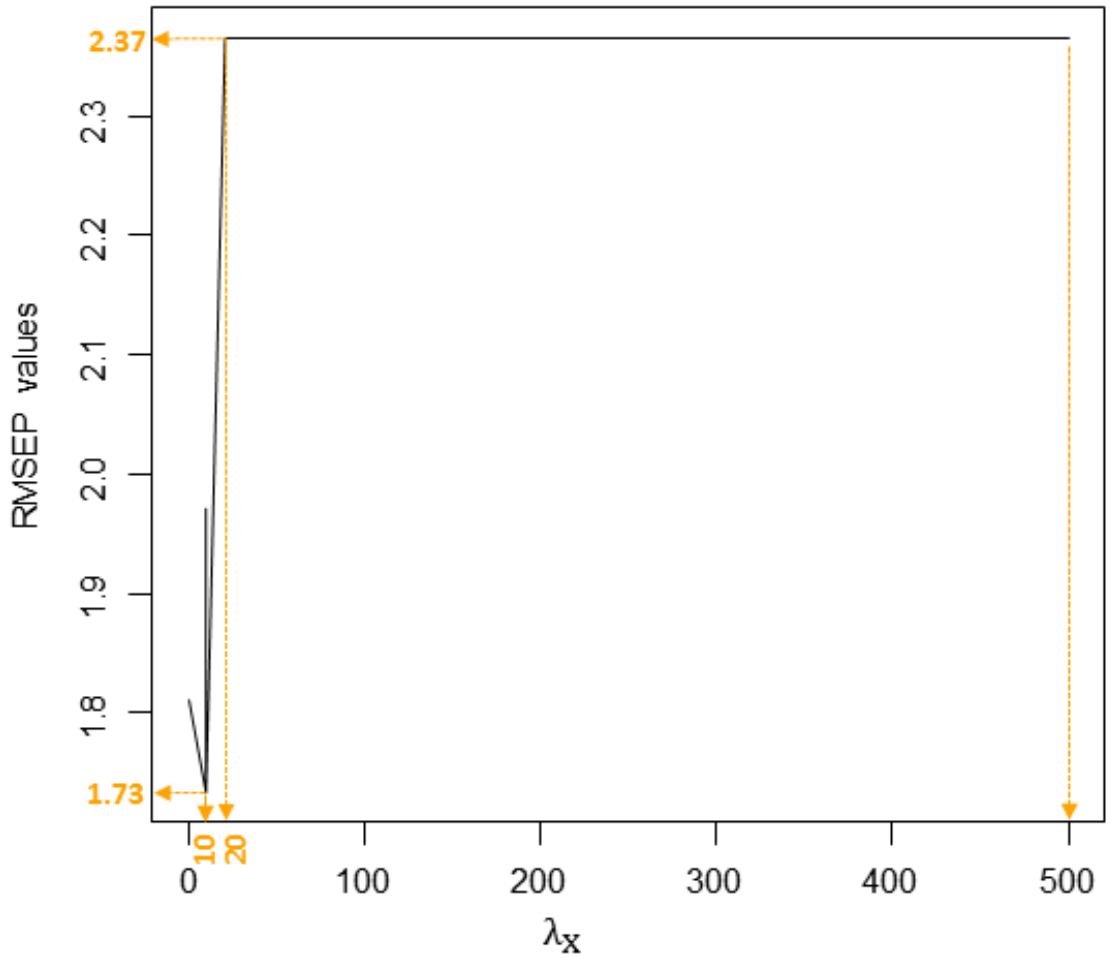

Figure 7.2 A plot of $\lambda_X \in (0, 500)$ values and their respective RMSEP value, for the cereal data.

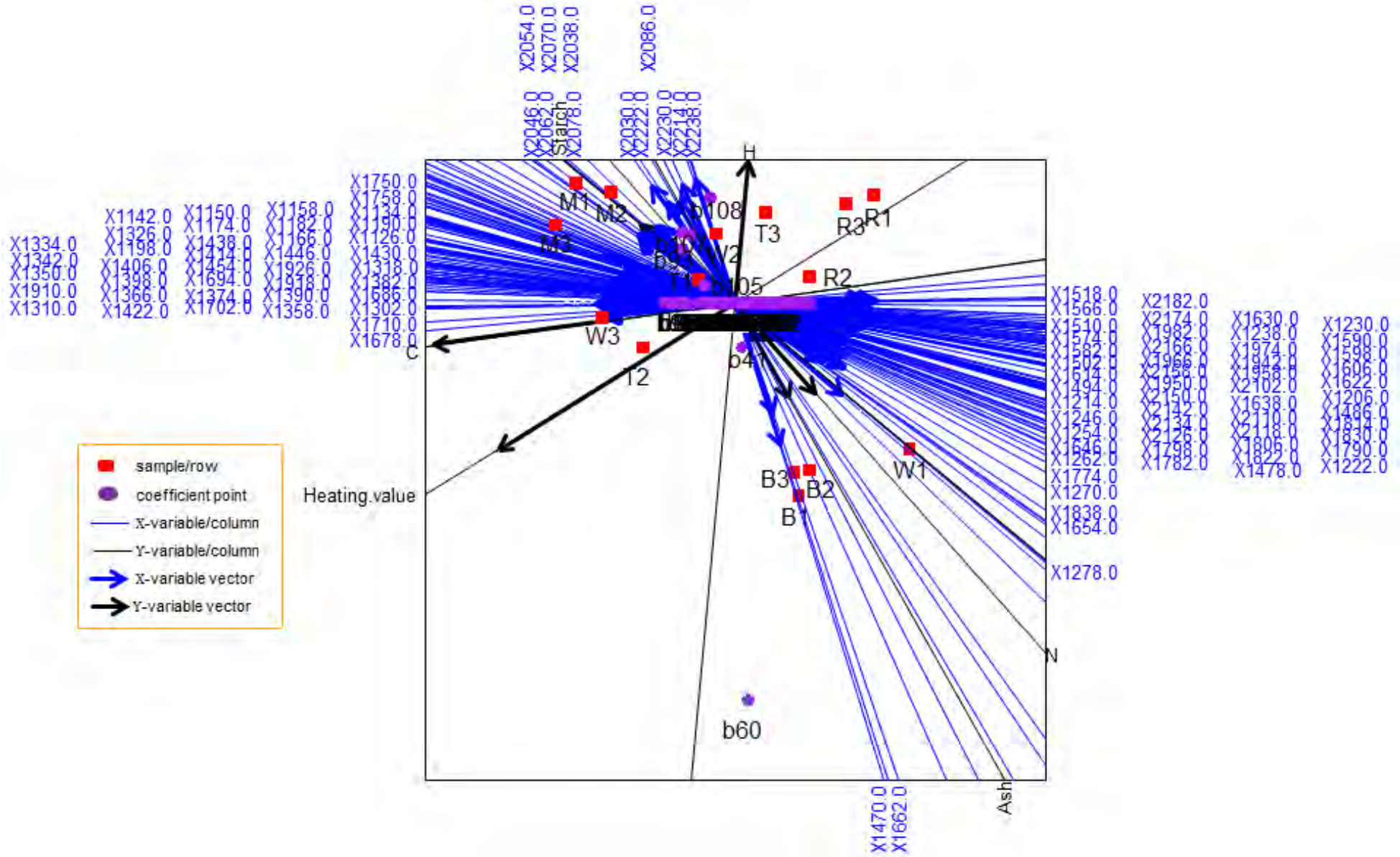

Figure 7.3 The PLS biplot for a SPLS of the cereal data (Algorithm 7.1), with $\lambda_X = 10$ and $\lambda_Y = 0$.

With $\lambda_X = 10$ and $\lambda_Y = 0$, one hundred and eight spectra (X-variables) were selected in the resulting SPLS analysis, out of the one hundred and forty five spectra used. For this reason, the number of X-variables used in the resulting biplot display (Figure 7.3) was reduced from one hundred and forty five to one hundred and eight, i.e., $\mathbf{X}\,(15 \times 108)$. Using Algorithm 7.1, with $\lambda_X = 10$ and $\lambda_Y = 0$, the resulting PLS biplot is shown above in Figure 7.3. A representation of the variance of each variable is also shown in Figure 7.3. Comparing the length of the thicker arrows (vectors) to each other, C and Heating.value can be said to have a larger standard deviation, followed by H. Axis tick markers have been excluded from Figure 7.3. The high number of selected spectra, 108, can be seen in the display. Orthogonally projecting each red point in the biplot onto the axes will give the approximated values of the cereal data. Samples B3, B1, B2 and W1 can be said to have a small standard deviation, compared to the others.

Although there is no enormous reduction in the number of X-variables (145 to 108), the resulting display (Figure 7.3) is useful in revealing the structure in the data. The (selected) spectra can be divided into three groups, namely,

group 1:

X1518.0, X2182.0, X1566.0, X2174.0, X1574.0, X1614.0, X1510.0, X2166.0, X1974.0, X1982.0, X1630.0, X1238.0, X1966.0, X1230.0, X1582.0, X1958.0, X1590.0, X2134.0, X1622.0, X2102.0, X1606.0, X2110.0, X1830.0, X1814.0, X2158.0, X1942.0, X1638.0, X1950.0, X1214.0, X1502.0, X1494.0, X2150.0, X2126.0, X2118.0, X1486.0, X1206.0, X1254.0, X1246.0, X2142.0, X1598.0, X1646.0, X1798.0, X1262.0, X1782.0, X1822.0, X1806.0, X1478.0, X1222.0, X1790.0, X1774.0, X1270.0, X1838.0 and X1654.0;

group 2:

X2046.0, X2054.0, X2062.0, X2070.0, X2038.0, X2030.0, X2078.0, X2214.0, X2230.0, X1278.0, X2086.0, X2222.0 and X2238.0;

group 3:

X1750.0, X1758.0, X1158.0, X1150.0, X1134.0, X1142.0, X1174.0, X1182.0, X1190.0, X1326.0, X1334.0, X1406.0, X1198.0, X1438.0, X1166.0, X1126.0, X1414.0, X1446.0, X1430.0, X1342.0, X1350.0, X1454.0, X1926.0, X1318.0, X1398.0, X1694.0, X1918.0, X1382.0, X1910.0, X1374.0, X1366.0, X1390.0, X1686.0, X1422.0, X1310.0, X1358.0, X1702.0, X1302.0, X1710.0 and X1678.0.

These deductions can be seen by the closeness of the axes. From Figure 7.3, group 1 can be said to be negatively related to groups 2 and 3. In addition, it is negatively related to chemical properties Starch and C, but positively related to property N. These relationship directions can be seen by the directions of the axes. It can also be deduced from the positive or negative sign infront of their respective coefficient values in Table 7.1. On the other hand, group 2 can be said to be positively related to group 3 and negatively related to group 1. With respect to the

chemical properties, group 2 can be is positively related to properties H and Starch, but negatively related to properties N and Ash. Group 3 can be said to be positively related to property C. Given their position in Figure 7.3, one can consider spectra X1470.0 and X1662.0 as outliers. Compared to the above listed spectra groups, from Figure 7.3, spectra X1470.0 and X1662.0 can be said to be positively related to group 1, but negatively related to groups 2 and 3.

Furthermore, from Figure 7.3, the SPLS coefficient points bi, for $i = 1, 2, \cdots, 108$, are projected onto the axes representing the chemical properties (in black). Using a zoomed-in display of the coefficient points, shown in Figure 7.4 below, the obtained coefficient values of the cereal data are shown in Table 7.1.

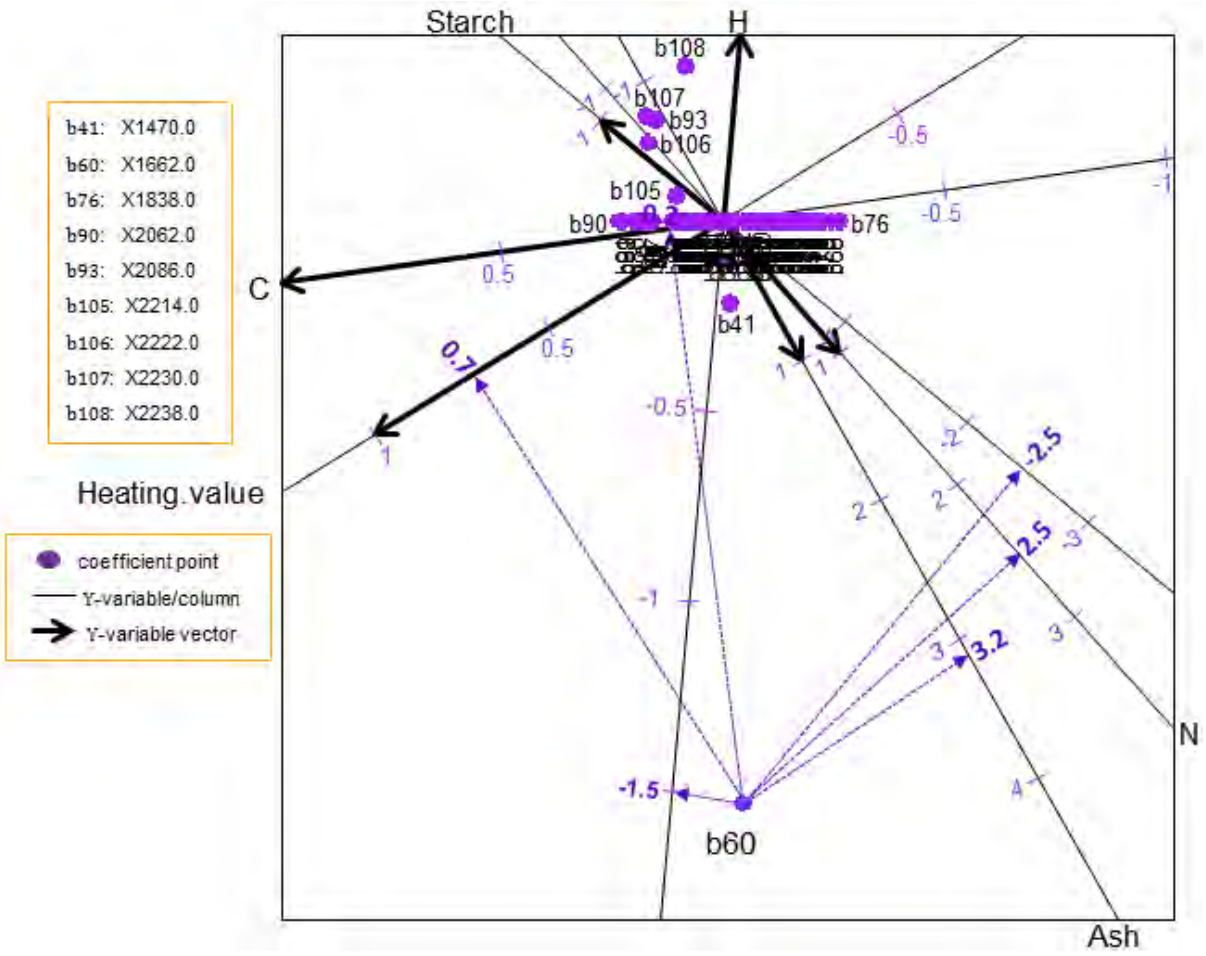

Figure 7.4 A zoomed-in display of the coefficient points in the PLS biplot for a SPLS of the cereal data, with $\lambda_X = 10$ and $\lambda_Y = 0$.

Figure 7.4 can be used for easier orthogonal projections of the coefficient points. For example, coefficient point b60 projected orthogonally onto the Heating.value, C, H, N, Starch and Ash axes yields $0.7, 0.2, -1.5, 2.5, -2.5$ and 3.2 respectively, as shown in Figure 7.4. Though only points b41, b60, b76, b90, b93, b105, b106, b107 and b108 can be seen clearly in this

display (Figure 7.4), the exact coefficient values can be obtained by printing out the coefficients matrix $\widehat{\mathbf{B}}_{\mathrm{SPLS}}$. This matrix is shown in Table 7.1 below.

Table 7. 1 The predicted coefficient values.

		Heating.value	C	H	N	Starch	Ash
b1:	**X1126.0**	0.1481	0.1578	-0.0168	-0.2673	0.3477	-0.2183
b2:	**X1134.0**	0.1531	0.1632	-0.0174	-0.2764	0.3596	-0.2258
b3:	**X1142.0**	0.1532	0.1633	-0.0174	-0.2765	0.3597	-0.2259
b4:	**X1150.0**	0.1514	0.1613	-0.0172	-0.2732	0.3554	-0.2231
b5:	**X1158.0**	0.1496	0.1594	-0.0170	-0.2700	0.3512	-0.2205
b6:	**X1166.0**	0.1493	0.1591	-0.0169	-0.2695	0.3506	-0.2201
b7:	**X1174.0**	0.1538	0.1639	-0.0174	-0.2776	0.3611	-0.2267
b8:	**X1182.0**	0.1632	0.1740	-0.0185	-0.2946	0.3833	-0.2407
b9:	**X1190.0**	0.1756	0.1871	-0.0199	-0.3169	0.4122	-0.2588
b10:	**X1198.0**	0.0709	0.0756	-0.0080	-0.1280	0.1665	-0.1045
b11:	**X1206.0**	-0.1692	-0.1804	0.0192	0.3055	-0.3974	0.2495
b12:	**X1214.0**	-0.1839	-0.1961	0.0209	0.3320	-0.4319	0.2712
b13:	**X1222.0**	-0.1886	-0.2010	0.0214	0.3404	-0.4428	0.2780
b14:	**X1230.0**	-0.1898	-0.2023	0.0215	0.3426	-0.4456	0.2798
b15:	**X1238.0**	-0.1872	-0.1996	0.0212	0.3380	-0.4396	0.2760
b16:	**X1246.0**	-0.1780	-0.1897	0.0202	0.3213	-0.4179	0.2624
b17:	**X1254.0**	-0.1598	-0.1704	0.0181	0.2885	-0.3753	0.2357
b18:	**X1262.0**	-0.1412	-0.1505	0.0160	0.2550	-0.3317	0.2082
b19:	**X1270.0**	-0.1380	-0.1471	0.0156	0.2491	-0.3240	0.2035
b20:	**X1278.0**	-0.1157	-0.1233	0.0131	0.2088	-0.2717	0.1706
b21:	**X1302.0**	0.0026	0.0027	-0.0003	-0.0046	0.0060	-0.0038
b22:	**X1310.0**	0.0542	0.0578	-0.0061	-0.0979	0.1273	-0.0800
b23:	**X1318.0**	0.0762	0.0812	-0.0086	-0.1375	0.1788	-0.1123
b24:	**X1326.0**	0.0851	0.0907	-0.0096	-0.1537	0.1999	-0.1255
b25:	**X1334.0**	0.0884	0.0942	-0.0100	-0.1595	0.2075	-0.1303
b26:	**X1342.0**	0.0898	0.0957	-0.0102	-0.1621	0.2109	-0.1324
b27:	**X1350.0**	0.0907	0.0967	-0.0103	-0.1638	0.2131	-0.1338
b28:	**X1358.0**	0.0882	0.0940	-0.0100	-0.1592	0.2071	-0.1300
b29:	**X1366.0**	0.0800	0.0853	-0.0091	-0.1444	0.1879	-0.1180
b30:	**X1374.0**	0.0684	0.0729	-0.0078	-0.1234	0.1605	-0.1008
b31:	**X1382.0**	0.0600	0.0640	-0.0068	-0.1083	0.1409	-0.0885
b32:	**X1390.0**	0.0574	0.0612	-0.0065	-0.1036	0.1347	-0.0846
b33:	**X1398.0**	0.0596	0.0635	-0.0068	-0.1076	0.1400	-0.0879
b34:	**X1406.0**	0.0661	0.0705	-0.0075	-0.1194	0.1553	-0.0975
b35:	**X1414.0**	0.0745	0.0794	-0.0084	-0.1345	0.1750	-0.1099
b36:	**X1422.0**	0.0810	0.0864	-0.0092	-0.1463	0.1903	-0.1195
b37:	**X1430.0**	0.0835	0.0890	-0.0095	-0.1507	0.1960	-0.1231
b38:	**X1438.0**	0.0798	0.0850	-0.0090	-0.1440	0.1874	-0.1176
b39:	**X1446.0**	0.0656	0.0699	-0.0074	-0.1184	0.1540	-0.0967

Table 7.1 Cont[D].

		Heating.value	C	H	N	Starch	Ash
b40:	**X1454.0**	0.0289	0.0308	-0.0033	-0.0521	0.0678	-0.0425
b41:	**X1470.0**	0.0539	0.0001	-0.1396	0.2590	-0.2563	0.3189
b42:	**X1478.0**	-0.1469	-0.1565	0.0166	0.2651	-0.3449	0.2165
b43:	**X1486.0**	-0.1162	-0.1238	0.0132	0.2097	-0.2728	0.1713
b44:	**X1494.0**	-0.0948	-0.1010	0.0107	0.1711	-0.2226	0.1398
b45:	**X1502.0**	-0.0722	-0.0770	0.0082	0.1303	-0.1695	0.1064
b46:	**X1510.0**	-0.0444	-0.0473	0.0050	0.0801	-0.1042	0.0654
b47:	**X1518.0**	-0.0091	-0.0097	0.0010	0.0164	-0.0213	0.0134
b48:	**X1566.0**	-0.0114	-0.0121	0.0013	0.0205	-0.0267	0.0168
b49:	**X1574.0**	-0.0542	-0.0578	0.0061	0.0978	-0.1272	0.0799
b50:	**X1582.0**	-0.0789	-0.0841	0.0089	0.1425	-0.1854	0.1164
b51:	**X1590.0**	-0.0951	-0.1013	0.0108	0.1716	-0.2232	0.1402
b52:	**X1598.0**	-0.1053	-0.1123	0.0119	0.1902	-0.2474	0.1553
b53:	**X1606.0**	-0.1128	-0.1202	0.0128	0.2035	-0.2648	0.1662
b54:	**X1614.0**	-0.1195	-0.1273	0.0135	0.2157	-0.2805	0.1761
b55:	**X1622.0**	-0.1259	-0.1342	0.0143	0.2273	-0.2956	0.1856
b56:	**X1630.0**	-0.1342	-0.1431	0.0152	0.2423	-0.3152	0.1979
b57:	**X1638.0**	-0.1445	-0.1541	0.0164	0.2609	-0.3394	0.2131
b58:	**X1646.0**	-0.1591	-0.1696	0.0180	0.2872	-0.3736	0.2346
b59:	**X1654.0**	-0.1770	-0.1887	0.0201	0.3195	-0.4156	0.2610
b60:	**X1662.0**	0.7100	0.1412	-1.5149	2.5454	-2.4450	3.2320
b61:	**X1678.0**	0.0458	0.0488	-0.0052	-0.0827	0.1076	-0.0676
b62:	**X1686.0**	0.0941	0.1003	-0.0107	-0.1698	0.2209	-0.1387
b63:	**X1694.0**	0.1062	0.1132	-0.0120	-0.1917	0.2493	-0.1566
b64:	**X1702.0**	0.0922	0.0983	-0.0105	-0.1665	0.2165	-0.1360
b65:	**X1710.0**	0.0310	0.0330	-0.0035	-0.0560	0.0728	-0.0457
b66:	**X1750.0**	0.0463	0.0494	-0.0053	-0.0836	0.1088	-0.0683
b67:	**X1758.0**	0.0266	0.0284	-0.0030	-0.0481	0.0626	-0.0393
b68:	**X1774.0**	-0.2255	-0.2404	0.0256	0.4071	-0.5296	0.3325
b69:	**X1782.0**	-0.2240	-0.2387	0.0254	0.4043	-0.5259	0.3302
b70:	**X1790.0**	-0.2076	-0.2213	0.0235	0.3748	-0.4876	0.3061
b71:	**X1798.0**	-0.1931	-0.2058	0.0219	0.3485	-0.4534	0.2847
b72:	**X1806.0**	-0.1809	-0.1928	0.0205	0.3265	-0.4248	0.2667
b73:	**X1814.0**	-0.1714	-0.1827	0.0194	0.3093	-0.4024	0.2526
b74:	**X1822.0**	-0.1717	-0.1830	0.0195	0.3099	-0.4031	0.2531
b75:	**X1830.0**	-0.1938	-0.2065	0.0220	0.3498	-0.4550	0.2857
b76:	**X1838.0**	-0.2433	-0.2593	0.0276	0.4392	-0.5713	0.3587
b77:	**X1910.0**	0.0021	0.0022	-0.0002	-0.0038	0.0049	-0.0031
b78:	**X1918.0**	0.0052	0.0056	-0.0006	-0.0094	0.0123	-0.0077
b79:	**X1926.0**	0.0020	0.0022	-0.0002	-0.0037	0.0048	-0.0030
b80:	**X1942.0**	-0.0836	-0.0891	0.0095	0.1509	-0.1963	0.1232
b81:	**X1950.0**	-0.0844	-0.0899	0.0096	0.1523	-0.1981	0.1244
b82:	**X1958.0**	-0.0792	-0.0844	0.0090	0.1430	-0.1860	0.1168
b83:	**X1966.0**	-0.0679	-0.0724	0.0077	0.1226	-0.1595	0.1002
b84:	**X1974.0**	-0.0512	-0.0545	0.0058	0.0924	-0.1202	0.0754

Table 7.1 Cont[D].

		Heating.value	C	H	N	Starch	Ash
b85:	**X1982.0**	-0.0254	-0.0270	0.0029	0.0458	-0.0596	0.0374
b86:	**X2030.0**	0.0676	0.0721	-0.0077	-0.1220	0.1587	-0.0997
b87:	**X2038.0**	0.1477	0.1574	-0.0167	-0.2665	0.3467	-0.2177
b88:	**X2046.0**	0.1868	0.1991	-0.0212	-0.3372	0.4386	-0.2754
b89:	**X2054.0**	0.2074	0.2210	-0.0235	-0.3743	0.4869	-0.3057
b90:	**X2062.0**	0.2155	0.2297	-0.0244	-0.3889	0.5059	-0.3177
b91:	**X2070.0**	0.2138	0.2279	-0.0242	-0.3859	0.5020	-0.3152
b92:	**X2078.0**	0.2098	0.2236	-0.0238	-0.3787	0.4926	-0.3093
b93:	**X2086.0**	0.0263	0.1381	0.2535	-0.7317	0.7969	-0.8038
b94:	**X2102.0**	-0.0595	-0.0634	0.0067	0.1074	-0.1397	0.0877
b95:	**X2110.0**	-0.1187	-0.1266	0.0135	0.2143	-0.2788	0.1751
b96:	**X2118.0**	-0.1342	-0.1431	0.0152	0.2423	-0.3152	0.1979
b97:	**X2126.0**	-0.1416	-0.1509	0.0160	0.2556	-0.3324	0.2087
b98:	**X2134.0**	-0.1405	-0.1497	0.0159	0.2536	-0.3298	0.2071
b99:	**X2142.0**	-0.1325	-0.1412	0.0150	0.2391	-0.3110	0.1953
b100:	**X2150.0**	-0.1180	-0.1257	0.0134	0.2130	-0.2770	0.1739
b101:	**X2158.0**	-0.0998	-0.1063	0.0113	0.1801	-0.2342	0.1471
b102:	**X2166.0**	-0.0730	-0.0778	0.0083	0.1318	-0.1715	0.1077
b103:	**X2174.0**	-0.0427	-0.0455	0.0048	0.0770	-0.1001	0.0629
b104:	**X2182.0**	-0.0152	-0.0162	0.0017	0.0274	-0.0356	0.0224
b105:	**X2214.0**	0.0622	0.0930	0.0550	-0.2779	0.3240	-0.2769
b106:	**X2222.0**	0.0580	0.1445	0.1861	-0.6187	0.6884	-0.6602
b107:	**X2230.0**	0.0089	0.1155	0.2459	-0.6750	0.7289	-0.7498
b108:	**X2238.0**	-0.1198	0.0351	0.3927	-0.7953	0.8056	-0.9543

7.7.2 *SPLS biplot of the possum diversity data*

Consider the possum diversity data discussed in Section 6.5. Here, the Diversity variable is assigned as the response variable, while the remaining variables are assigned as the predictors. That is, $\mathbf{X}\,(151 \times 13)$ and $\mathbf{y}\,(151 \times 1)$. With only $\mathrm{M} = 1$ Y-variable, as discussed in Subsection 7.4.1, there is no need to perform variables selection on the Y-variable. As a result, $\lambda_Y = 0$ throughout this example and much attention is given to the value of λ_X that gives the lowest RMSEP value in the experiment described in Subsection 7.4.1. This experiment is performed using Algorithm 7.2. Since the Diversity variable is a count variable, the SPLS-GLM analyses performed in this experiment are Poisson-fittted, where $\boldsymbol{\eta} = \log(\boldsymbol{\mu}) = \mathbf{Xb}$ and $\boldsymbol{\mu} = \mathrm{e}^{\mathbf{Xb}}$. Figure 7.5 shows a plot of the obtained RMSEP values along with their respective λ_X value. Here, $\lambda_X \in (0, 48)$ and $\lambda_Y = 0$. From this plot, one can see that the minimum RMSEP value is found when $1 \leq \lambda_X < 4$.

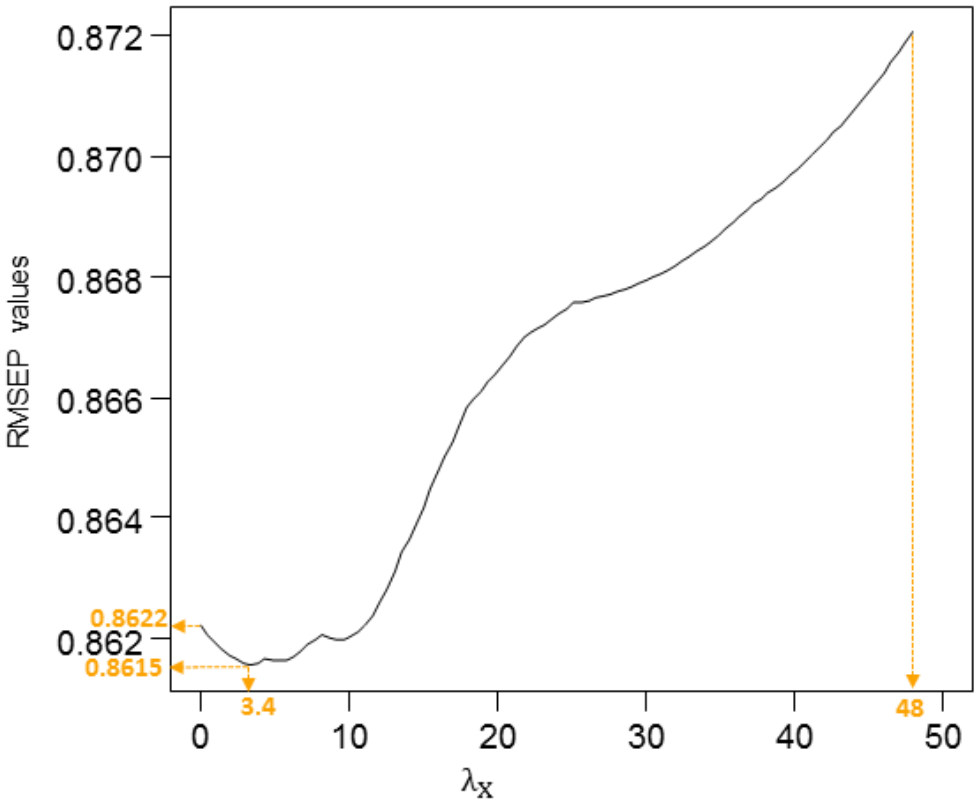

Figure 7.5 A plot of $\lambda_X \in (0, 48)$ values and their respective RMSEP value, for the possum diversity data.

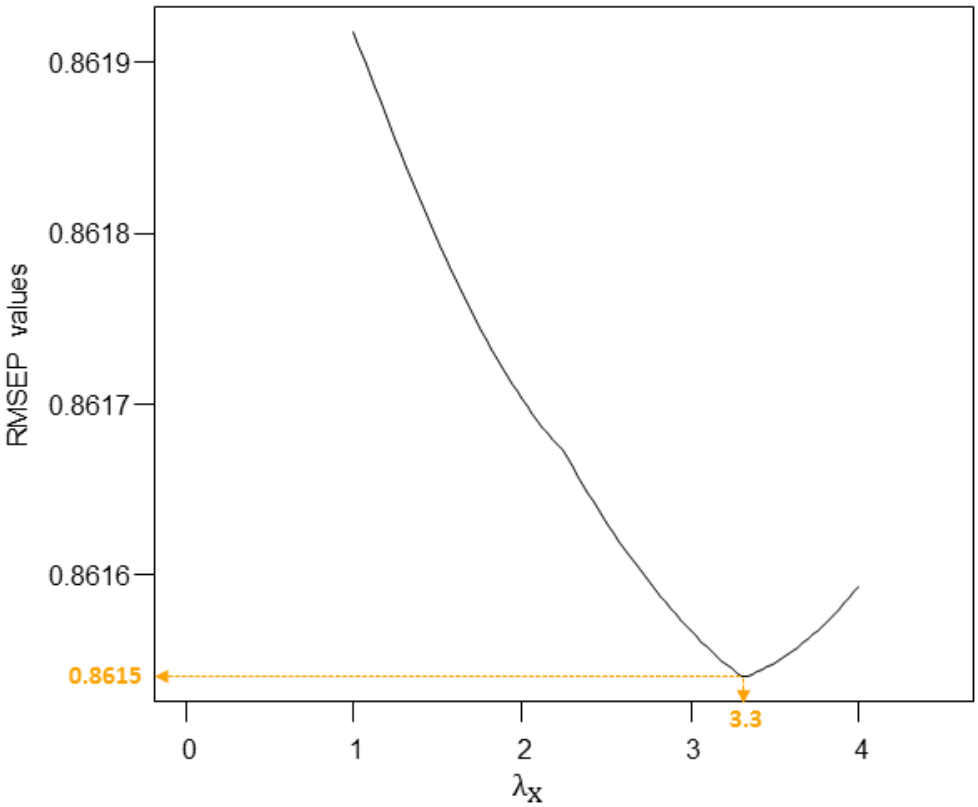

Figure 7.6 A plot of $\lambda_X \in (1, 4)$ values and their respective RMSEP value, for the possum diversity data.

Figure 7.6 above shows a plot of the RMSEP values obtained, after the experiment was re-ran using $\lambda_X \in (1, 4)$ and $\lambda_Y = 0$. In this plot, the minimum RMSEP value of 0.8615 is observed at $\lambda_X = 3.3$. Thus, for the SPLS-GLM analysis of the possum diversity data, $\lambda_X = 3.3$ and $\lambda_Y = 0$. In the resulting SPLS-GLM analysis (with $\lambda_X = 3.3$ and $\lambda_Y = 0$), no values in the X-weights matrix $\mathbf{R}_+$ was set to zero. This is not surprising, seeing as the number of X-variables ($P = 13$) can be considered as small. Using these penalty parameters ($\lambda_X = 3.3$ and $\lambda_Y = 0$) and Algorithm 7.2, the resulting PLS biplot is shown in Figure 7.7 below. Another display of Figure 7.7, where the sample point names have been excluded is shown in Figure 7.8.

Orthogonally projecting each of the coefficient points bi, for $i = 1, 2, \cdots, 13$, in Figure 7.7 onto the Diversity axis yields the coefficient values. As discussed in Subsection 5.2.3 and Section 7.7, *the purple markers on the Diversity axis* are used to read off these values. A zoomed-in display of the coefficient points, shown in Figure 7.9 below, can be used for easier orthogonal projections. For example, coefficient points b2 and b3 projected orthogonally onto the Diversity axis yields -0.06 and 0.24 respectively, as shown below in Figure 7.9. The obtained coefficient values of the possum diversity data are shown in Table 7.2. With their respective sign denoting their respective effect direction on Diversity, all the variables can be said to have a positive effect on Diversity, except for Stumps, E.delegatensis, E.regnans, NW-NE and SW-NW. Variable Stags can be said to have a high effect on Diversity, followed by Habitat, Bark, SW-NW and BAcacia. However, the other variables can be seen to have a low effect, with E.regnans and NW-NE having the lowest.

Table 7.2 The predicted coefficient values.

		Diversity
b1:	**Shrubs**	0.037
b2:	**Stumps**	-0.063
b3:	**Stags**	0.233
b4:	**Bark**	0.152
b5:	**Habitat**	0.213
b6:	**BAcacia**	0.107
b7:	**E.regnans**	-0.009
b8:	**E.delegatensis**	-0.016
b9:	**E.nitens**	0.037
b10:	**NW-NE**	-0.006
b11:	**NW-SE**	0.040
b12:	**SE-SW**	0.062
b13:	**SW-NW**	-0.142

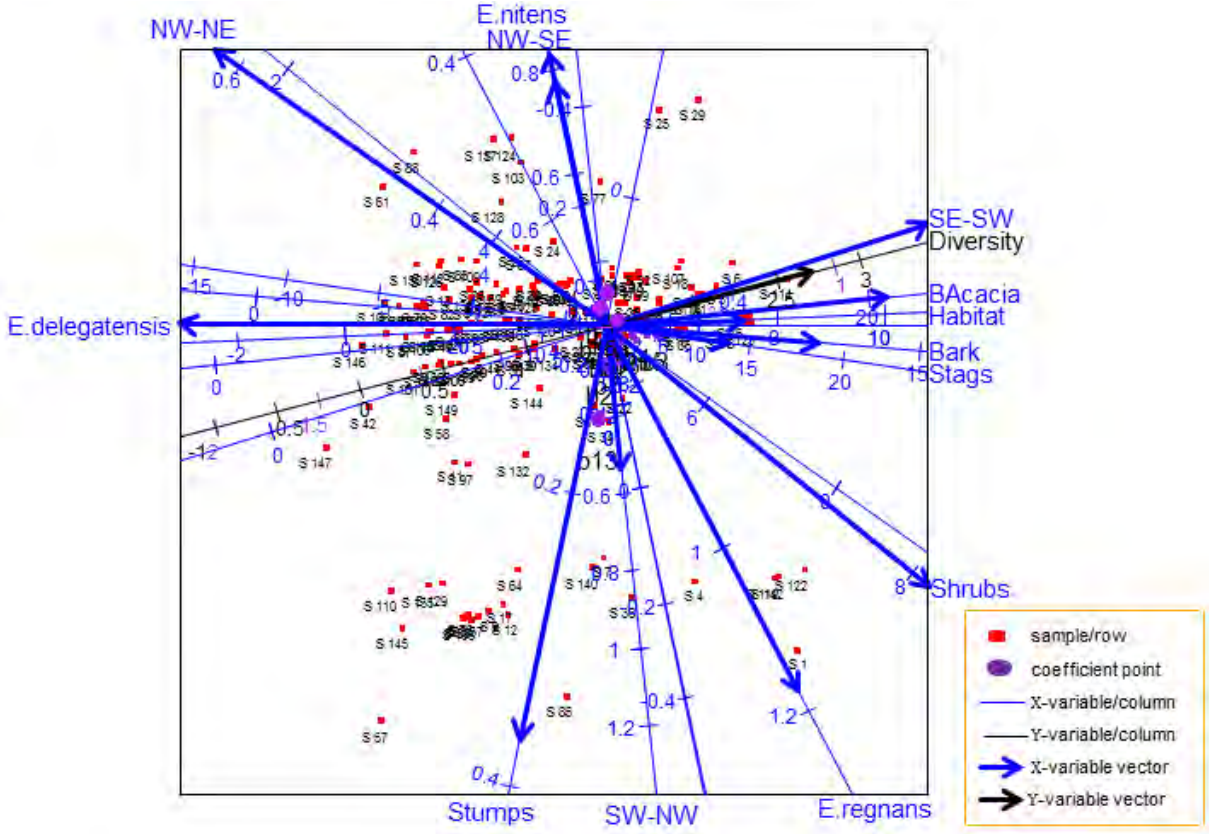

Figure 7.7 The PLS biplot of a Poisson SPLS-GLM of the possum diversity data (Algorithm 7.2), with $\lambda_X = 3.3$ and $\lambda_Y = 0$.

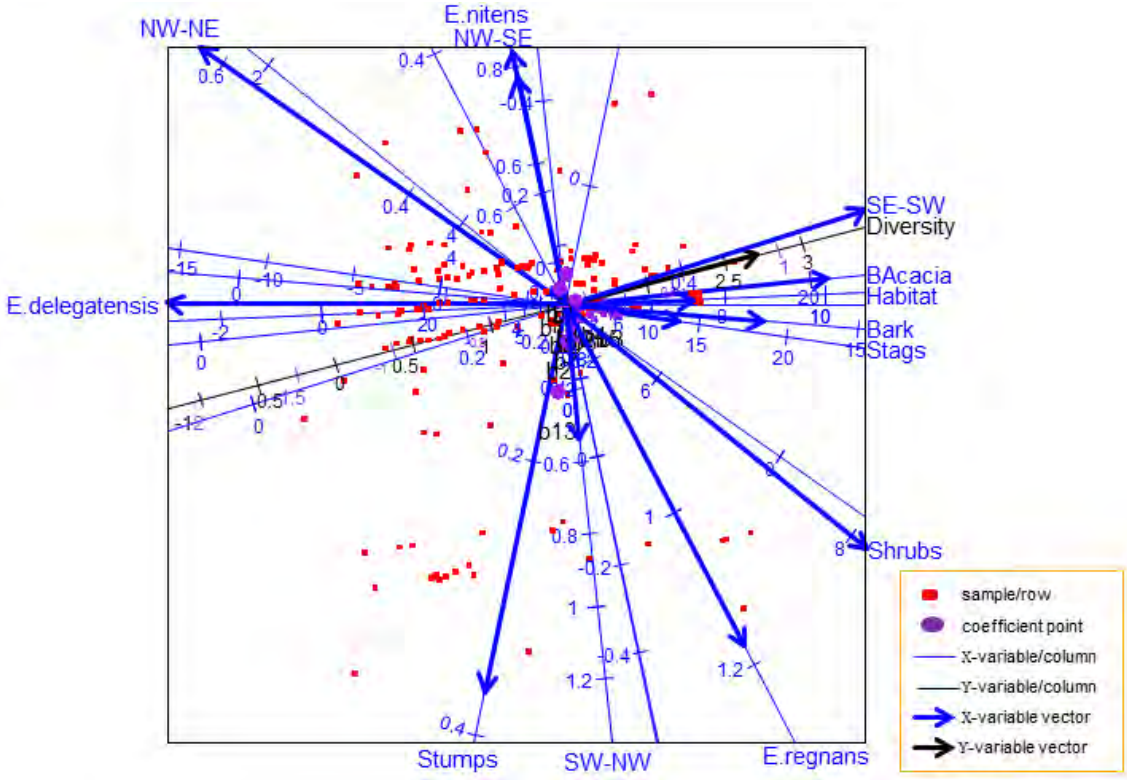

Figure 7.8 The PLS biplot of a Poisson SPLS-GLM of the possum diversity data (Algorithm 7.2), with no sample point names, for $\lambda_X = 3.3$ and $\lambda_Y = 0$.

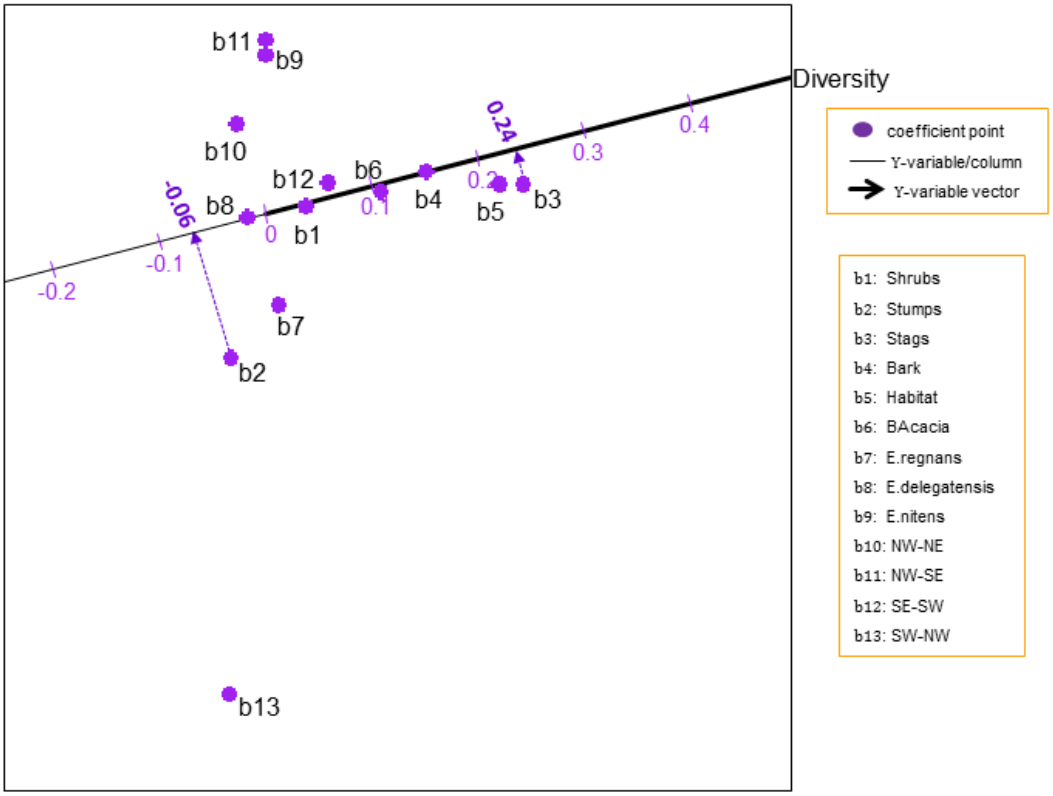

Figure 7.9 A zoomed-in display of the coefficient points in the PLS biplot for a SPLS-GLM of the possum diversity data, with $\lambda_X = 3.3$ and $\lambda_Y = 0$.

Furthermore, from Figure 7.7, the relation between Stags, Bark, E.delegatensis, Habitat, BAcacia, Diversity and SE-SW; between Shrubs and NW-NE, as well as between SW-NW, E.nitens and NW-SE can be seen. These deductions are similar to those observed from the PLS-GLM biplot of this data (Figure 6.1). For example, the relation between Diversity, BAcacia, SE-SW, Habitat, Bark and Stags can be seen in Figure 6.1. This can also be seen in Figure 7.7. Also, the relation between E.nitens, NW-SE and SW-NW can be seen in both biplots. However, the positions of the coefficient points are slightly different in both biplots. See Figures 6.2 and 7.9, for clearer displays. This difference is due to the way the X-weights matrix, $\mathbf{R}_*$ in PLS-GLM and $\mathbf{R}_+$ in SPLS-GLM, was computed in their respective algorithms (see Algorithms 6.2 and 7.2). The estimated coefficient values from the SPLS-GLM analysis (Figure 7.7) are shown in Table 7.2, while the coefficient values obtained from the PLS-GLM analysis (Figure 6.1) are shown in Table 6.1 (see Chapter 6). Although the same data set was used in the PLS-GLM and SPLS-GLM analyses, one can notice the different biplot displays obtained as a result of the different GLM framework used. Hence, *different PLS-GLM frameworks can result in different biplot displays*. For the possum diversity data, there is no detriment in choosing Figure 6.1 over Figure 7.7. If the minimum RMSEP value (0.8615) occurred when $\lambda_X = 0$

and $\lambda_Y = 0$, then one can conclude that the PLS-GLM analysis is more suitable for the data, than the SPLS-GLM analysis. However, this is not the case here, as the minimum RMSEP value (0.8615) occurred when $\lambda_X = 3.3 \neq 0$, indicating that an SPLS-GLM analysis is still suitable for this data. An SPLS-GLM analysis is recommended for a (very) large data set, like the colon data (see Chapter 8).

7.8 Summary

With large data sets, PLS, SPLS and SPLS-GLM can be useful tools for analysis. Results found by the SPLS and/or SPLS-GLM of a data set can be visualized graphically using the PLS biplot. With different PLS and PLS-GLM frameworks, comes different PLS biplot displays.

If some non-linear relationship exists between the X- and Y-variables, the PLS-GLM or SPLS-GLM analysis should be used, else, use the PLS or SPLS analysis, if it is a linear relationship.

When the number of predictor variables and response variables are small, performing variables selection on these variables is unnecessary. As a result, the penalty parameters $\lambda_X = 0$ and $\lambda_Y = 0$, for the SPLS and/or SPLS-GLM analysis of the data. In addition, when this happens ($\lambda_X = 0$ and $\lambda_Y = 0$), it indicates that a standard PLS (PLS-GLM) analysis is more suitable for the data, than the SPLS (SPLS-GLM) analysis. Furthermore, if (i) $\lambda_Y = 0$ and $\lambda_X \neq 0$, or (ii) $\lambda_X = 0$ and $\lambda_Y \neq 0$, or (iii) $\lambda_X \neq 0$ and $\lambda_Y \neq 0$, all these indicate that an SPLS or SPLS-GLM analysis is suitable for the data under consideration. In these situations, there is no detriment in choosing the PLS over the SPLS analysis and/or the PLS-GLM over the SPLS-GLM analysis. However, SPLS and SPLS-GLM are useful for (very) large data sets.

Next, in Chapter 8, a detailed discussion of three different applications of the PLS biplot is given.

CHAPTER 8

APPLICATION OF THE PLS BIPLOT TO THREE DATA SETS

8.1 Introduction

In this chapter, the developed PLS biplot is applied to four different data sets, namely, the SOVR data from Umetrics MKS (2013), the Pima.tr data from Smith *et al.* (1988) and the colon data from Alon *et al.* (1999). A copy of the SOVR data set can be found on the dropbox link

https://www.dropbox.com/sh/wr66u07t1vjm9da/AACg_E4h8MvgOHuCXk69yDIya

under the "*Data Sets*" folder. The Pima.tr and colon data sets can be obtained from the **MASS** and **plsgenomics** packages respectively in R, downloaded freely from CRAN, http://cran.r-project.org/. As is customary for PLS, both the predictor and response matrices of each data set is centred before the analysis. In addition, each of these data matrices is standardized by dividing each centred variable by their respectively standard deviation, as this facilitates the direct comparison of regression coefficients. In each of the resulting PLS biplots, the blue, black and purple calibration markers on the axes are fitted using sensible scale values

$$\mu_X = (\mu_{0_X} + \bar{x}_k)s_{X_k},\ \mu_Y = (\mu_{0_Y} + \bar{y}_k)s_{Y_k} \text{ and } \mu_b = \mu_{0_Y}$$

respectively.

Furthermore, the use of the developed PLS biplot software (**PLSbiplot1**) is illustrated here, in Section 8.5, using the ash and glass data from Varmuza & Filzmoser (2009), the cocktail data from Husson *et al*. (2013), the nutrimouse data by Martin *et al*. (2007) and the spider data from Van der Aart & Smeenk-Enserink (1975). The ash and glass data can be obtained from the **chemometrics** package in R, while the spider data can be obtained from the **mvabund** package. Likewise, the cocktail data can be obtained from the **SensoMineR** package, while the nutrimouse data can be obtained from the **mixOmics** package. These packages can be downloaded freely from CRAN, http://cran.r-project.org/. More details are given in Subsection 8.2.5.

8.2 The PLS biplot of the SOVR data

This mineral sorting production data shows the quality evaluation of five hundred and seventy two processes used to produce a final product. Twelve process factors were used in the evaluation, namely, total load (TON_IN), load of grinder 30 (KR30_IN), load of grinder 40 (KR40_IN), concentration mull (PARM), velocity of separator 1 (HS_1), velocity of separator 2

(HS_2), effect of grinder 30 (PKR_30), effect of grinder 40 (PKR_40), ore waste (GBA), load of separator 3 (TON_S3), waste from grinding (KRAV_F) and total waste (TOTAVF). The aim of this evaluation was to investigate the relationships between the process factors and the quality of the final product. Six output variables, amount of concentration type 1 (PAR), amount of concentration type 2 (FAR), distribution of concentration type 1 and 2 (r-FAR), iron in FAR (Percent_Fe_FAR), phosphor in FAR (Percent_P_FAR) and iron in raw ore (Percent_Fe_malm), were used to measure the quality of the final product. The processes are assigned as the samples, while the process factors and output variables are the predictor and response variables respectively. Thus, the SOVR data can be viewed as a (572×18) data matrix, comprising of an $\mathbf{X}$ (572×12) matrix and a $\mathbf{Y}$ (572×6) matrix. A copy of this data can be found on the dropbox link

https://www.dropbox.com/sh/wr66u07t1vjm9da/AACg_E4h8MvgOHuCXk69yDIya

under the "*Data Sets*" folder.

8.2.1 *Number of PLS components*

A 12-component PLS was performed using the SIMPLS algorithm, but after inspecting the RMSEP plot in Figure 8.1 below (see Subsection 3.7.2), three components can be suggested as the final number of PLS components to use in the modelling of $\mathbf{Y}$.

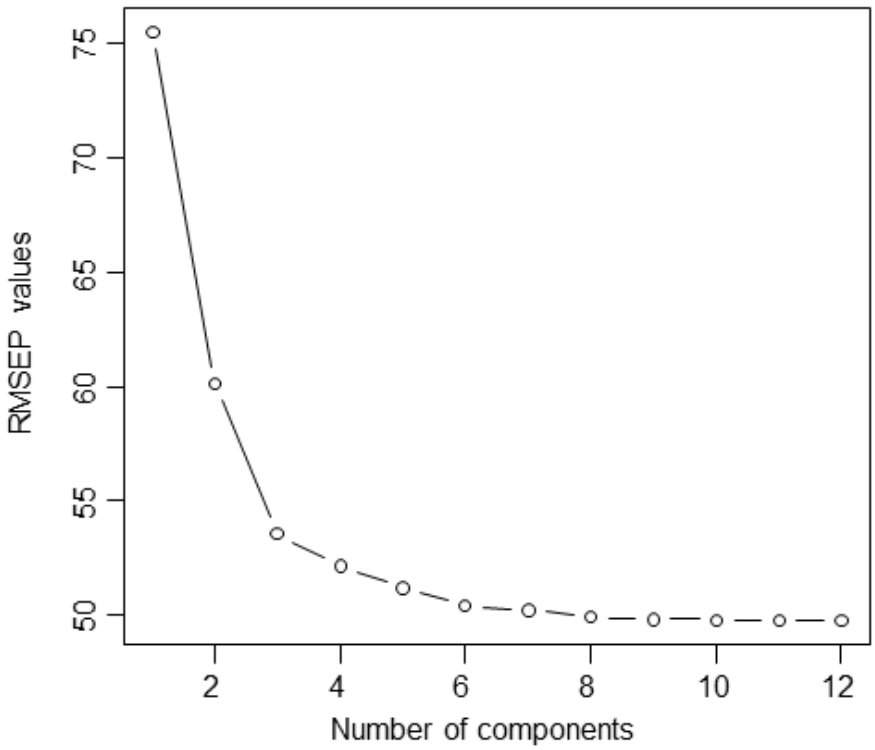

Figure 8.1 The RMSEP plot of the SOVR data.

8.2.2 *Important variables*

Using the three components, the unimportant process factors are identified based on their VIP values. The obtained VIP value for each predictor variable is shown below in Table 8.1. After inspecting these VIP values, process factors KR40_IN, PKR_40 and KRAV_F were identified as unimportant factors for this analysis. For this reason, the number of predictor variables used in the (final) PLS biplot was reduced to nine, i.e., $\mathbf{X}\,(572 \times 9)$.

Table 8.1 VIP values of the process factors.

TON_IN	KR30_IN	KR40_IN	PARM	HS_1	HS_2
0.797	0.822	0.747*	1.130	1.563	1.531

PKR_30	PKR_40	GBA	TON_S3	KRAV_F	TOTAVF
0.789	0.686*	1.130	0.760	0.683*	0.809

* VIP $\geq$ 0.8 indicates important predictor variables.

8.2.3 *PLS biplot of the SOVR data*

Although three components was suggested as the number of components to use for the modelling of the Y-variables of this data, for the PLS biplot (Chapter 5), only two components are represented. While not discussed in this dissertation, using the newly reduced predictor variables $\mathbf{X}\,(572 \times 9)$, if a three-dimensional PLS biplot was used on this data, by (5.16), the resulting biplot would have an overall quality value of 0.995. The two-dimensional PLS biplot (Chapter 5) is shown in Figure 8.2 below. This display (Figure 8.2) has an overall quality value of 0.992, which is virtually perfect and as good as using three components.

A zoomed-in display of the samples, with a representation of the variance of each variable, is shown in Figure 8.3. Observing the length of the thicker arrows (vectors) on the black axes, output variable Percent_Fe_malm can be said to have the smallest standard deviation, compared to the others. Likewise, comparing the length of the blue vectors, factor HS_1 has the smallest deviation. In addition, observing the angles between the blue vectors, all the (newly reduced) predictor variables can be said to be positively related to each other. Similarly, observing the angles between the black vectors, all the response variables can be said to be positively related, except for Percent_Fe_FAR.

With an overall quality of 0.992, various inter-variable relationships can be deduced from Figure 8.2, such as the relation between output variable PAR and factors TOTAVF and GBA. Looking at the directions of these axes in the biplot, this relation is a positive one.

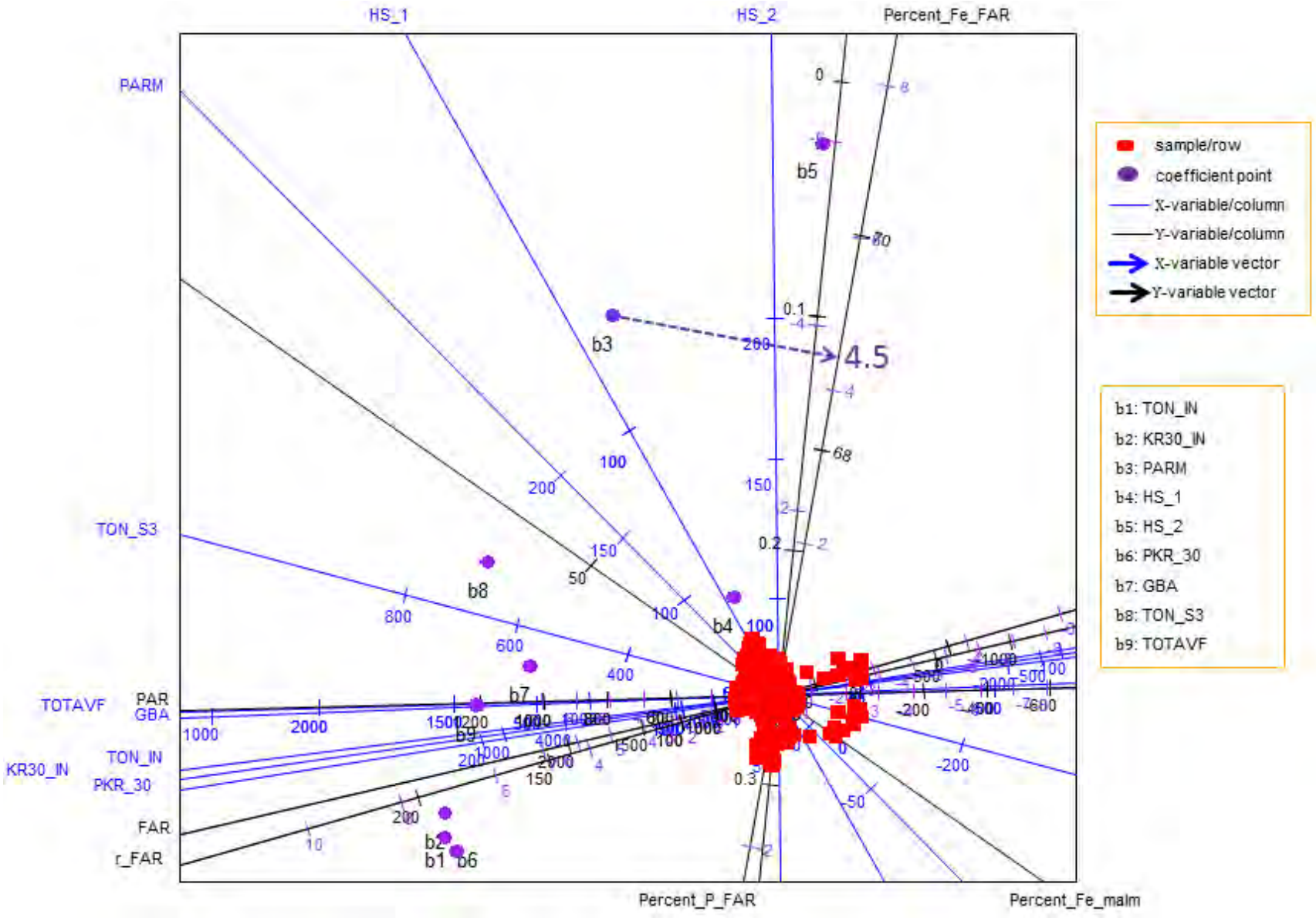

Figure 8.2 The PLS biplot of the SOVR data, using components 1 and 2.

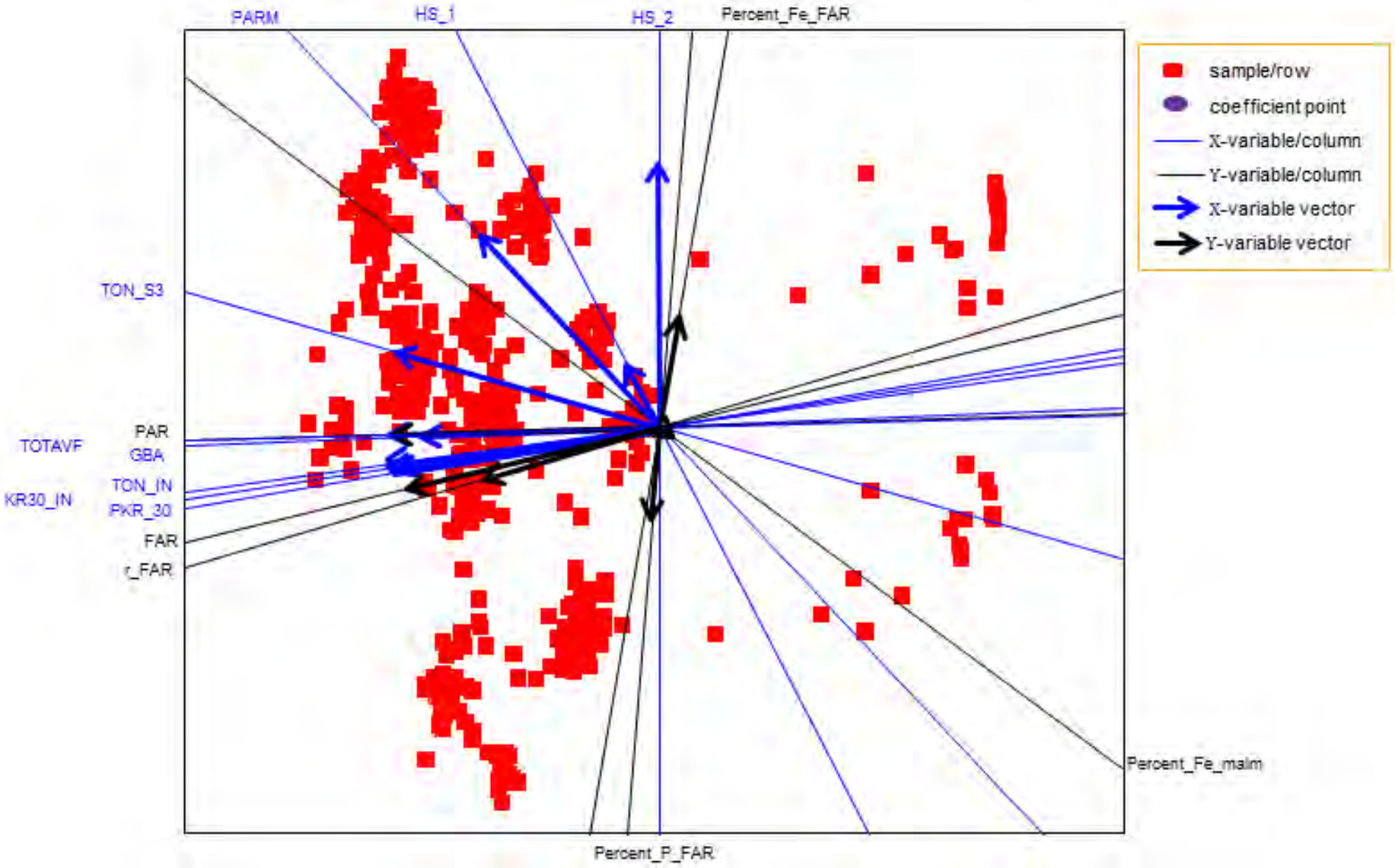

Figure 8.3 The PLS biplot of the SOVR data (Figure 8.2), with zoomed-in display of the samples and variable vectors.

Also, the relation between process factor HS_2 and output variables Percent_P_FAR and Percent_Fe_FAR can be seen. The relation between factor HS_2 and Percent_P_FAR is a negative one, while the relation between factor HS_2 and Percent_Fe_FAR is a positive one. More deductions are listed in Subsection 8.2.4.

Sample names have been excluded from this biplot (Figure 8.2). Orthogonally projecting each red point in the biplot onto the axes will give the approximated values of the SOVR data. Similarly, the PLSR coefficient points $\mathrm{b}i$, for $i = 1, 2, \cdots, 9$, are projected onto the axes representing the output variables. However, *the purple markers on these axes are used to read off these values*, see Subsection 5.2.3. For example, *coefficient point* b3 *projected orthogonally onto the Percent_Fe_FAR axis yields a value of* 4.5, as shown in Figure 8.2. The estimated regression coefficient values of the SOVR data are shown below in Table 8.2.

Alternatively, one can use the area biplot methodology, see Section 5.5, to obtain the coefficient values. Here, the coefficients are estimated by the areas of triangles. However, it is not intuitive to estimate the exact area of a triangle visually, but as an exploratory tool, larger and smaller coefficients can be easily discerned. Large triangles indicate large coefficient values, while small triangles indicate small coefficient values. Figure 8.4 shows an illustration of the area biplot methodology to estimate coefficient points $\mathrm{b}i$, for $i = 1, 2, \cdots, 9$, under the output variable Percent_Fe_FAR. Point b5 can be seen to have a larger value, followed by b3, while point b7 has the smallest value. The exact coefficient values can be obtained by printing the coefficient matrix $\widehat{\mathbf{B}}_{\mathrm{PLSR}}$, shown below in Table 8.2.

Table 8.2 The estimated PLSR coefficient values, using components 1 and 2.

	PAR	FAR	r_FAR	Percent_Fe_FAR	Percent_P_FAR	Percent_Fe_malm
b1: TON_IN	10.030	10.426	7.383	-2.513	1.862	-0.345
b2: KR30_IN	9.970	10.213	7.207	-2.191	1.593	-0.373
b3: PARM	4.600	1.909	0.895	4.496	-3.902	-0.710
b4: HS_1	1.250	0.555	0.275	1.150	-1.000	-0.186
b5: HS_2	-1.790	-5.264	-4.268	7.134	-5.961	-0.591
b6: PKR_30	9.680	10.183	7.230	-2.664	1.999	-0.309
b7: GBA	7.340	6.774	4.660	-0.153	-0.057	-0.417
b8: TON_S3	8.500	7.195	4.833	1.102	-1.143	-0.608
b9: TOTAVF	8.990	8.588	5.960	-0.758	0.411	-0.455

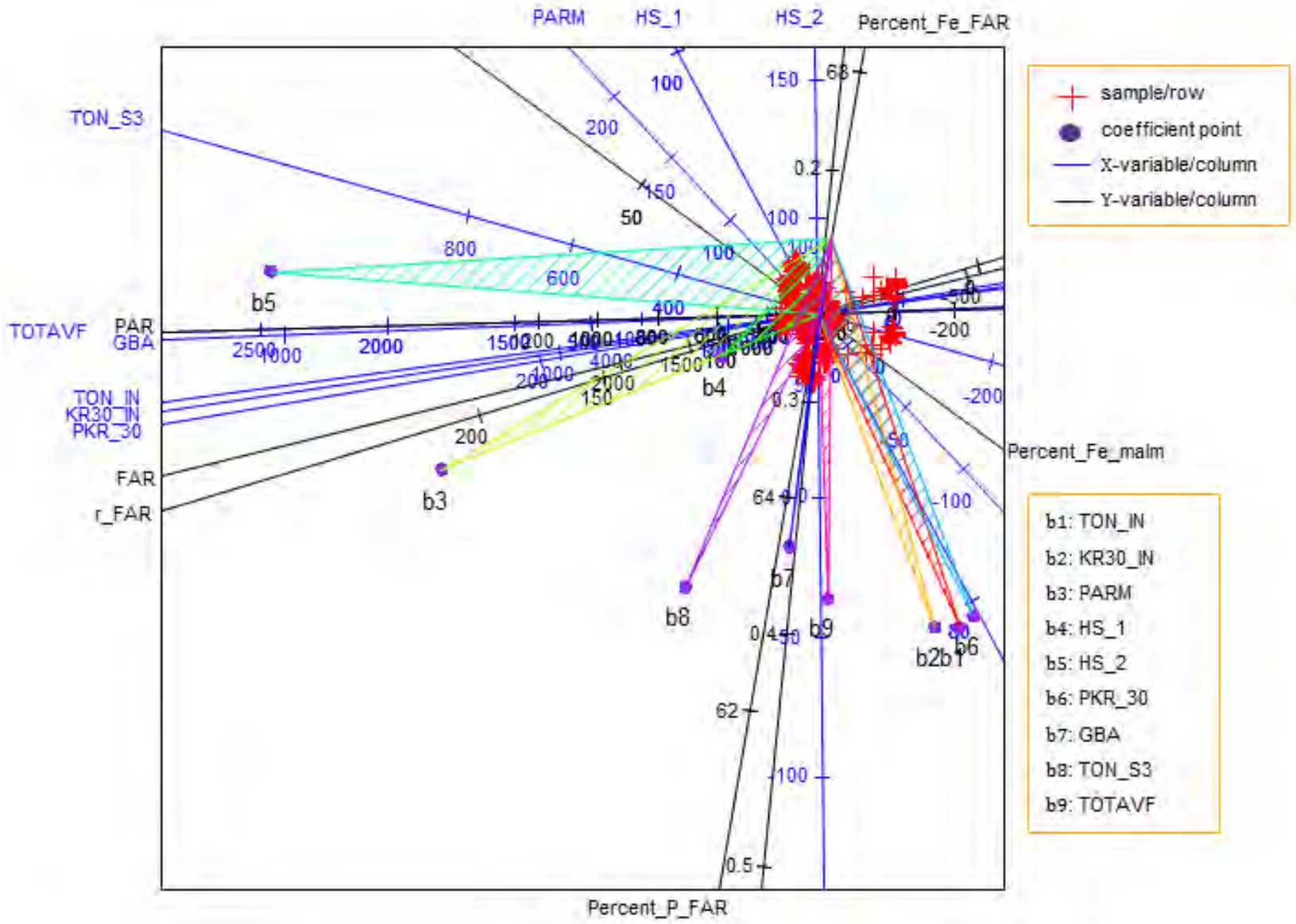

Figure 8.4 The triangles for points bi, i = 1, 2, ⋯ , 9, with bases defined by the Percent_Fe_FAR axis in the PLS biplot of the SOVR data.

Observing the mean plot of the absolute values of these coefficients, shown below in Figure 8.5, HS_1 seems to be the process factor with the lowest influence, while TON_IN has the highest influence. Choosing a benchmark of 0.7 as moderate, the following influential coefficients can be deduced, with their respective magnitudes given in parenthesis. A low magnitude indicates that the coefficient is not quite influential.

b1: TON_IN (highest)
b2: KR30_IN (higher)
b3: PARM (lower)
b4: HS_1 (lowest)
b5: HS_2 (high)
b6: PKR_30 (higher)
b7: GBA (low)
b8: TON_S3 (medium)
b9: TOTAVF (high).

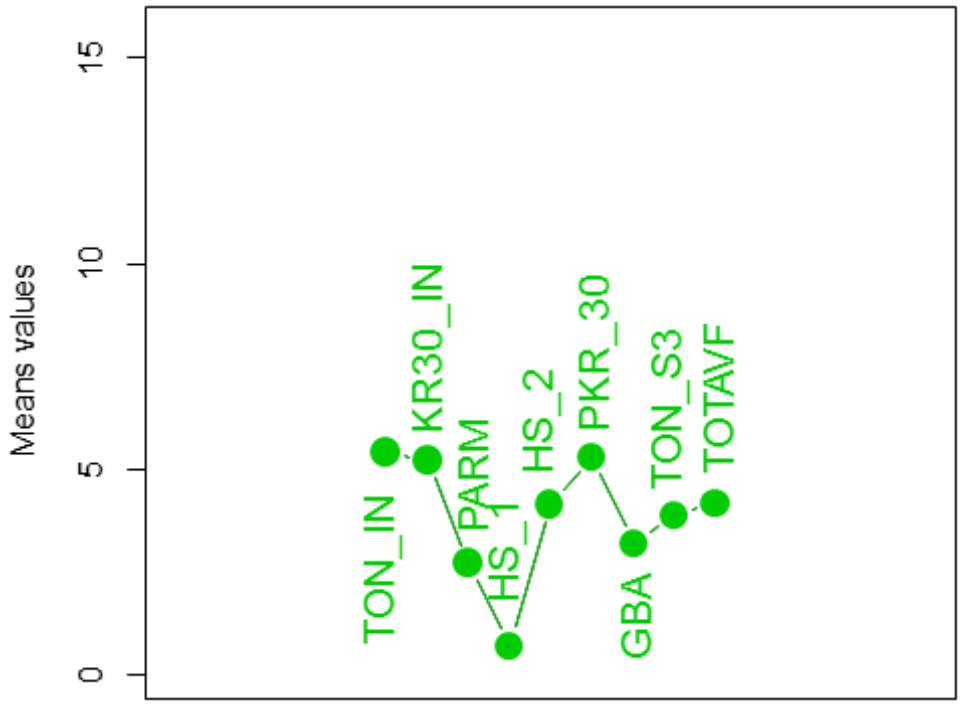

Figure 8.5 Mean plot of the absolute PLSR coefficients of the SOVR data, using components 1 and 2.

Furthermore, for Figure 8.2, the predictivity of each biplot axis is calculated, and this is shown below in Table 8.3. Each of these axes represents the original data quite well, with the Percent_Fe_FAR axis having the highest axis predictivity value of 1.000, followed by the

Percent_Fe_malm axis with 0.999. However, the GBA axis has the lowest axis predictivity value of 0.954. These predictivity values, along with the overall quality value (0.992), indicate that the PLS biplot approximates the SOVR data very well.

Table 8.3 Axis predictivity of the PLS biplot of the SOVR data.

TON_IN	KR30_IN	PARM	HS_1	HS_2	PKR_30	GBA	TON_S3
0.996	0.993	0.957	0.998	0.997	0.991	0.954	0.994

TOTAVF	PAR	FAR	r_FAR	Percent_Fe_FAR	Percent_P_FAR	Percent_Fe_malm
0.991	0.988	0.983	0.959	1.000	0.980	0.999

8.2.4 *Summary*

With the aim of investigating the relationships between the process factors and output variables, the PLS biplot of the SOVR data helps to reveal the relation

(i) within output variables FAR and r-FAR;

(ii) within process factors TON_IN, KR30_IN and PKR_30;

(iii) within output variables Percent_P_FAR and Percent_Fe_FAR;

(iv) within process factors TOTAVF and GBA;

(v) between process factor HS_2 and output variables Percent_P_FAR and Percent_Fe_FAR; and

(vi) between output variable PAR and process factors TOTAVF and GBA.

It further reveals quite the lack of relation

(i) within output variables Percent_P_FAR and Percent_Fe_malm;

(ii) between process factor TON_S3 and output variable Percent_P_FAR; and

(iii) between output variable Percent_Fe_malm and process factor HS_2.

Moreover, a positive relation exists within output variables FAR and r-FAR, within process factors TOTAVF and GBA, within process factors TON_IN, KR30_IN and PKR_30, between output variable PAR and process factors TOTAVF and GBA, as well as between process factor HS_2 and output variable Percent_Fe_FAR. On the other hand, a negative relation exists within output variables Percent_P_FAR and Percent_Fe_FAR, and between process factor HS_2 and output variable Percent_P_FAR. In addition, process factors TON_IN, PKR_30, KR30_IN, TOTAVF and HS_2 have a high influence in the analysis, while factors GBA, PARM and HS_1 seems to have a low influence. However, factor TON_S3 has a medium influence in the analysis.

From the overall quality and axes predictivity of this PLS biplot, it can be concluded that the abovementioned relationships were approximated quite well. Thus, the PLS biplot can be used as a graphical tool for displaying the relationships in such multivariate data.

8.3 The PLS biplot of the Pima.tr data

This data shows the diabetes testing of two hundred women from Pima Indian heritage, living near Phoenix Arizona. This testing was done using the World Health Organization criteria. Observations were recorded on eight different variables, namely, number of pregnancies (npreg), plasma glucose concentration in an oral glucose tolerance test (glu), diastolic blood pressure (bp), triceps skin fold thickness (skin), body mass index (bmi), diabetes pedigree function (ped), age in years (age) and Yes or No, for diabetic according to the organization's criteria (type). This data can be obtained from the **MASS** package in R, downloaded freely from CRAN, http://cran.r-project.org/. Since the type variable is binary, a Binomial PLS-GLM is fitted. Here, $\boldsymbol{\eta} = \log\left(\frac{\boldsymbol{\pi}}{1-\boldsymbol{\pi}}\right) = \mathbf{Xb}$ and $\boldsymbol{\pi} = e^{\mathbf{X}_0\mathbf{b}}/\left[1 + e^{\mathbf{X}_0\mathbf{b}}\right]$. The two hundred women are assigned as the samples. Variables npreg, glu, bp, skin, bmi, ped and age are the predictor variables, while variable type is the response variable. Thus, $\mathbf{X}$: 200×7 and $\mathbf{y}$: 200×1 in the analysis.

Prior to this analysis, the type vector $\mathbf{y}$: 200×1 was recoded as 1 for Yes and 0 for No. The resulting PLS biplot is shown in Figure 8.6 below, along with a representation of the variance of each variable. Here, the PLS-GLM is fitted using Algorithm 6.2 and the PLS biplot discussed in Chapter 6 is used. Another display of this biplot, where the sample point names have be excluded in the biplot, is shown in Figure 8.7 below. Observing the length of the thicker arrows (vectors), bmi can be said to have a larger standard deviation, followed by bp. In addition, observing the angles between the blue vectors, all the predictor variables can be said to be positively related to each other.

To get the approximated values from Figure 8.6, each sample point is orthogonally projected onto the axes and their values read off. Due to the large number of samples, 200, only the first fifteen predicted values of the seven variables and the expected type of diabetic are shown in Table 8.5. Likewise, the PLS-GLM coefficient points bi, for $i = 1, 2, \cdots, 7$, are projected onto the axis representing the type of diabetic, and their values are *read off using the purple markers on the axis* (Subsection 5.2.3 and Section 6.4).

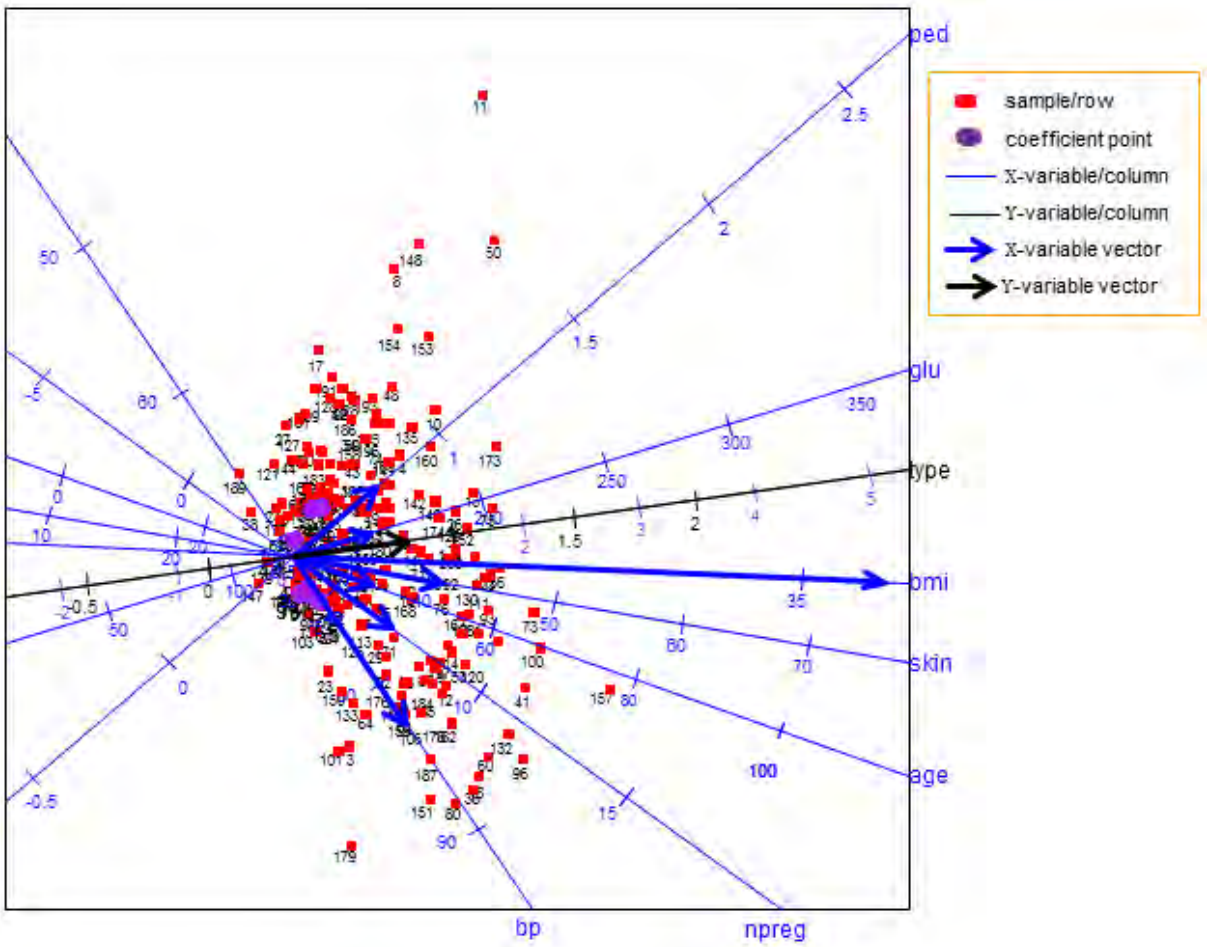

Figure 8.6 The PLS biplot of a Binomial PLS-GLM of the Pima.tr data (Algorithm 6.2).

For a zoomed-in display of the coefficient points, see Figure 8.8. This display can be used for easier orthogonal projections of the coefficient points. For example, coefficient points b5 and b6 projected orthogonally onto the type axis yields 0.05 and 0.22 respectively, as shown in Figure 8.8. The obtained coefficient values from the PLS-GLM are shown below in Table 8.4.

Table 8.4 The predicted coefficient values.

b1: npreg	b2: glu	b3: bp	b4: skin	b5: bmi	b6: ped	b7: age
0.084	0.316	0.004	0.092	0.045	0.217	0.169

From Table 8.4, all the variables can be said to have a positive effect on the type of diabetic. This can also be observed in the biplot display in Figure 8.6 above, by looking at the directions of the blue axes. Variables glu, ped and age can be said to have a high effect on the type of diabetic. Other variables have low effect, with bp having the lowest to no effect.

Table 8.5 The predicted values of the variables and the expected type of diabetic.

	npreg	glu	bp	skin	bmi	ped	age	type*
1	5	130	74	32	33	0.447	38	0
2	10	192	82	47	33	0.749	62	1
3	8	125	82	37	33	0.237	48	0
4	5	176	70	37	33	0.879	42	1
5	3	128	70	29	32	0.528	31	0
6	8	150	82	40	33	0.441	53	1
7	4	131	72	31	32	0.507	34	0
8	1	195	61	34	33	1.207	33	1
9	9	166	82	42	33	0.565	56	1
10	5	195	69	39	33	1.038	44	1
11	1	248	56	38	33	1.718	36	1
12	9	168	83	43	33	0.557	58	1
13	7	199	74	43	33	0.968	53	1
14	7	163	77	39	33	0.641	49	1
15	5	163	72	36	33	0.739	42	1

*1 = Yes and 0 = No.

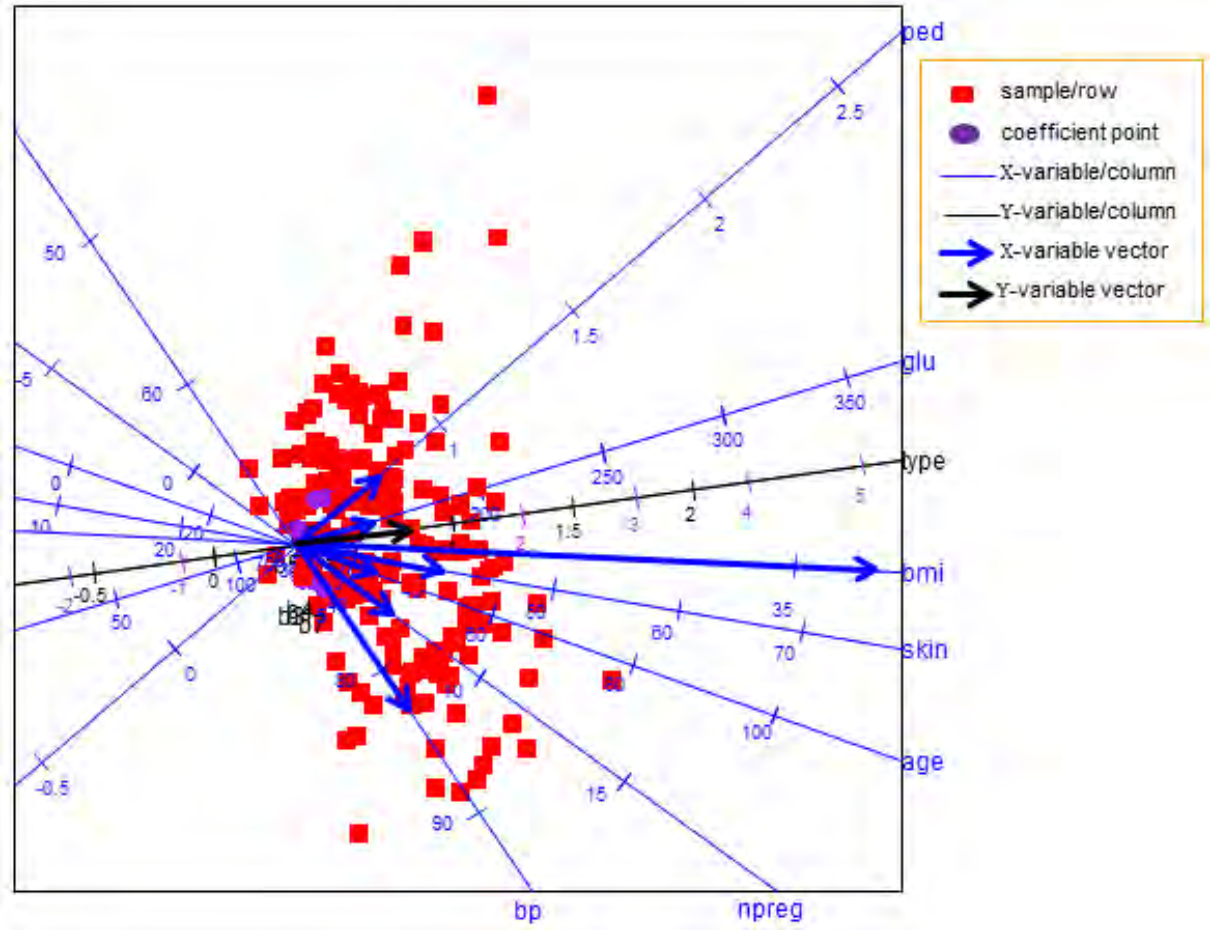

Figure 8.7 The PLS biplot of a Binomial PLS-GLM of the Pima.tr data (Algorithm 6.2), without sample names.

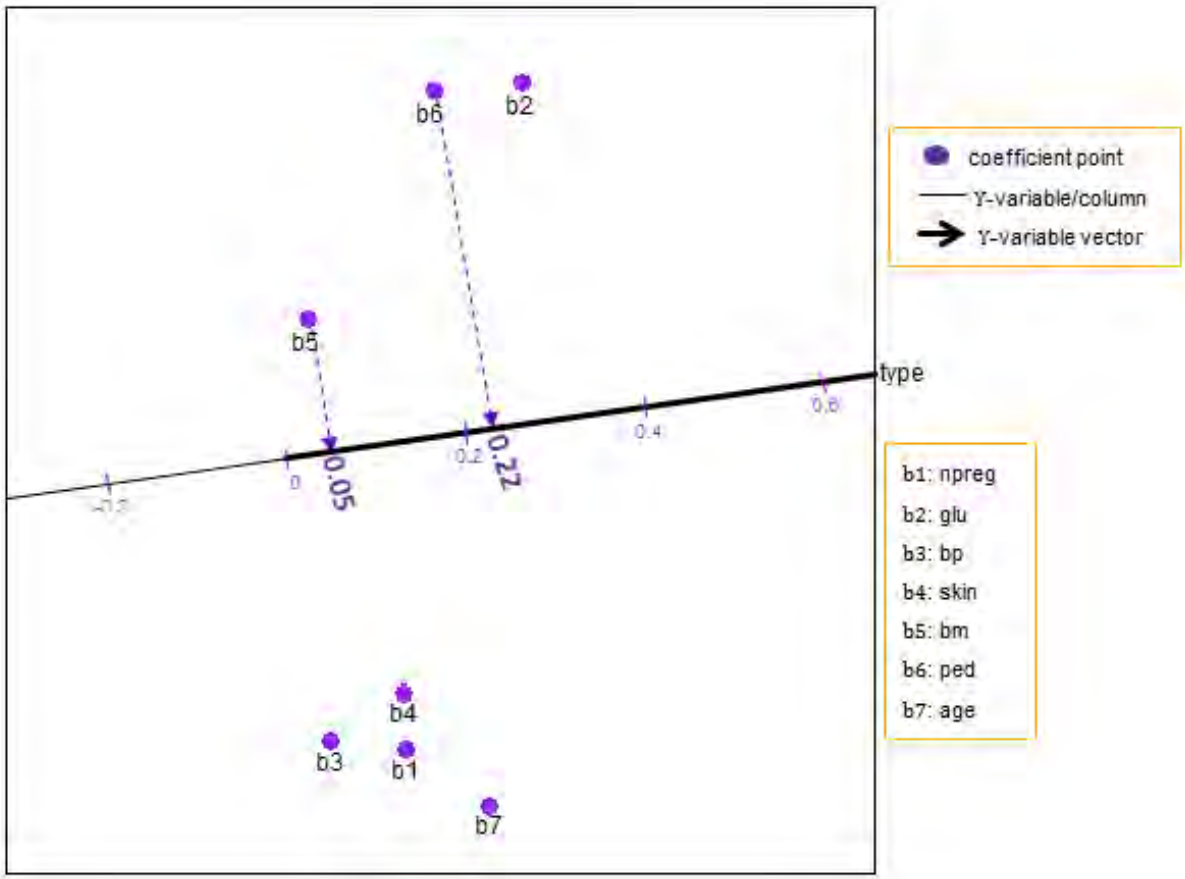

Figure 8.8 A zoomed-in display of the coefficient points in the PLS biplot of a Binomial PLS-GLM of the Pima.tr data.

8.3.1 *Summary*

In this section, the PLS biplot is used as a graphical tool for the Binomial PLS-GLM. Applying the PLS biplot to this Binomial PLS-GLM, the resulting biplot provides a single graphical representation for displaying the result from the PLS-GLM analysis of the data set.

8.4 The PLS biplot of the colon data

This data shows the gene expression levels from the microarray experiments of colon tissue samples taken from sixty two patients. Two thousand genes were used. In addition, the tissue samples were categorized into two types, namely, the tumor tissues and the normal tissues. Out of the sixty two patients, forty were categorized as having tumor tissues, while twenty two were categorized as having normal tissues. The types of tissue samples were coded as one for the normal tissues and two for the tumor tissues, thereby forming a vector of ones and twos. In this data, this vector is named the tissue variable. The sixty two patients are the samples. The genes and tissue variable are assigned as the predictor and response variables respectively. Thus, the colon data can be viewed as a data matrix $\mathbf{X}$ (62×2000) of predictors and a matrix $\mathbf{Y}$ (62×1)

of responses. Since $M = 1$, the matrix $\mathbf{Y}\,(62 \times 1)$ is in fact a vector of length (62×1), as mentioned above. This data can be obtained from the **plsgenomics** package in R, downloaded freely from CRAN, http://cran.r-project.org/.

With the large number of X-variables (genes), there is the need to perform variables selection on these X-variables, as some of these genes might be irrelevant in the analysis. Thus, an SPLS-GLM analysis is more appropriate for the colon data, than a PLS-GLM analysis.

Before performing an SPLS-GLM analysis on this data, the experiment described in Subsection 7.4.1 was performed, using Algorithm 7.2. Prior to running this experiment, the tissue vector $\mathbf{y}\,(62 \times 1)$ was recoded as 0 for normal tissues and 1 for the tumor tissues, to employ the Binomial SPLS-GLM, where $\boldsymbol{\eta} = \log\left(\frac{\boldsymbol{\pi}}{1-\boldsymbol{\pi}}\right) = \mathbf{Xb}$ and $\boldsymbol{\pi} = e^{\mathbf{X}_0\mathbf{b}}/\left[1 + e^{\mathbf{X}_0\mathbf{b}}\right]$. Seeing as the colon data has only one Y-variable, i.e., $M = 1$, there is no need to perform variables selection on this Y-variable. As a result, $\lambda_Y = 0$ throughout this experiment and much attention is given to the value of λ_X that gives the lowest RMSEP value. Figure 8.9 below shows a plot of the obtained RMSEP values along with their respective λ_X value, for $\lambda_X \in (0, 9)$ and $\lambda_Y = 0$.

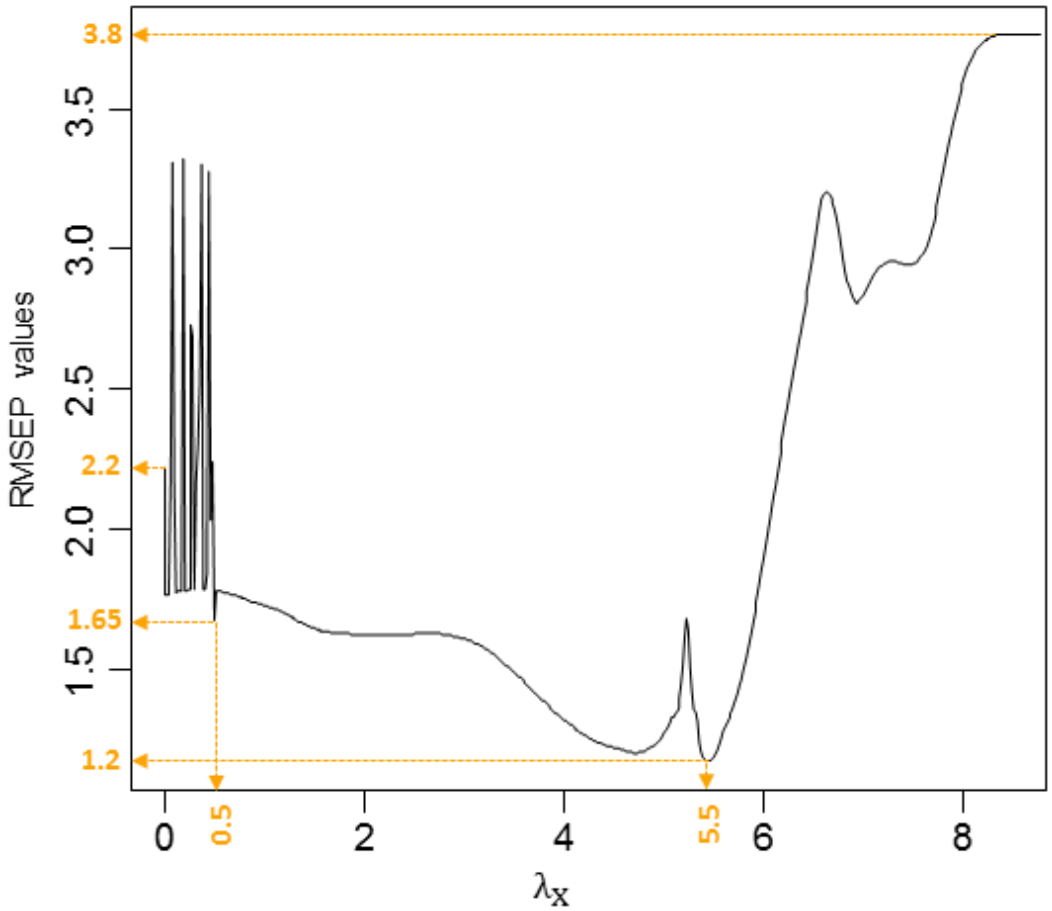

Figure 8.9 A plot of $\lambda_X \in (0, 9)$ values and their respective RMSEP value, for the colon data.

From Figure 8.9, the lowest RMSEP value of 1.2 is observed at $\lambda_X = 5.5$. Thus, for the SPLS-GLM analysis of the colon data, $\lambda_X = 5.5$ and $\lambda_Y = 0$. In the resulting SPLS-GLM analysis ($\lambda_X = 5.5$ and $\lambda_Y = 0$), one hundred and twenty eight genes (X-variables) were selected, out of the two thousand genes used. For this reason, the number of X-variables used in the resulting biplot display (Figure 8.10) was reduced from two thousand to one hundred and twenty eight, i.e., $\mathbf{X}\,(62 \times 128)$. With $\lambda_X = 5.5$ and $\lambda_Y = 0$, using Algorithm 7.2, the resulting PLS biplot is shown below in Figure 8.10. A representation of the variance of each variable is also shown in this biplot display. Comparing the length of the thicker arrows (vectors) to each other, the tissue variable and genes 715 and 22 can be said to have a smaller standard deviation.

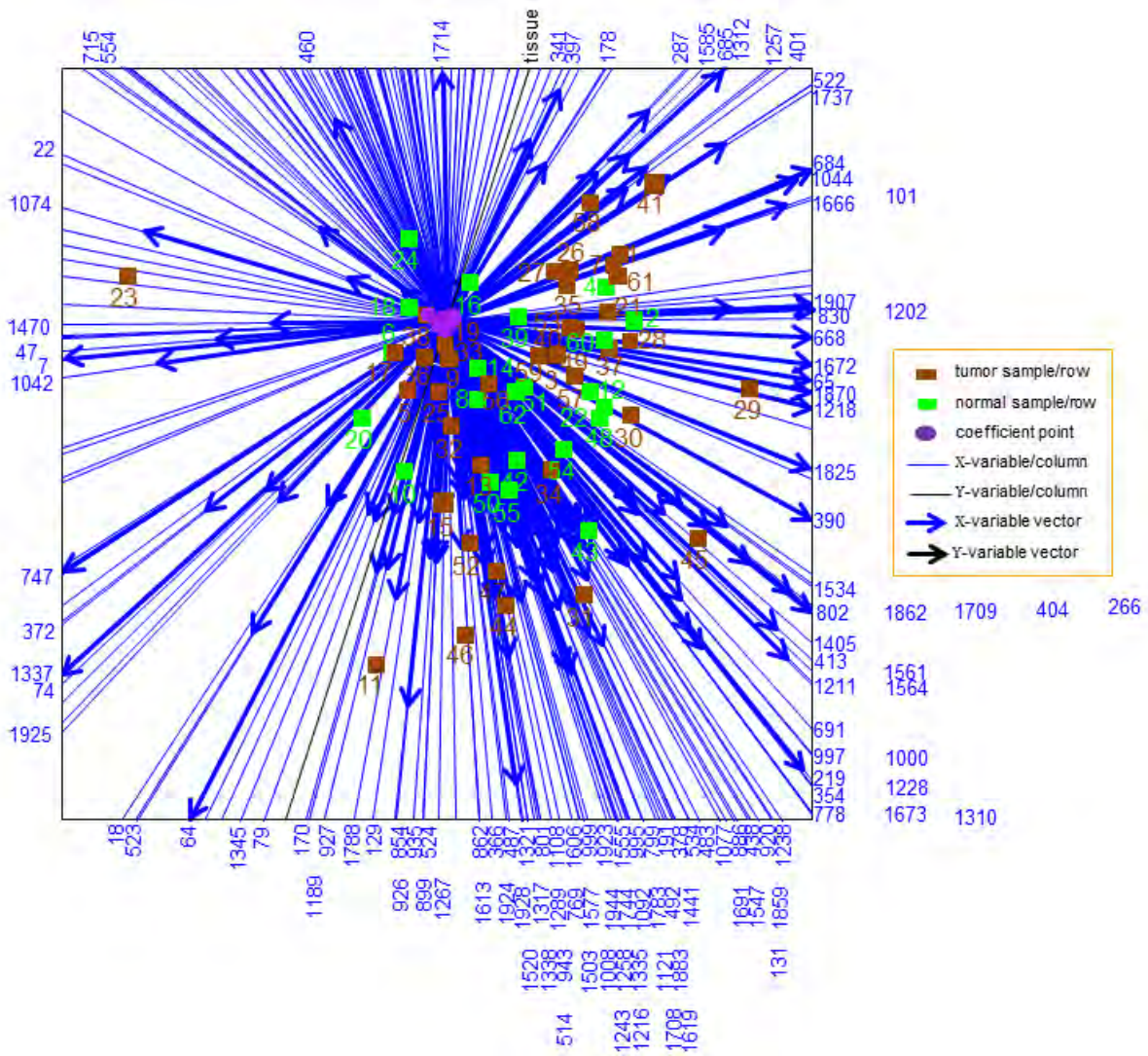

Figure 8.10 The PLS biplot of a Binomial SPLS-GLM of the colon data (Algorithm 7.2), with $\lambda_X = 5.5$ and $\lambda_Y = 0$.

Axis tick markers have been excluded, and as mentioned above, there are two types of tissue samples in this data, namely, tumor and normal. In the biplot display (Figure 8.10), the tumor tissue samples are represented by the brown sample points, while the normal tissue samples are

represented by the green points. Tumor samples 23, 29 and 45 can be classified as outliers. Although there is no clear separation between the tumor and normal samples in the biplot, the normal samples have less variability. Therefore, one can conclude that the normal samples have some characteristics similar to the tumor samples.

In addition, some tumor samples also appear towards the centre (overall mean) of the biplot (Figure 8.10), however, tumor samples 41 and 58 can be said to have a higher value on genes 287, 1585, 685, 1312, 1257, 401, 522 and 1737, but a low value on genes 747, 372, 1337, 74 and 1925. Similarly, tumor sample 29 has a higher value on genes 1907, 830, 1202, 1672, 65, 1870, 1218 and 668, but a low value on genes 1042, 47, 7 and 1470. Tumor samples 31, 44 and 46 can be said to have a high value on genes 1613, 862, 487, 366, 1924, 1928, 801, 1321, 1520, 1317, 1606, 1108, 1289, 1338, 514, 943, 1577, 999, 1923, 769, 1503, 1243, 1008, 1944, 1555, 1258, 1744, 995, 1216, 1335, 1783, 1092, 799, 1708, 1121, 378, 492, 191, 1883, 1619, 1441, 534 and 483, but a low value on gene 460. On the other hand, tumor sample 11 has a low value on gene 1714, but high a value on genes 170, 1189, 927, 1788, 129, 926, 854, 935, 899, 524 and 1267. In addition, tumor sample 45 can be said to have a high value on genes 1534, 802, 1862, 1709, 404, 266, 1405, 413, 1561, 1211 and 1564, but a low value on genes 715 and 554. Tumor sample 23, clearly an outlier as mentioned above, can be said to have a higher value on genes 1042, 47, 7 and 1470. These deductions can be observed by the closeness of the sample points to the gene axes.

Similarly, the relation between the selected genes can be observed by the closeness of the blue axes. To name a few, the relation between genes 684, 1044, 101 and 1666; between genes 1907, 830, 1202, 668, 1672, 65, 1870, 1218, 1470, 47, 7 and 1042; as well as between genes 18, 523 and 178.

Since the name of the genes were not provided in the data source, besides their respective number, one might consider consulting a biologist to see if these deductions are insightful and extremely meaningful, especially, in a biological context.

Furthermore, a zoomed-in display of the coefficient points is shown below in Figure 8.11. Orthogonally projecting each of the coefficient points bi, for $i = 1, 2, \cdots, 128$, onto the tissue axis yields the coefficient values. For example, coefficient points $b29$, $b43$, $b85$ and $b86$ projected orthogonally onto the tissue axis yields 0.06, 0, 0.14 and -0.17 respectively, as shown in Figure 8.11. Although not all coefficient points can be seen clearly in this display (Figure 8.11), the exact coefficient values can be obtained by printing out the coefficients vector

$\hat{\mathbf{b}}_{\mathrm{SPLS-GLM}}$. This vector is shown in Table 8.6 below. The corresponding gene number for each coefficient point is given along with the coefficient point number. For example, b1 and b2 denotes genes 7 and 18 respectively, as shown below in Figure 8.11 and Table 8.6.

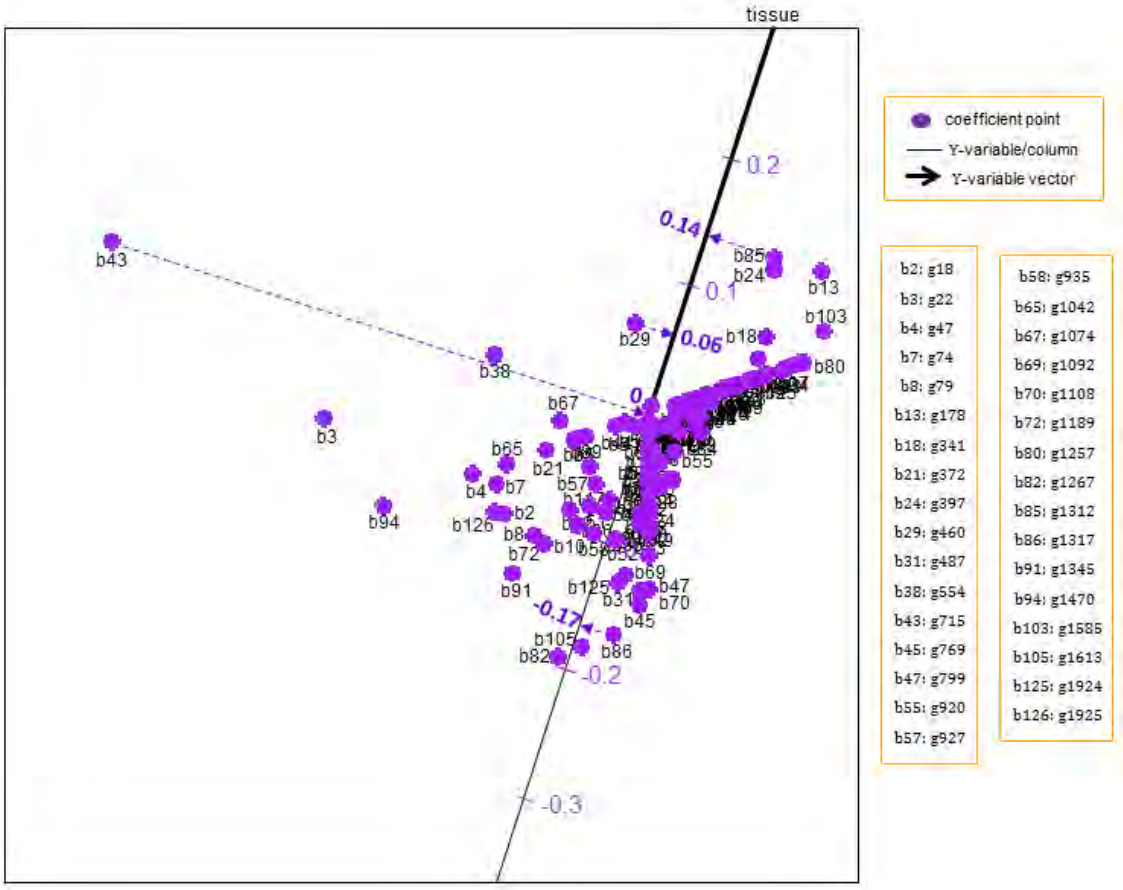

Figure 8.11 A zoomed-in display of the coefficient points in the PLS biplot of a Binomial SPLS-GLM of the colon data, with $\lambda_X = 5.5$ and $\lambda_Y = 0$.

In addition, the positive or negative sign before each coefficient value in Table 8.6 does not measure the effect level of the genes on the tissue variable, but rather, is the effect direction of each gene on the tissue variable. For example, genes 7, 18, 22, 47 and 64 all have a negative effect on the tissue variable, while genes 65, 101 and 131 have a positive effect.

Table 8.6 The predicted coefficient values.

b1: g7	**b2: g18**	**b3: g22**	**b4: g47**	**b5: g64**	**b6: g65**	**b7: g74**	**b8: g79**
-0.03494	-0.10364	-0.07661	-0.08234	-0.01127	0.02604	-0.08411	-0.11184

b9: g101	**b10: g129**	**b11: g131**	**b12: g170**	**b13: g178**	**b14: g191**	**b15: g219**	**b16: g266**
0.05364	-0.09549	0.03909	-0.08499	0.14172	-0.00345	0.01745	0.04473

b17: g287	**b18: g341**	**b19: g354**	**b20: g366**	**b21: g372**	**b22: g378**	**b23: g390**	**b24: g397**
0.04889	0.08193	-0.07902	-0.04653	-0.04785	-0.04732	0.00280	0.13205

Table 8.6 Cont^{D}.

b25: g401	b26: g404	b27: g413	b28: g438	b29: g460	b30: g483	b31: g487	b32: g492
0.06977	0.03452	0.02031	-0.04067	0.06246	-0.00549	-0.12617	-0.00504

b33: g514	b34: g522	b35: g523	b36: g524	b37: g534	b38: g554	b39: g668	b40: g684
-0.08264	0.07272	-0.03328	-0.07805	-0.03260	0.00770	0.04544	0.06548

b41: g685	b42: g691	b43: g715	b44: g747	b45: g769	b46: g778	b47: g799	b48: g801
0.06861	0.02650	-0.00074	-0.01575	-0.13725	0.03343	-0.12412	-0.04264

b49: g802	b50: g830	b51: g854	b52: g862	b53: g886	b54: g899	b55: g920	b56: g926
0.03523	0.04948	-0.07848	-0.09589	0.00732	-0.06998	-0.01972	-0.01117

b57: g927	b58: g935	b59: g943	b60: g995	b61: g997	b62: g999	b63: g1000	b64: g1008
-0.04934	-0.09734	-0.02191	-0.03734	0.01084	-0.01192	0.03876	-0.04179

b65: g1042	b66: g1044	b67: g1074	b68: g1077	b69: g1092	b70: g1108	b71: g1121	b72: g1189
-0.06708	0.04475	-0.02467	0.02146	-0.09943	-0.12707	-0.02578	-0.11571

b73: g1202	b74: g1211	b75: g1216	b76: g1218	b77: g1228	b78: g1238	b79: g1243	b80: g1257
0.05223	0.04204	-0.03254	0.01267	0.01356	0.01048	-0.01080	0.07090

b89: g1337	b90: g1338	b91: g1345	b92: g1405	b93: g1441	b94: g1470	b95: g1503	b96: g1520
-0.02983	-0.07103	-0.14417	0.05054	-0.00305	-0.12470	-0.04710	-0.07898

b97: g1534	b98: g1547	b99: g1555	b100: g1561	b101: g1564	b102: g1577	b103: g1585	b104: g1606
0.04043	0.02373	-0.04163	0.04131	0.02136	-0.07707	0.09901	-0.07149

b105: g1613	b106: g1619	b107: g1666	b108: g1672	b109: g1673	b110: g1691	b111: g1708	b112: g1709
-0.17971	-0.00779	0.06646	0.04649	0.01539	0.02078	0.02184	0.02375

8.4.1 *Summary*

In this section, the PLS biplot is used as a graphical tool to visualize the SPLS-GLM of a data set. After applying the PLS biplot to the SPLS-GLM of a large gene expression large data, the resulting biplot demonstrates, graphically, the association between samples and variables as well as between the variables. However, since the name of the genes were not provided in the data source, besides their respective number, one might consider consulting a biologist to see if the observed deductions made from the PLS biplot of the data are insightful and extremely meaningful, especially, in a biological context.

8.5 The PLS biplot software

The PLS biplot is available in the newly developed R package called **PLSbiplot1**. This package is available electronically via CRAN (http://cran.r-project.org/) and the dropbox link

https://www.dropbox.com/sh/wr66u07t1vjm9da/AACg_E4h8MvgOHuCXk69yDIya.

A total of thirty five functions are in this package, namely,

`cov.biplot`	(covariance biplot)
`cov.monoplot`	(covariance monoplot)
`Mag.Bmat.plot`	(mean plot of the absolute values of the PLS coefficients matrix)
`mod.KernelPLS_L`	(kernel PLS algorithm by Lindgren *et al.* (1993))
`mod.KernelPLS_R`	(kernel PLS algorithm by Rännar *et al*. (1994))
`mod.MMLR`	(MMLR analysis)
`mod.NIPALS`	(NIPALS algorithm)
`mod.PCA`	(PCA analysis)
`mod.PCR`	(PCR analysis)
`mod.SIMPLS`	(SIMPLS algorithm)
`mod.SPLS`	(SPLS algorithm by Lê Cao *et al*. (2008))
`mod.VIP`	(VIP values of the X-variables)
`opt.penalty.values`	(optimal value of the penalty parameters λ_X and λ_Y , for SPLS and SPLS-GLM)
`PCA.biplot`	(PCA biplot)
`PCA.biplot_no.SN`	(PCA biplot, with no sample names)
`PLS.binomial.GLM`	(PLS-GLM algorithm for Binomial-distributed Y-variables)
`PLS.biplot`	(PLS biplot)
`PLS.biplot.area`	(PLS biplot, with the area biplot idea for approximating the coefficient values)
`PLS.biplot_no.SN`	(PLS biplot with no sample names)
`PLS.biplot_no_labels`	(PLS biplot, where the labels of the samples, coefficient points and tick markers have been excluded)
`PLS.GLM`	(PLS-GLM algorithm)
`PLS.GLM.biplot`	(PLS biplot for PLS-GLM)
`PLS.GLM.biplot_bvec`	(zoomed-in display of the coefficient points in the PLS biplot for PLS-GLM)
`PLS.GLM.biplot_no.SN`	(PLS biplot for PLS-GLM, with no sample names)

`PLS.GLM.biplot_SIMPLS`	(PLS biplot for PLS-GLM, fitted using the SIMPLS algorithm)
`PLS.GLM.biplot_SIMPLS_no.SN`	(PLS biplot for PLS-GLM, fitted using the SIMPLS, with no sample names)
`PLS.GLM_SIMPLS`	(PLS-GLM algorithm, fitted using the SIMPLS algorithm)
`SPLS.binomial.GLM`	(SPLS-GLM algorithm for Binomial-distributed Y-variables)
`SPLS.biplot`	(PLS biplot for SPLS)
`SPLS.biplot_Bmat`	(zoomed-in display of the coefficient points in the PLS biplot for SPLS)
`SPLS.biplot_no_labels`	(PLS biplot for SPLS, where the labels of the samples, coefficient points and tick markers have been excluded)
`SPLS.GLM`	(SPLS-GLM algorithm)
`SPLS.GLM.biplot`	(PLS biplot for SPLS-GLM)
`SPLS.GLM.biplot_bvec`	(zoomed-in display of the coefficient points in the PLS biplot for SPLS-GLM)
`SPLS.GLM.biplot_no.SN`	(PLS biplot for SPLS-GLM, with no sample names)

with their main purpose given in parenthesis. These functions requires some input parameters, such as the number of PLS components A, the X- and Y-variables matrices, when utilized.

For more information on a particular function, type `?` followed by the function's name in the R console. For example, `?cov.biplot` opens a window containing the basic information about the `cov.biplot` function and an example on how to execute the function in R, as shown in Figure 8.12 below.

Furthermore, the **PLSbiplot1** package was developed in the 3.1.0 version of R (R Core Team, 2014), using the RStudio software (RStudio, 2012) on a Windows 7 operating system.

The purpose of this section is to provide a brief introductory information on the use of the **PLSbiplot1** package. A detailed documentation for all routines in this package can be found on the dropbox link, given above.

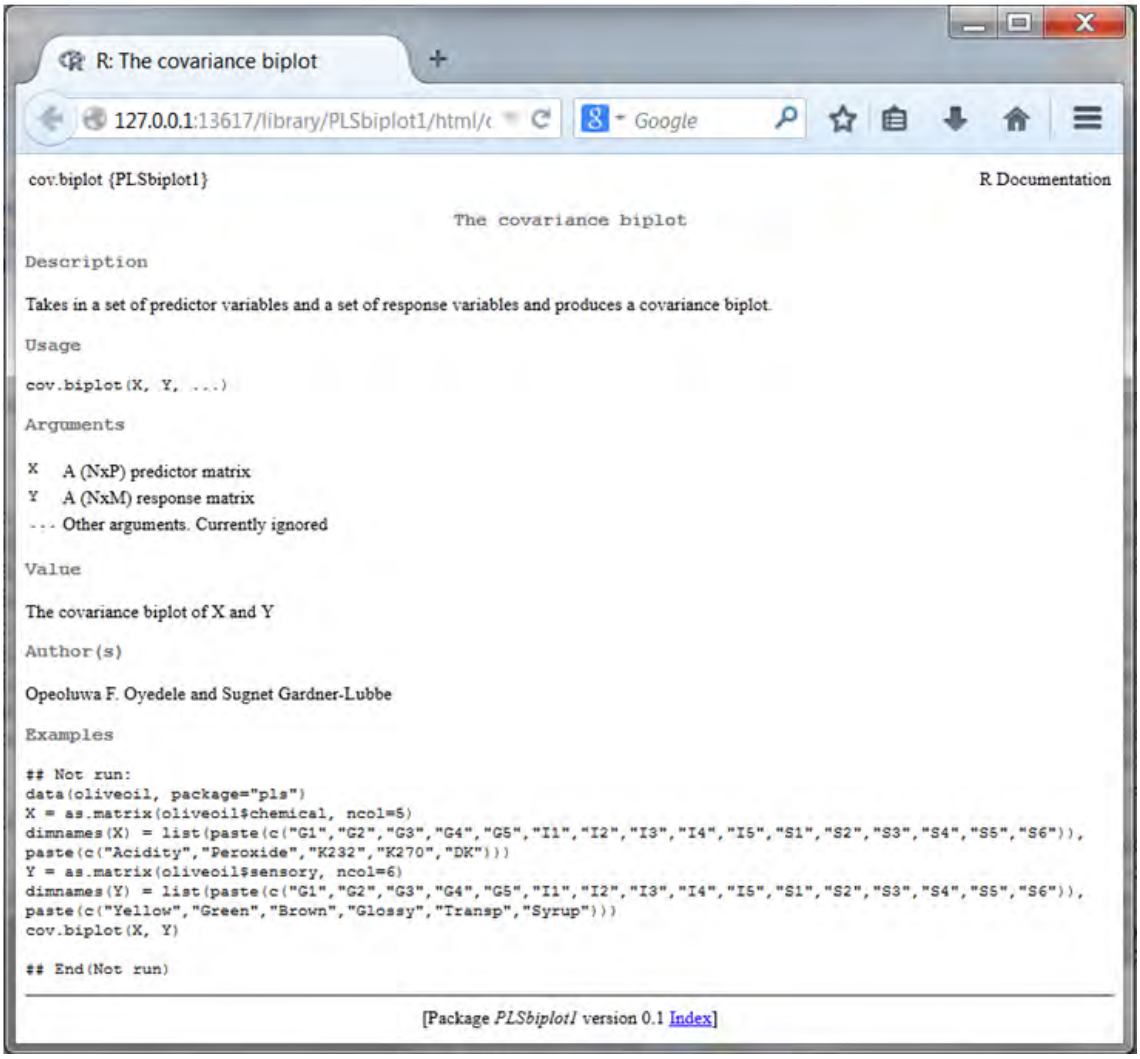

Figure 8.12 The help file of the `cov.biplot` function.

8.5.1 *Installation*

For easier file navigation, the downloaded **PLSbiplot1_0.1.tar.gz** file is saved on the operating system's *Desktop*. To install this **PLSbiplot1** package in R, open the *command prompt (cmd)* window of the operating system, (*Start* menu → *Run* → *cmd*), and type

```
dir Desktop
cd Desktop
R CMD INSTALL PLSbiplot1_0.1.tar.gz
```

These commands tell the operating system to install the **PLSbiplot1** package into R, from the **PLSbiplot1_0.1.tar.gz** file saved on the desktop. An illustration is shown in Figure 8.13. Afterwards, the **PLSbiplot1** package is installed successfully into the R library of the operating system.

To check, open R and type `library()`. Amongst the list of installed packages in the R library shown, the **PLSbiplot1** package can be seen, as shown in Figure 8.14.

To use the **PLSbiplot1** package in R, it must first be loaded into R, using any of these commands: `library(PLSbiplot1)` or `require(PLSbiplot1)`.

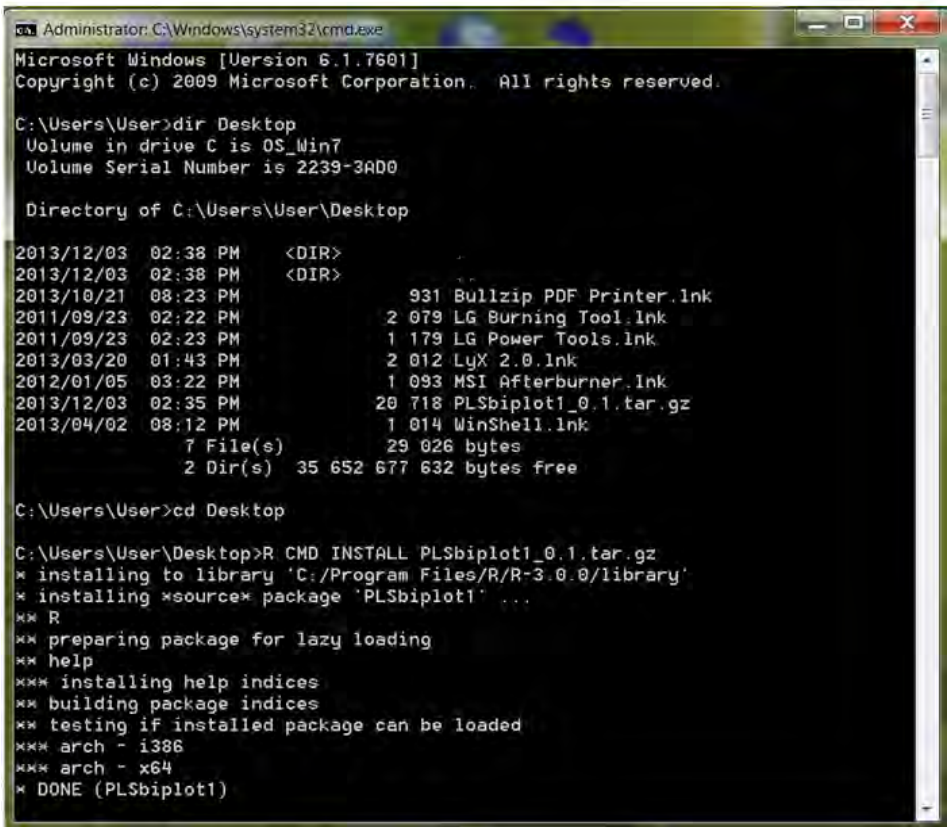

Figure 8.13 The command prompt window.

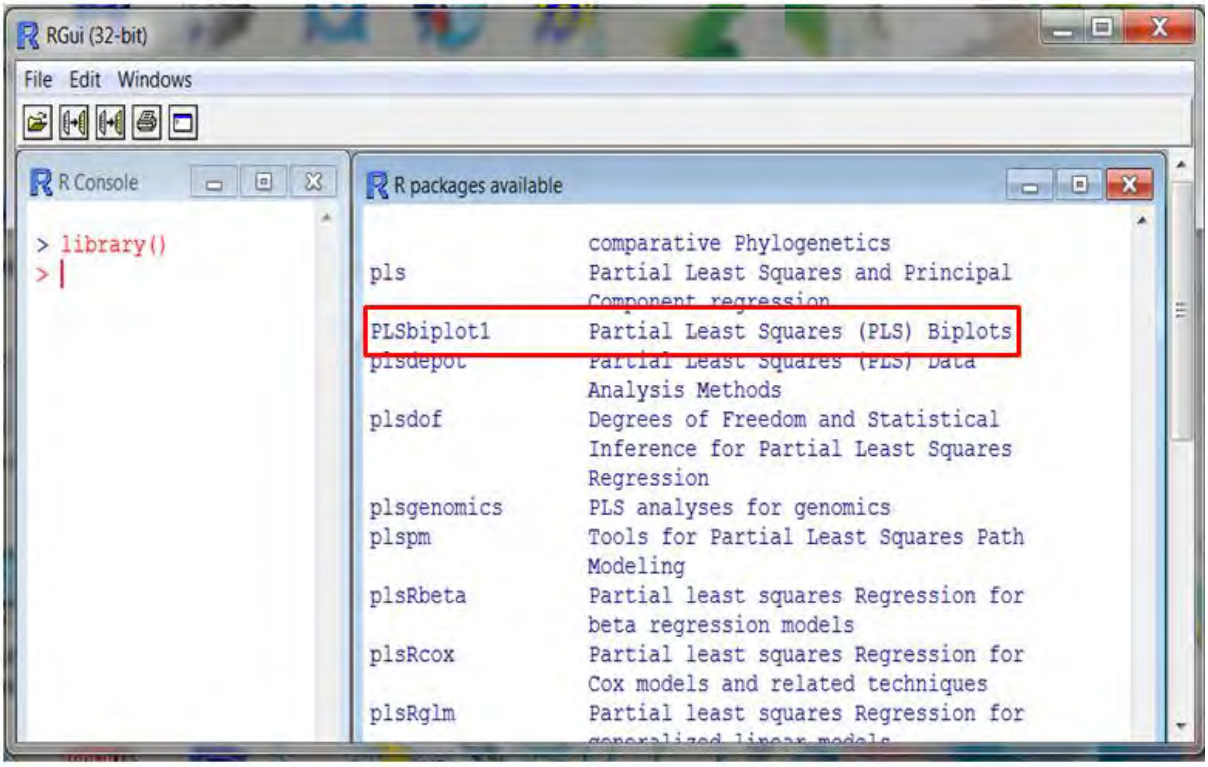

Figure 8.14 List of installed packages in R.

8.5.2 *Data sets*

Five data sets are used for illustrating the use of the **PLSbiplot1** package, namely, the ash and glass data from Varmuza & Filzmoser (2009), the cocktail data from Husson *et al*. (2013), the nutrimouse data by Martin *et al*. (2007) and the spider data from Van der Aart & Smeenk-Enserink (1975).

`ash`

This data shows the experimental softening temperature (SOT) and elemental composition of ninety nine ash samples taken from different biomass. Eight elemental compositions (P2O5, SiO2, Fe2O3, Al2O3, CaO, MgO, Na2O and K2O) were used. The ash samples from different biomass are assigned as samples, while the elemental compositions and SOT are assigned as the predictor and response variables respectively. Thus, the ash data can be viewed as a data matrix $\mathbf{X}\ (99 \times 8)$ of predictors and a matrix $\mathbf{Y}\ (99 \times 1)$ of responses. Here, since $\mathrm{M} = 1$, the matrix $\mathbf{Y}\ (99 \times 1)$ is in fact a vector of length (99×1).

`cocktail`

This data shows the sensory and composition evaluation of sixteen cocktail juices. The composition of each cocktail was measured using four ingredients (Orange, Banana, Mango and Lemon). There were thirteen sensory panel descriptors (Colour_intensity, Odour_intensity, Odour_orange, Odour_banana, Odour_mango, Odour_lemon, Strongness, Sweet, Acidity, Bitterness, Persistence, Pulp and Thickness) used in this evaluation. The sixteen cocktail juices are assigned as the samples. The ingredients and sensory panel descriptors are the predictor and response variables respectively. As a result, the cocktail data can be viewed as a data matrix $\mathbf{X}: 16 \times 4$ of predictors and a matrix $\mathbf{Y}: 16 \times 13$ of responses.

`glass`

This data shows the measurements taken from one hundred and eighty archaeological glass vessels. Thirteen different measurements were taken, and these are assigned as the variables. The glass vessels are assigned as samples. Hence, the glass data can be viewed as a (180×13) data matrix.

`nutrimouse`

This data shows the expression measure of one hundred and twenty genes potentially involved in the nutritional problems and the concentrations of twenty one hepatic fatty

acids of forty mice. The forty mice are assigned as the samples. The hepatic fatty concentrations and gene expressions are the predictor and response variables respectively. Thus, the nutrimouse data can be viewed as a data matrix $\mathbf{X}$: 40×21 of predictors and a matrix $\mathbf{Y}$: 40×120 of responses.

`spider`

This data shows the distributions of hunting spider species and environmental characteristics observed at a dune area. There were twelve spider species used. The twenty eight sites are assigned as the samples. The spider species and environmental characteristics are the response and predictor variables respectively. Therefore, the spider data can be viewed as a data matrix $\mathbf{X}$ (28×6) of predictors and a matrix $\mathbf{Y}$ (28×12) of responses.

The ash and glass data can be obtained from the **chemometrics** package in R, while the spider data can be obtained from the **mvabund** package. Similarly, the cocktail data can be obtained from the **SensoMineR** package, while the nutrimouse data can be obtained from the **mixOmics** package. These packages can be downloaded freely from CRAN, http://cran.r-project.org/. To use these data sets in R, they must first be loaded into R using

```
data(ash, package="chemometrics")
data(cocktail, package="SensoMineR")
data(glass, package="chemometrics")
data(nutrimouse, package="mixOmics")
data(spider, package="mvabund")
```

In addition, to obtain the variable names in each data set, use `names()` or `colnames()`. That is,

```
names(ash) #for ash data
names(c(compo.cocktail,senso.cocktail)) #for cocktail data
colnames(glass) #for glass data
names(nutrimouse) #for nutrimouse data
names(spider) #for spider data
```

8.5.3 *PCA biplot*

To obtain the PCA biplot using the **PLSbiplot1** package in R, the

```
PCA.biplot(D, method=mod.PCA, ax.tickvec.D)
```

function is used. Here, `D` is the data matrix to use for the PCA biplot, `method` is the PCA procedure and `ax.tickvec.D` is for the tick markers per variable axis in the PCA biplot.

This function also gives the overall quality value of the biplot (`overall.quality`), the axis and sample predictivity values (`axis.pred` and `sample.pred`), as well as the approximated values of the data (`D.hat`). The PCA biplot of the glass data is obtained using

```
Dmat = matrix(glass,nc=13)
dimnames(Dmat) = list(1:180, paste(c("Na2O", "MgO", "Al2O3",
                        "SiO2", "P2O5", "SO3", "Cl", "K2O",
                         "CaO", "MnO", "Fe2O3", "BaO", "PbO")))
PCA.biplot(D=Dmat, method=mod.PCA, ax.tickvec.D=rep(5,
              ncol(Dmat)))
```

and is shown in Figure 8.15. The output is shown below.

```
$overall.quality
[1] 0.997

$axis.pred
 Na2O     MgO    Al2O3    SiO2          .       .       .      PbO
 0.990    0.952   0.977   0.999          .       .       .      0.611

$sample.pred
   1    2           .       .       .      180
   1    1           .       .       .       1

$D.hat
      Na2O     MgO          .       .       .       PbO
 1   15.246    2.46         .       .       .      -0.167
 2   15.026    2.43         .       .       .      -0.104
       .
                                 .                      .
 180  11.187    2.71         .       .       .       0.223
```

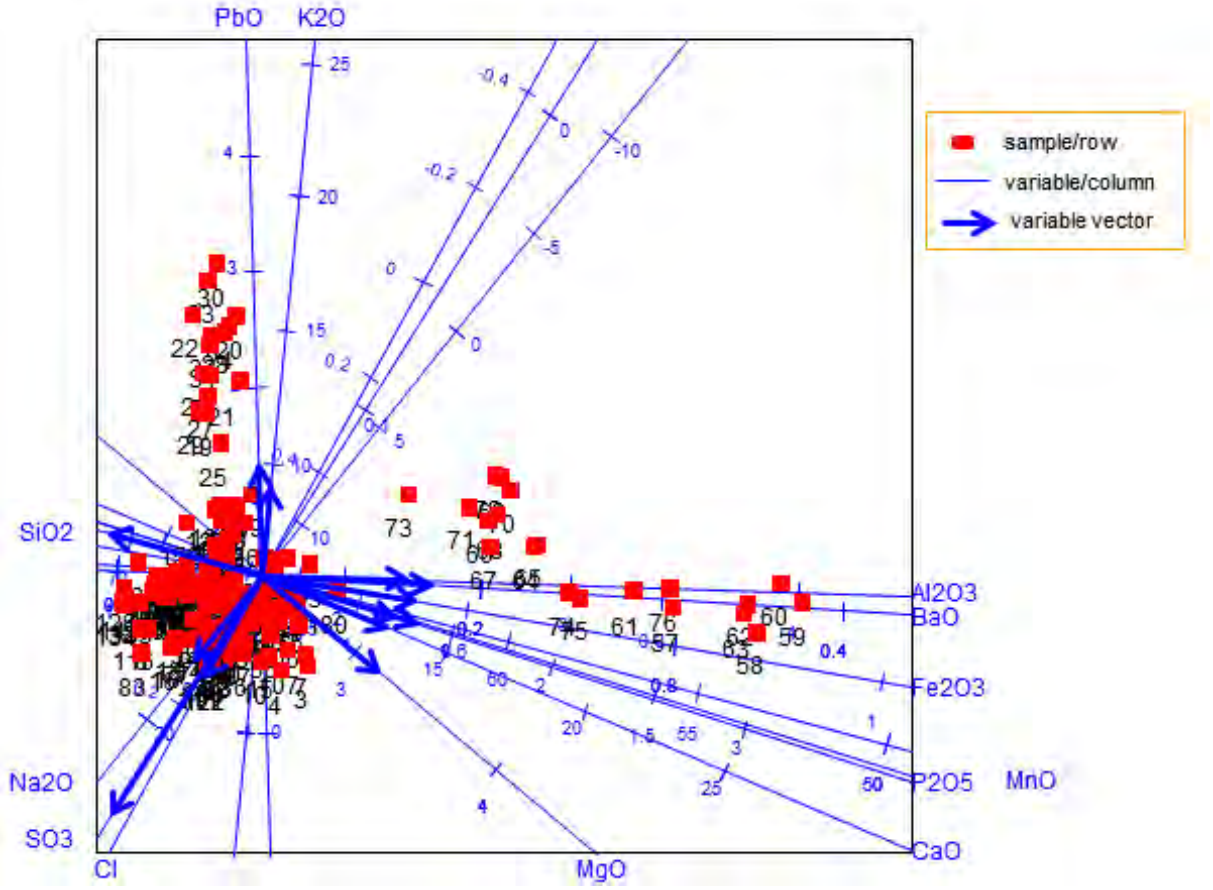

Figure 8.15 The PCA biplot of the glass data.

In the obtained biplot display (Figure 8.15), a representation of the variance of each variable is provided. Observing the length of the thicker arrows (vectors), SO3 can be said to have a large

standard deviation, while K2O has a small deviation. With an overall quality value (`overall.quality`) of 0.997, the following relationships can be deduced: a relation between measurements Na2O, SO3 and Cl; between PbO and K2O; between SiO2, MnO, P2O5 and CaO; as well as between Al2O3, BaO and Fe2O3. However, MgO can be seen to have no clear relation with the others. From the axis predictivity values (`axis.pred`), each of the biplot axes quite well represents the original data, with the SiO2 axis having the highest axis predictivity value of 0.999, followed by the Na2O axis with 0.990. An axis predictivity of 1.000 means that all values can be read off the axis exactly. The lower the axis predictivity value, the less accurately the axis approximates the observed values under that variable. The PbO axis has the lowest predictivity value of 0.611. This means that the axis represents the original data, but not quite as well as the other axes. Although the full output is not printed out here, users are encouraged to print out the whole output in order to see these deductions. Furthermore, from the sample predictivity values (`sample.pred`), all one hundred and eighty samples are in the biplot plane of approximation. These predictivities along with the overall quality value indicate that the PCA biplot approximates the glass data well.

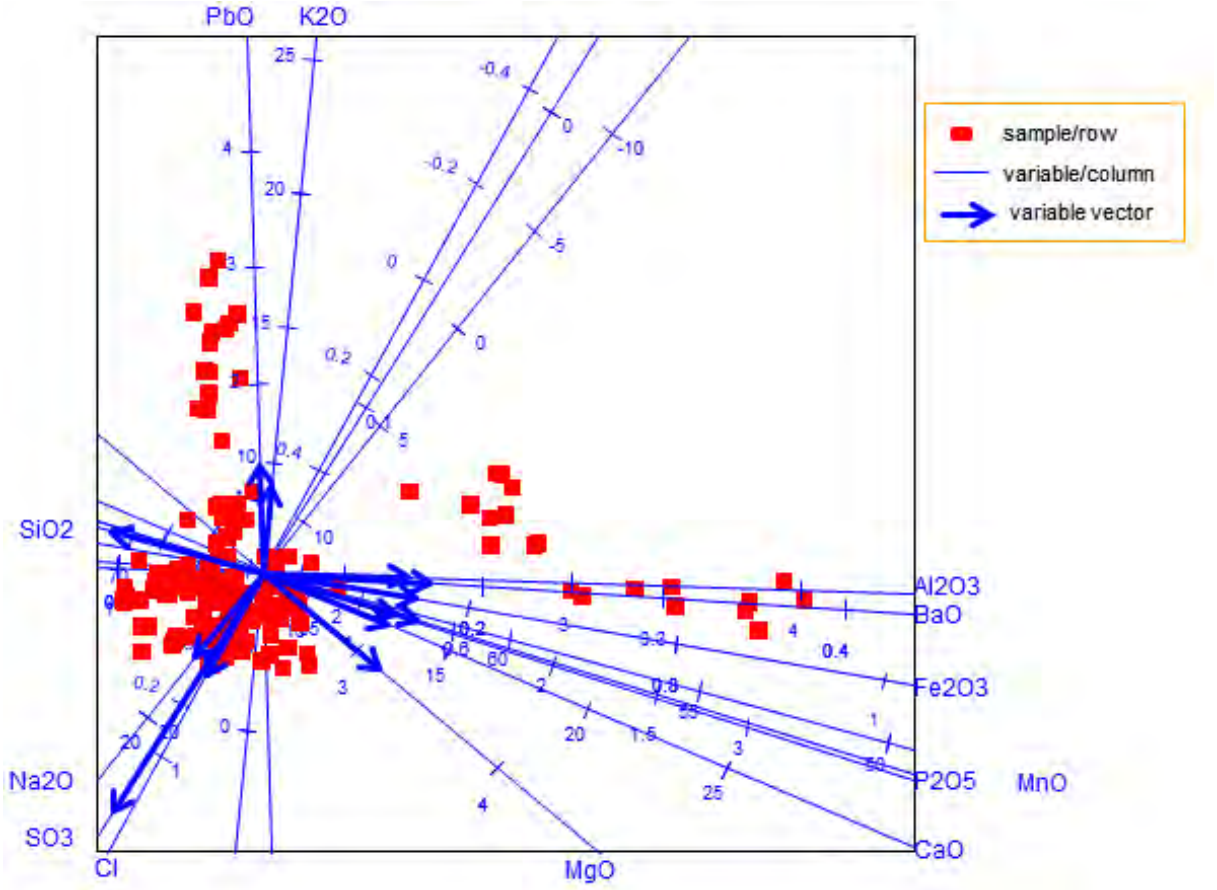

Figure 8.16 The PCA biplot of the glass data, with no sample names.

One can also get a PCA biplot, where sample names have been excluded. For this situation

```
PCA.biplot_no.SN (D=Dmat, method=mod.PCA,
                  ax.tickvec.D=rep(5,ncol(Dmat)))
```

is used. The resulting biplot display is shown above in Figure 8.16.

8.5.4 *PLSR*

The PLSR analysis of a data set is obtained using any of the algorithms discussed in Section 3.4. In the **PLSbiplot1** package, this can be achieved by using these functions:

`mod.NIPALS(X,Y,A)` for the NIPALS algorithm,

`mod.KernelPLS_R(X,Y,A)` for the kernel algorithm of Rännar *et al*. (1994), where there are few(er) variables,

`mod.KernelPLS_L(X,Y,A)` for the kernel algorithm of Lindgren *et al*. (1993), where there are few(er) samples,

`mod.SIMPLS(X,Y,A)` for the SIMPLS algorithm.

In these functions, `X` is the matrix of predictors, `Y` the matrix of responses and `A` the number of PLS components. Each function gives an output comprising of the RMSEP values (`RMSEP`), **T** (`X.scores`), **R** (`X.weights.trans`), **P** (`X.loadings`), **Q** (`Y.loadings`) and $\hat{\mathbf{Y}}_{\text{PLSR}}$ (`Y.hat`) matrices.

Consider the nutrimouse data. Using the SIMPLS algorithm, the PLSR analysis of this data is done by

```
X1 = as.matrix(nutrimouse$lipid, ncol=21)
Y1 = as.matrix(nutrimouse$gene, ncol=120)
mod.SIMPLS(X=X1, Y=Y1, A=17)
```

and the following output is obtained.

```
$X.scores
    Comp 1   Comp 2       .      .      .     Comp 17
1   0.1644   0.1582       .      .      .     0.00304
2   0.0331   0.0815       .      .      .     0.02840
       .
                          .
                                              .
40 -0.1938   0.0879       .      .      .     0.08456

$X.weights.trans
          Comp 1    Comp 2       .      .      .   Comp 17
C14.0    -0.00726  -0.0613       .      .      .    0.0199
C16.0     0.40629   0.1482       .      .      .   -0.2806
       .
                                  .
                                                    .
C22.6n.3  0.25970   0.4559       .      .      .   -0.0435

$X.loadings
          Comp 1   Comp 2        .      .      .      Comp 17
C14.0       1.33   -3.98         .      .      .      -2.92
C16.0      19.48    5.17         .      .      .      -2.60
       .
                                  .
```

```
                                                                    .
C22.6n.3   11.15   24.54          .      .      .      13.12

$Y.loadings
          Comp 1   Comp 2         .      .      .     Comp 17
X36b4      0.0674  -0.0456        .      .      .    -0.0719
ACAT1     -0.0147  -0.0514        .      .      .    -0.0939
       .
                                  .                      .
mHMGCoAS  0.5208   0.1426         .      .      .    -0.1720

$Y.hat
, , 1 Comps
    X36b4   ACAT1    .      .      .  mHMGCoAS
1  -0.444  -0.658    .      .      .  -0.135
2  -0.453  -0.656    .      .      .  -0.204
       .
                     .                    .
40 -0.468  -0.652    .      .      .  -0.322

, , 2 Comps
    X36b4   ACAT1    .      .      .  mHMGCoAS
1  -0.451  -0.666    .      .      .  -0.113
2  -0.457  -0.660    .      .      .  -0.192
       .
                     .                    .

40 -0.472  -0.657    .      .      .  -0.309

 .                     .                    .
 .                     .                    .
 .                     .                    .

, , 17 Comps
    X36b4   ACAT1    .      .      .  mHMGCoAS
1  -0.497  -0.632    .      .      .  -0.0650
2  -0.451  -0.685    .      .      .  -0.0952
       .
                     .                    .

40 -0.509  -0.687    .      .      .  -0.4720

$RMSEP
1.006  0.943  0.911  0.890  0.864       .      .      .   0.598
```

The choice of `A=17` is chosen arbitrarily, as the initial number of the PLS components. Thus, to determine the final number of components (`A.final`) to use for the PLSR model, a plot of the RMSEP is examined. This is done using

```
RMSEP = mod.SIMPLS(X=X1, Y=Y1, A=17)$RMSEP
plot(t(RMSEP), type = "b", xlab="Number of components",
      ylab="RMSEP values")
```

Looking at the elbow of the RMSEP plot in Figure 8.17 below, nine components, i.e., $A = 9$, can be suggested as the number of components to use in the final PLSR analysis for the nutrimouse data. Therefore,

```
A.final = 9 #from the RMSEP plot in Figure 8.17
mod.SIMPLS(X=X1, Y=Y1, A=A.final)
```

gives the PLSR analysis of the nutrimouse data, using the SIMPLS algorithm.

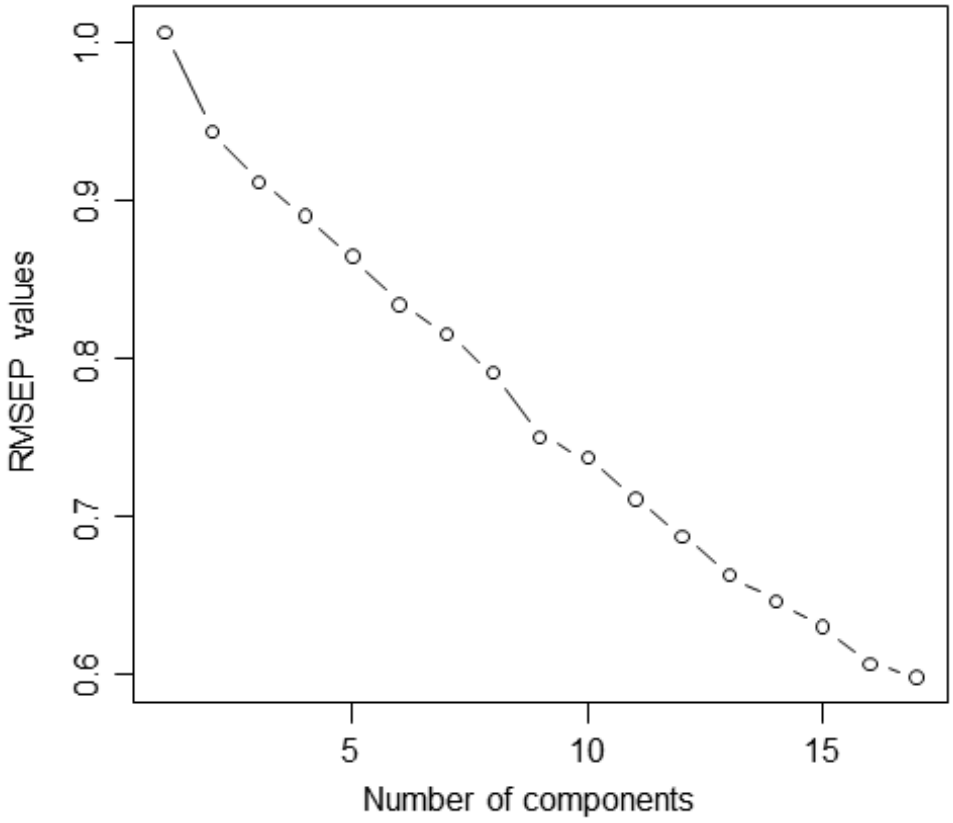

Figure 8.17 The RMSEP plot of the nutrimouse data.

Sometimes, the number of X-variables in the data set can be large. For this reason, the VIP analysis (Section 3.9) of the X-variables is necessary, to identify which variables are important. In the **PLSbiplot1** package, the VIP analysis is performed by using

```
mod.VIP(X, Y, algorithm=mod.SIMPLS, A=A.final, cutoff=0.80)
```

where `A=A.final` is the final number of PLS components, decided from the RMSEP plot and `cutoff` is the desired cut off value to use for selecting the important variables. Here, `cutoff=0.80` means that the X-variables are identified as important if their respective VIP value is greater than or equal to 0.80, i.e., VIP $\geq$ 0.80.

The VIP function gives the VIP values (`VIP.values`) for each X-variables and the important variables (`X.impor`) in its output. Although the number of X-variables in the nutrimouse data are not large, the output obtained from the VIP analysis performed, using `mod.VIP` with `cutoff=0.80`, is shown below. Out of the twenty one X-variables, only sixteen variables were identified as important. In this output, under the `$X.impor`, both the important X-variables and their respective column number in the **X** matrix are shown. For example, the first, third, fourth,

fifth, sixth, eighth and ninth X-variables are identified as important, followed by the thirteenth, fourteenth, sixteenth, seventeenth, eighteenth, nineteenth, twentieth and twenty first X-variables.

```
$VIP.values
C14.0   C16.0   C18.0   C16.1n.9  C16.1n.7  C18.1n.9  C18.1n.7  C20.1n.9
 1.012   0.644   0.974   1.115     1.073     0.843     0.738     1.571
C20.3n.9  C18.2n.6 C18.3n.6 C20.2n.6 C20.3n.6 C20.4n.6 C22.4n.6 C22.5n.6
   1.058   0.657    1.194    0.597    1.112    1.059    0.737    0.918
C18.3n.3  C20.3n.3  C20.5n.3  C22.5n.3  C22.6n.3
   1.268    1.034    0.840     0.927     1.084

$X.impor
C14.0  C18.0  C16.1n.9  C16.1n.7  C18.1n.9  C20.1n.9  C20.3n.9  C18.3n.6
  1      3       4         5         6         8         9        11
C20.3n.6  C20.4n.6 C22.5n.6 C18.3n.3 C20.3n.3 C20.5n.3 C22.5n.3 C22.6n.3
  13        14       16        17       18       19       20        21
```

To print out the newly reduced X-variables matrix $\mathbf{X}_{\text{new}}$, use

```
X.important = mod.VIP(X=X1, Y=Y1, algorithm=mod.SIMPLS, A=
                          A.final, cutoff=0.80)$X.impor
X.new = X1[,c(X.important)]  #important X-variables
ncol(X.new) #=16
colnames(X.new)
```

Using `cutoff=0.75` gives an `X.impor` output identical to the one obtained when `cutoff=0.80` was used.

```
X.important_B = mod.VIP(X=X1, Y=Y1, algorithm=mod.SIMPLS, A=
                          A.final, cutoff=0.75)$X.impor
ncol(X1[,c(X.important_B)]) #=16
```

This is not always the case for all data sets, seeing as the first VIP analysis employs VIP ≥ 0.80 to select the important variables, while the second analysis employs VIP ≥ 0.75. However, since none of the X-variables in the nutrimouse data had a VIP value (`VIP.values`) between 0.75 and 0.80, as shown above under `$VIP.values`, the same number of important variables (`X.new`) will be obtained, when `cutoff=0.80` and `cutoff=0.75` are used.

Furthermore, to see which PLSR coefficients are influential in the analysis, using the SIMPLS algorithm,

```
Mag.Bmat.plot(X=X.new, Y1, algorithm=mod.SIMPLS, A=A.final)
```

is used. This function gives a plot of the magnitude of the absolute values of the coefficients matrix ($\mathbf{B}_{\text{PLSR}}$). A small magnitude indicates that the coefficient is not quite influential. For the nutrimouse data, using the newly reduced X-variables, the following figure is obtained. Alternatively, one can use

```
X.scal = scale(X.new, center = TRUE, scale = TRUE)
Y.scal = scale(Y1, center = TRUE, scale = TRUE)
main3 = mod.SIMPLS(X.scal, Y.scal, A.final)
Bmat = main3$X.weights.trans %*% t(main3$Y.loadings)
#PLSR coefficient matrix
dimnames(Bmat) = list(colnames(X.new), colnames(Y))
Abs.Bmat = abs(Bmat)  #absolute values of the coefficients
```

```
rowMeans(Abs.Bmat)
```

to deduce the influential coefficients, with the following output.

```
C14.0    C18.0  C16.1n.9  C16.1n.7  C18.1n.9  C20.1n.9  C20.3n.9
0.439    1.306    2.387     0.576     2.008     3.970     2.343
C18.3n.6  C20.3n.6  C20.4n.6  C22.5n.6  C18.3n.3  C20.3n.3  C20.5n.3
 1.788     1.982     1.135     1.206     1.881     1.030     1.003
C22.5n.3  C22.6n.3
  0.485     0.622
```

From this output, as well as Figure 8.18 below, coefficient **b**6 (C20.1n.9) can be said to have the highest influence (3.970), followed by **b**3 (C16.1n.9) and **b**7 (C20.3n.9) with values 2.387 and 2.343 respectively. The lowest influence is observed by **b**1 (C14.0), with a value of 0.439.

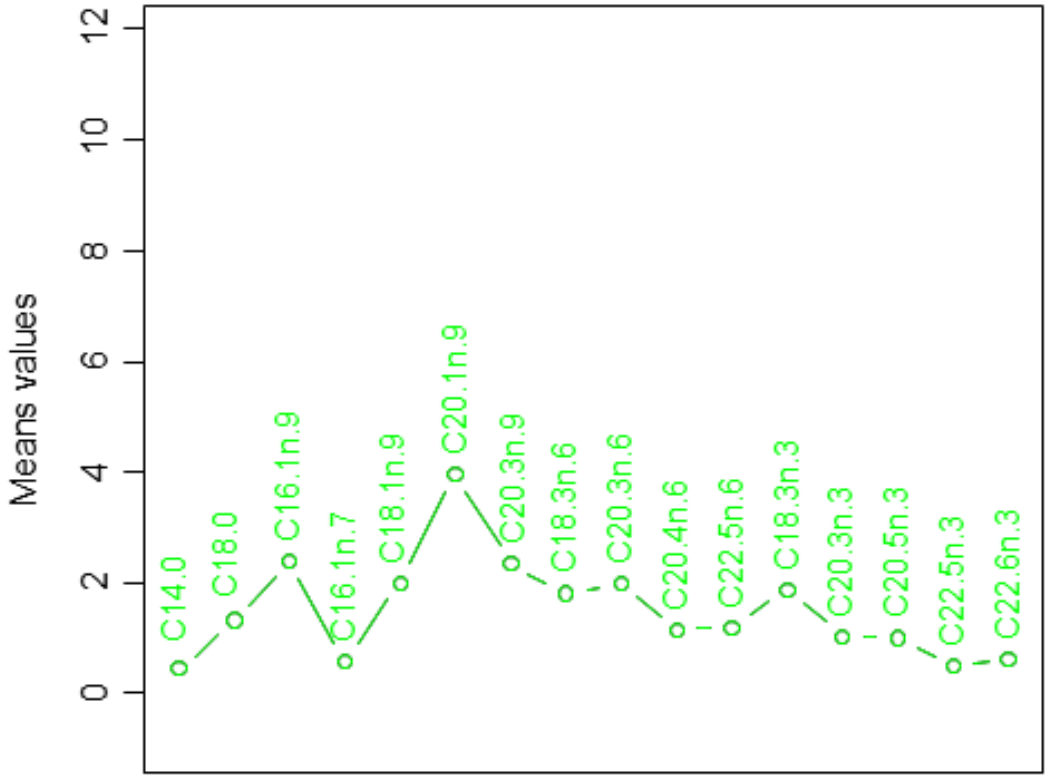

Figure 8.18 Mean plot of the absolute PLSR coefficients of the nutrimouse data.

The following influential coefficients can be deduced, with their respective magnitudes given in parenthesis.

b1: C14.0 (lowest)
b2: C18.0 (moderate)
b3: C16.1n.9 (high)
b4: C16.1n.7 (low)
b5: C18.1n.9 (high)
b6: C20.1n.9 (highest)
b7: C20.3n.9 (high)
b8: C18.3n.6 (high)
b9: C20.3n.6 (high)
b10: C20.4n.6 (moderate)

b11: C22.5n.6 (moderate)
b12: C18.3n.3 (high)
b13: C20.3n.3 (moderate)
b14: C20.5n.3 (moderate)
b15: C22.5n.3 (lower)
b16: C22.6n.3 (low).

In addition, the MMLR and PCR analyses of the nutrimouse data is obtained using functions

```
mod.MMLR(X=X.new, Y1)
mod.PCR(X=X.new, Y1, r=2)
```

respectively, where `r` is the desired number of PCA component.

8.5.5 *Covariance monoplot and biplot*

The covariance monoplot of one set of variables in a data is obtained using

```
cov.monoplot(Y)
```

This function gives the row and column markers matrices **G** (`G__VD`) and **H** (`H__VD`). Consider the cocktail data. The covariance monoplot of the Y-variables is shown in Figure 8.19 and the accompanying output is shown below. This monoplot is obtained using

```
Y3 = as.matrix(senso.cocktail, nc=13)
cov.monoplot(Y3)
    $G__VD
                      Comp 1  Comp 2
    color.intensity  -0.533 -1.0310
    odor.intensity    1.215 -0.9866
    odor.orange      -1.977  1.0876
    odor.banana       2.453 -0.0940
    odor.mango       -1.726 -1.2291
    odor.lemon       -1.223 -1.3571
    strongness       -1.913 -0.6901
    sweet             2.451 -0.3782
    acidity          -2.452 -0.2678
    bitterness       -2.327  0.0841
    persistence      -1.364 -1.1710
    pulp              1.844 -0.7879
    thickness         2.324 -0.7528

    $H__VD
                      Comp 1  Comp 2
    color.intensity  -0.533 -1.0310
    odor.intensity    1.215 -0.9866
    odor.orange      -1.977  1.0876
    odor.banana       2.453 -0.0940
    odor.mango       -1.726 -1.2291
    odor.lemon       -1.223 -1.3571
    strongness       -1.913 -0.6901
    sweet             2.451 -0.3782
    acidity          -2.452 -0.2678
    bitterness       -2.327  0.0841
    persistence      -1.364 -1.1710
    pulp              1.844 -0.7879
    thickness         2.324 -0.7528
```

As discussed in Chapter 4, see Section 4.2, the row and column markers of a covariance monoplot are the same, i.e., $\mathbf{G} = \mathbf{H}$. Hence, `G__VD` and `H__VD` have the same values in the above

output. Comparing the length of the thicker green arrows (vectors) to each other in Figure 8.19, odor.lemon can be said to have a small standard deviation.

Furthermore, several relationships can be deduced from this monoplot. For example, the relation between thickness, pulp and odor.orange; between sweet, bitterness, odor.banana and acidity; as well as between color.intensity, persistence and odor.mango can be observed. Seeing as only variables are being represented in the covariance monoplot and there are no samples to (orthogonally) project onto the monoplot axes, calibration markers are not necessary on these axes.

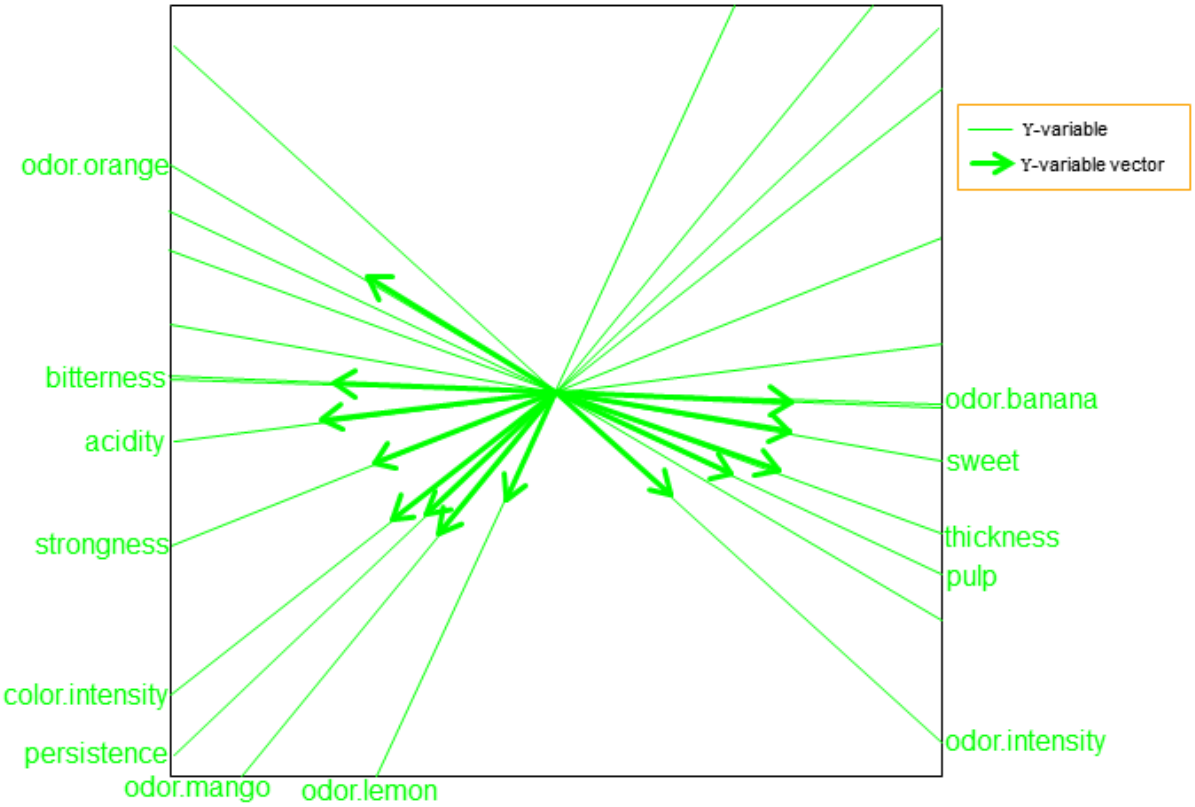

Figure 8.19 The covariance monoplot of the sensory panel descriptors.

For the covariance biplot, use `cov.biplot(X,Y)` as follows.

```
X3 = as.matrix(compo.cocktail, nc=4)
cov.biplot(X3,Y3)
```

The resulting display is shown below in Figure 8.20. The covariance biplot function gives the row and column markers matrices **G** (`G__UDhalf`) and **H** (`H__VDhalf`) of the data used, for $\alpha = 0.5$ (see Section 4.3). As discussed in this section, two different sets of variables (`X=X3` and `Y=Y3`) are involved in the covariance biplot. Therefore, the row and column markers of a covariance biplot will not be the same, i.e., $\mathbf{G} \neq \mathbf{H}$. Hence, `G__UDhalf` and `H__VDhalf`

obtained from `cov.biplot(X3,Y3)` will have different values in the output. Here, matrix **G** is for the X-variables, while matrix **H** is for the Y-variables.

```
$G__UDhalf
        Comp 1  Comp 2
orange -1.0729  -0.664
banana  1.1504  -0.124
mango   0.0747   0.596
lemon  -0.6981   0.879

$H__VDhalf
                  Comp 1  Comp 2
color.intensity -0.0728  0.0238
odor.intensity   0.3269  0.4546
odor.orange     -0.6048 -0.3758
odor.banana      0.6248 -0.0470
odor.mango      -0.2929  0.4541
odor.lemon      -0.1862  0.6782
strongness      -0.3973  0.4408
sweet            0.6166  0.0742
acidity         -0.5540  0.2462
bitterness      -0.5580  0.0321
persistence     -0.2308  0.4331
pulp             0.5740  0.2618
thickness        0.6633  0.2228
```

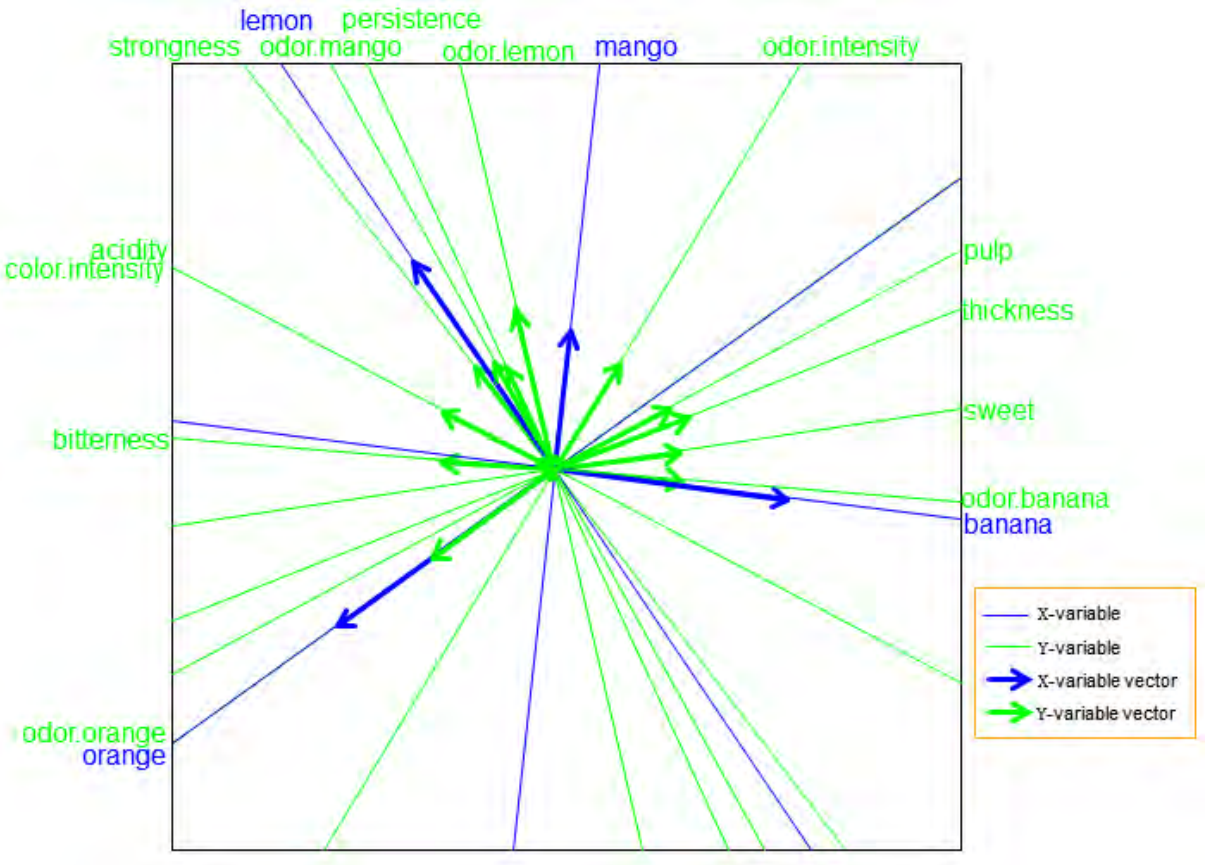

Figure 8.20 The covariance biplot of the cocktail data.

Observing the length of the thicker arrows in this biplot display (Figure 8.20), banana, orange and lemon can be said to have a large standard deviation, followed by odor.orange. In addition, the relation between strongness, lemon, odor.mango, persistence and odor.lemon; between

odor.orange, orange, pulp and thickness; between acidity and color.intensity; as well as between banana, odor.banana, bitterness and sweet can be seen. However, mango and odor.intensity can be seen to have no clear relation with the others. The actual correlation values of this data can be obtained using `cor(X3,Y3)`. Analogous to the covariance monoplot, calibration markers are not necessary on the axes in the covariance biplot, seeing as only variables are being represented in the biplot and there are no samples to project onto the biplot axes.

If only one set of variables is used in the `cov.biplot(X,Y)` function, the resulting display will be a monoplot display of that set of variables. For example, `cov.biplot(Y3,Y3)` will give a monoplot display of `Y3`, just like Figure 8.19 above, and matrices **G** (`G__UDhalf`) and **H** (`H__VDhalf`) will have the same output.

8.5.6 *PLS biplot*

From the **PLSbiplot1 package**, the PLS biplot of a data set can be obtained using the function

```
PLS.biplot(X, Y, algorithm, ax.tickvec.X, ax.tickvec.Y)
```

where `algorithm` is any of the PLS algorithms "`mod.NIPALS`", "`mod.KernelPLS_R`", "`mod.KernelPLS_L`" and "`mod.SIMPLS`", `ax.tickvec.X` and `ax.tickvec.Y` are for the tick markers of the X- and Y-variables axes respectively. The `PLS.biplot` function also gives the overall quality value of the biplot (`overall.quality`), the axis predictivity (`axis.pred`), the approximated values of the data (`D.hat`) and the PLSR coefficients values (`Bmat`). For example, using the SIMPLS algorithm and the cocktail data, the PLS biplot is shown in Figure 8.21. Below is its accompanying output:

```
$overall.quality
[1] 0.972

$axis.pred
 orange  banana  mango .  .  .  Strongness .  .  .  persistence thickness
  0.999   0.903  0.796 .  .  .    0.997    .  .  .    0.997        0.997

$D.hat
   orange   banana         .      .      .   thickness
1  0.796    0.862          .      .      .    5.96
2  0.896    1.774          .      .      .    7.03
       .
                           .                      .
16  2.678   0.707          .      .      .      4.20

$Bmat
            color.intensity   odor.intensity   .  .  .  thickness
b1:orange       0.1931           -1.980         .  .  .   -2.593
b2:banana      -0.2932            0.949        .  .  .    2.396
b3:mango        0.0422            0.896        .  .  .    0.428
b4:lemon        0.2655            0.616        .  .  .   -1.064
```

This biplot (Figure 8.21), also showing a representation of the variance of each variable, is obtained using

```
PLS.biplot(X=X3, Y=Y3, algorithm=mod.SIMPLS, ax.tickvec.X=
             rep(2,ncol(X3)), ax.tickvec.Y=rep(3,ncol(Y3)))
```

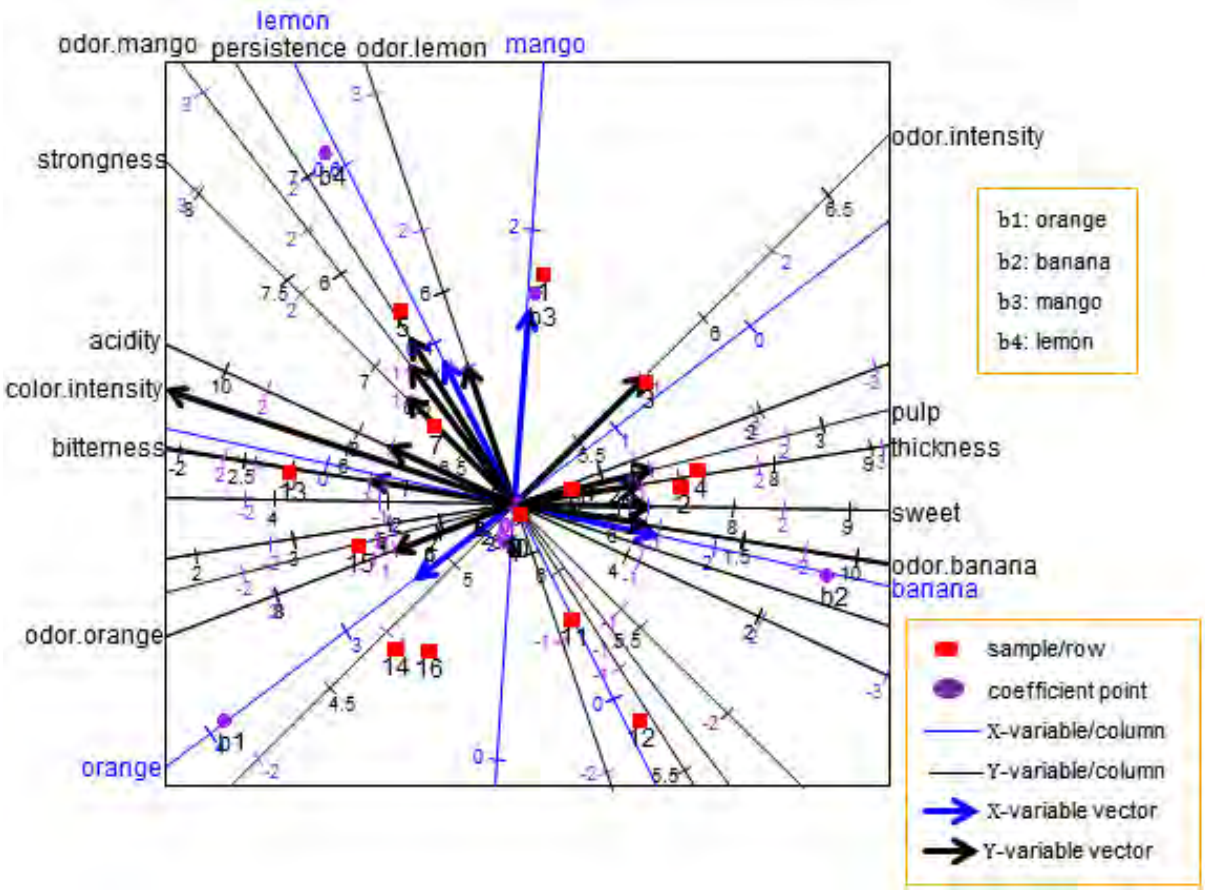

Figure 8.21 The PLS biplot of the cocktail data.

With an overall quality value (`overall.quality`) of 0.997, several relationships can be observed from this biplot. For example, a relation between strongness, odor.mango, persistence, lemon and odor.lemon; between orange and odor.intensity; between odor.orange, thickness and pulp; as well as between sweet, bitterness, odor.banana, banana, color.intensity and acidity can be observed. However, mango can be seen to have no clear relation with the others.

In addition, looking at the axis predictivity values (`axis.pred`), each of the biplot axes quite well represents the original data, with the orange axis having the highest axis predictivity of 0.999, followed by the strongness, persistence and thickness axes with 0.997. However, the mango axis has the lowest axis predictivity of 0.796. This means that the axis represents the original data, but not quite as well as the other axes. Although the full output was not printed out here, to observe these deductions, users are encouraged to see the full output by typing

```
PLS.biplot(X=X3, Y=Y3, algorithm=mod.SIMPLS, ax.tickvec.X=c(8,5,
```

```
5,5,5), ax.tickvec.Y=c(5,8,5,6,9,8))
```

in the R console. These predictivities along with the overall quality value indicate that the PLS biplot approximates the cocktail data well.

To obtain a PLS biplot where the sample names have been excluded, use

```
PLS.biplot_no.SN(X=X3, Y=Y3, algorithm=mod.SIMPLS,
                 ax.tickvec.X=rep(2,ncol(X3)),
                  ax.tickvec.Y=rep(3,ncol(Y3)))
```

An example of this is shown below in Figure 8.22, for the cocktail data.

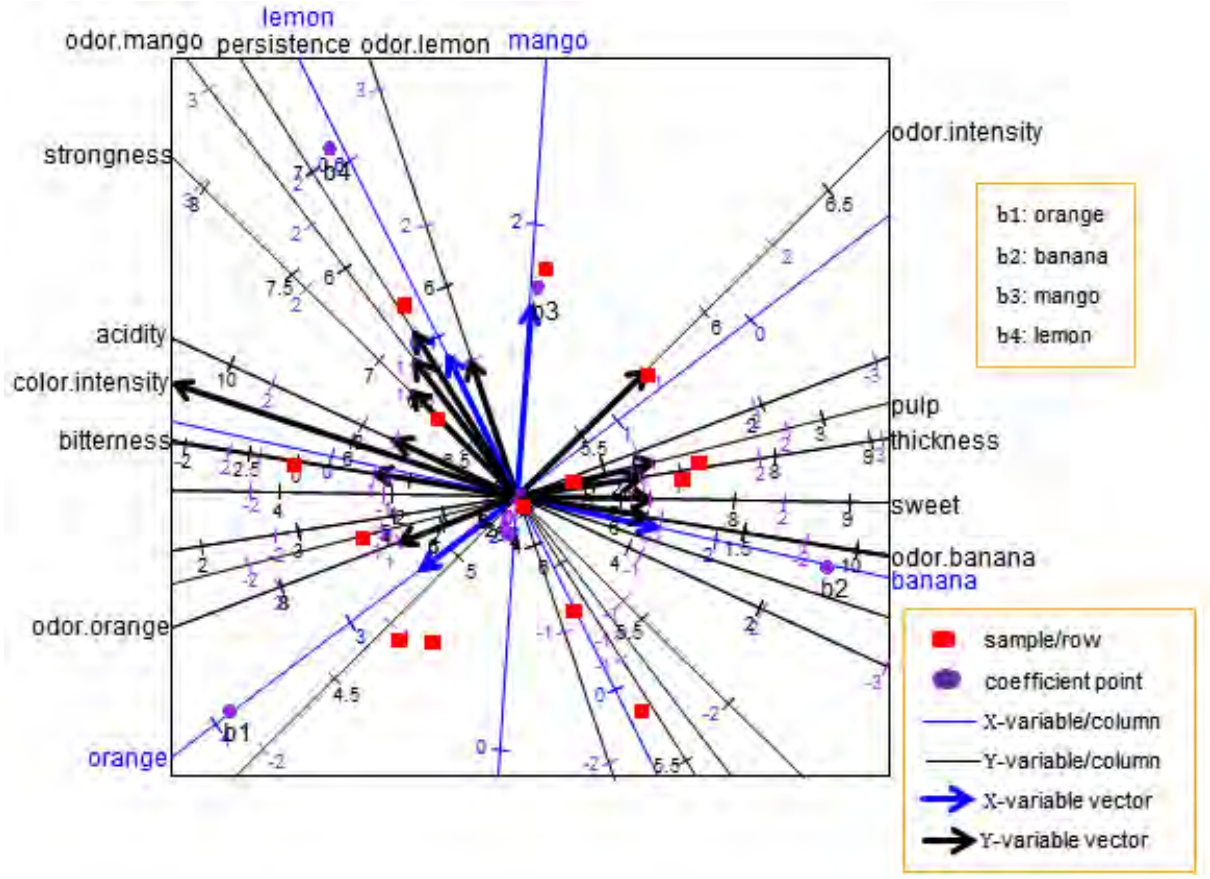

Figure 8.22 The PLS biplot of the cocktail data, with no sample names.

Alternatively, for a PLS biplot where the labels of the samples, coefficient points and tick markers have been excluded, use

```
PLS.biplot_no_labels(X3, Y3, algorithm=mod.SIMPLS, ax.tickvec.X=
                     rep(1,ncol(X3)), ax.tickvec.Y=rep(1,ncol(
                      Y3)))
```

An example is shown below in Figure 8.23. This display can be useful, if the data set under consideration is large and emphases are on exploring the relationships between the variables.

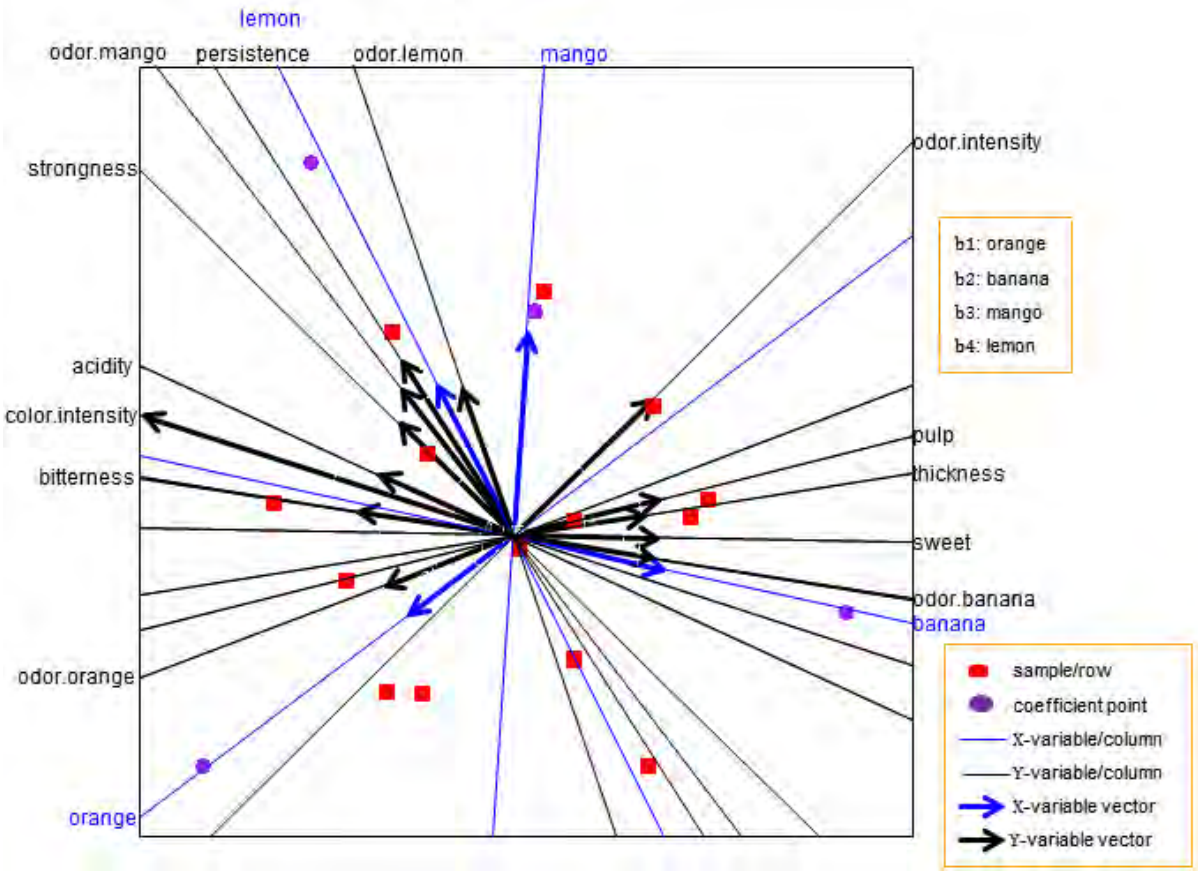

Figure 8.23 The PLS biplot of the cocktail data, without sample, coefficient points and tick markers labels.

Furthermore, besides orthogonally projecting each of the coefficient points bi (for $i = 1, 2, 3, 4$) in Figure 8.21 onto the axes representing the Y-variables, to get their respective values, the area biplot method can be utilized. In the **PLSbiplot1** package, the area biplot method is incorporated in the PLS biplot, for the coefficient values, through

```
PLS.biplot.area(X, Y, algorithm, ax.tickvec.X, ax.tickvec.Y,
                base.tri, bi.value)
```

Here, `base.tri` is the position number of the desired Y-variable in the **Y** matrix, to use as the base for the triangle, while `bi.value` is the desired rotated coefficient points bi to approximate using the area biplot method. An example of the area biplot method incorporated in the PLS biplot is shown in Figure 8.24, using the cocktail data.

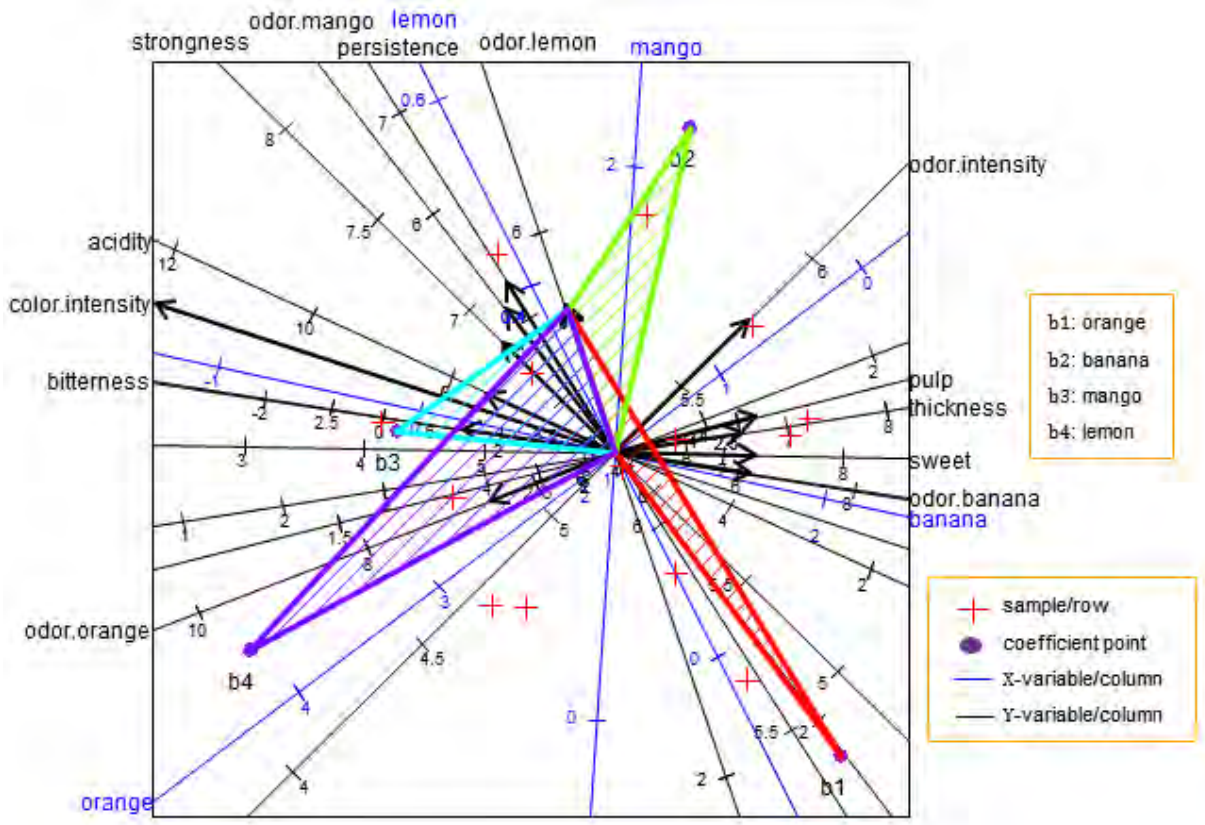

Figure 8.24 The triangles for points bi, for $i = 1, 2, 3, 4$, with base defined by the odor.lemon axis in the PLS biplot of the cocktail data.

Figure 8.24 shows an illustration of the area biplot methodology to estimate coefficient points bi, for $i = 1, 2, 3, 4$, under response odor.lemon. The resulting biplot is obtained using

```
PLS.biplot.area(X3, Y3, algorithm=mod.SIMPLS, ax.tickvec.X=
                    rep(2,ncol(X3)),  ax.tickvec.Y=rep(3,ncol(Y3)),
                    base.tri=6, bi.value=c(1:ncol(X3)))
```

and the following output is obtained.

```
$Bmat
          color.intensity   odor.intensity    .  .  .  thickness
b1:orange       0.1931           -1.980       .  .  .   -2.593
b2:banana      -0.2932            0.949       .  .  .    2.396
b3:mango        0.0422            0.896       .  .  .    0.428
b4:lemon        0.2655            0.616       .  .  .   -1.064
```

It is not intuitive to estimate the exact area of a triangle visually, but as an exploratory tool, larger and smaller coefficients can be easily discerned. Large triangles indicate large coefficient values, while small triangles indicate small coefficient values. From Figure 8.24, triangle b4 can be seen to have a large coefficient value under the odor.lemon variable, followed by b3 and b2. However, triangle b1 can be said to have a small coefficient value. The exact coefficient values are obtained by printing out the coefficient matrix $\widehat{\mathbf{B}}_{\text{PLSR}}$ (`Bmat`). Although not fully printed out here, using the `PLS.biplot.area` output, these deductions can be confirmed by looking at the coefficient values obtained under the odor.lemon variable, in `$Bmat`. In the full

output, it can be seen that b4 has a larger value of 2.713, while b1 has a small value of 0.759. The sign of each coefficient value gives an indication of the effect direction on the response variables. Here, b1 and b2 have a negative effect on the odor.lemon variable, while b3 and b4 have a positive effect.

8.5.7 *PLS biplot for GLM*

To get the PLS biplot for the PLS-GLM analysis of a data,

```
PLS.GLM.biplot(X, y, algorithm=PLS.GLM, ax.tickvec.X,
                  ax.tickvec.y, ax.tickvec.b)
```

is used. For $M > 1$ Y-variables, the Y-variables are used individually and separately in the PLS-GLM analysis. Here, `y` is one of the Y-variables, `algorithm=PLS.GLM` is the PLS-GLM algorithm, while `ax.tickvec.X` and `ax.tickvec.y` are for the tick markers of the X-variables and Y-variable axes respectively. The `ax.tickvec.b` is for the purple tick markers on the Y-variable axis, used for reading the coefficient points.

For the PLS biplot of a PLS-GLM fitted using the SIMPLS algorithm,

```
PLS.GLM.biplot_SIMPLS(X, y, algorithm=PLS.GLM_SIMPLS,
                           ax.tickvec.X, ax.tickvec.y,
                            ax.tickvec.b)
```

is used.

Both the `PLS.GLM.biplot` and `PLS.GLM.biplot_SIMPLS` functions give the approximated values of the data (`D.hat`) and the coefficient vector (`b.vec`) in their respective output. Consider the spider data. The PLS biplot of a Poisson PLS-GLM of this data, using species Trocterr, is shown in Figure 8.25 and the accompanying output is shown below.

```
$D.hat
    soil.dry  bare.sand    .      .       . Expected_Trocterr
S1     1.945     1.929     .      .       .   27.74
S2     2.760     0.672     .      .       .   43.97
       .
                           .
                                             .
S28    0.757     3.826     .      .       .  -14.21

$bvec
                  Trocterr
b1:soil.dry        0.2728
b2:bare.sand      -0.1562
b3:fallen.leaves  -0.1310
b4:moss           -0.1971
b5:herb.layer      0.3364
b6:reflection     -0.0151
```

Looking at the coefficient values (`bvec`) in the output, all the environmental characteristics can be seen to have a negative effect on species Trocterr, except for soil.dry and herb.layer.

Characteristic herb.layer can be said to have a high effect on Trocterr, followed by soil.dry, moss, bare.sand and fallen.leaves. However, reflection can be said to have a low effect on Trocterr.

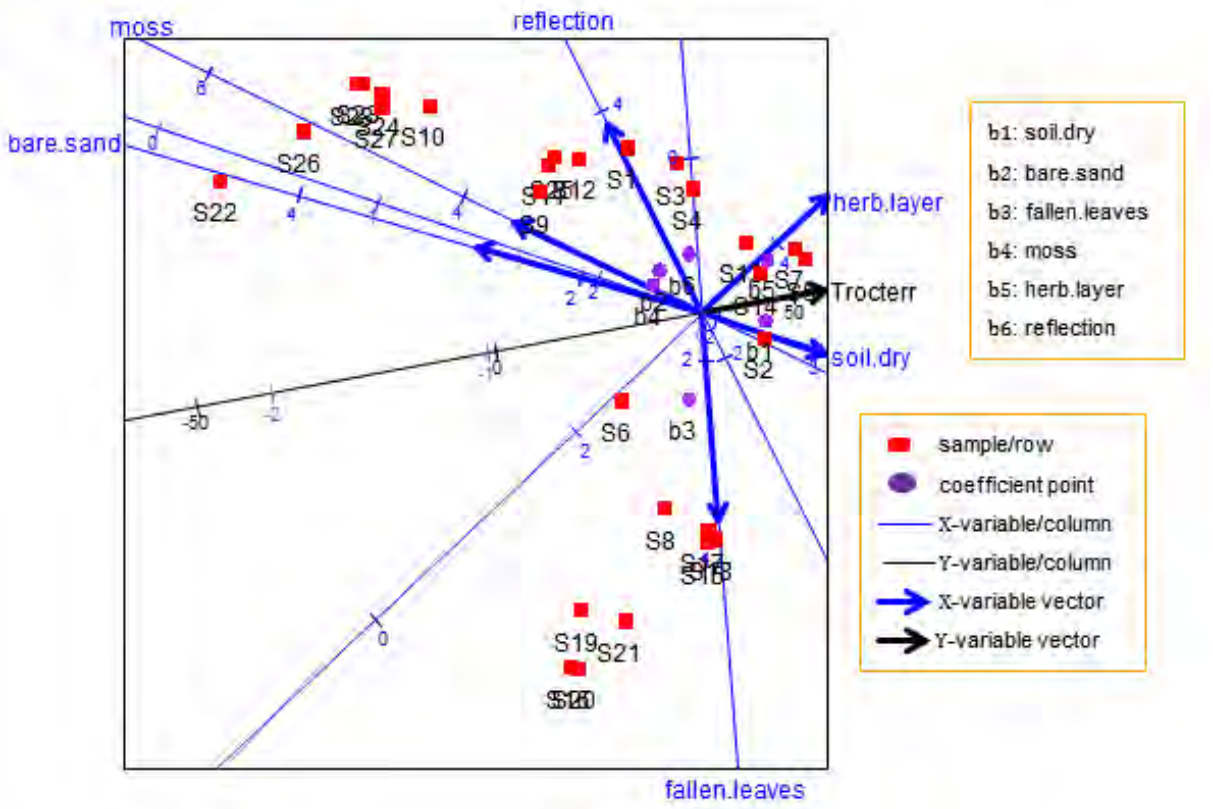

Figure 8.25 The PLS biplot for a Poisson PLS-GLM for species Trocterr of the spider data.

This display is obtained using

```
X2 = as.matrix(cbind(spider$x))
rownames(X2) = paste(c("S1","S2","S3","S4","S5","S6","S7","S8",
                       "S9","S10","S11","S12","S13","S14","S15",
                       "S16","S17","S18","S19","S20","S21",
                       "S22","S23","S24","S25","S26","S27",
                       "S28"))
Y2 = as.matrix(cbind(spider$abund))
rownames(Y2) = rownames(X2)
which.y.variable = 11  #specifying which y-variable to use
y = as.matrix(Y2[,which.y.variable])
dimnames(y) = list(rownames(Y2), colnames(Y2)[
                   which.y.variable])
PLS.GLM.biplot(X=X2, y, algorithm=PLS.GLM, ax.tickvec.X=
                 rep(2,ncol(X2)),ax.tickvec.y=3, ax.tickvec.b=3)
```

In Figure 8.25, a representation of the variance of each variable is shown. Observing the length of the thicker arrows (vectors), reflection can be said to have a large standard deviation, followed by fallen.leaves and bare.sand. In addition, observing the angles between the blue vectors, all the environmental characteristics can be said to be positively related to each other,

except for fallen.leaves. Also, the relation between species Trocterr and environmental characteristics soil.dry, moss, bare.sand and herb.layer can be seen.

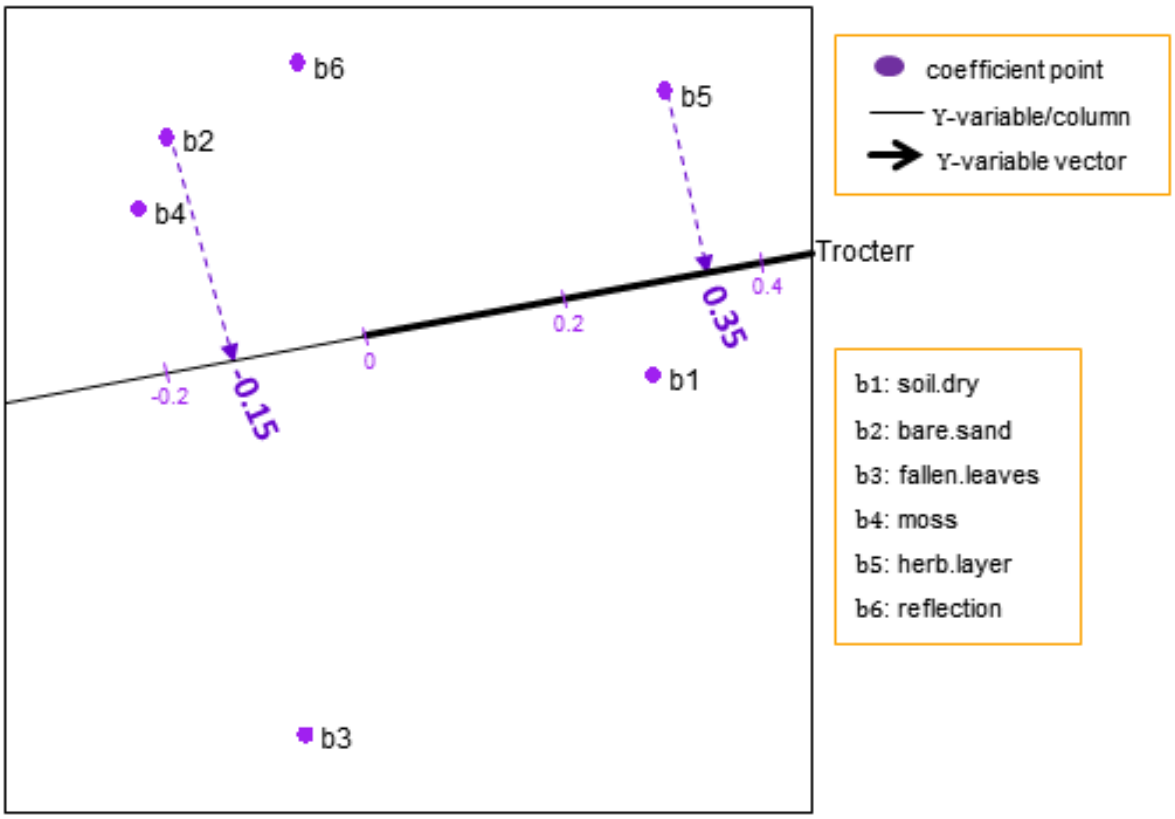

Figure 8.26 A zoomed-in display of the coefficient points in the PLS biplot a Poisson PLS-GLM for species Trocterr of the spider data.

A zoomed-in display of the coefficient points in Figure 8.25 is obtained using

```
PLS.GLM.biplot_bvec(X=X2, y, algorithm=PLS.GLM, ax.tickvec.b=15)
```

and the resulting output is shown in Figure 8.26 above. This can be used for easier orthogonal projections of the coefficient points. For example, coefficient points b2 and b5 projected orthogonally onto the Trocterr axis yields -0.15 and 0.35 respectively, as shown above in Figure 8.26. The obtained coefficient values are shown under `$bvec` in the output of the `PLS.GLM.biplot` function.

Figure 8.27 shows the display obtained when the SIMPLS-fitted Poisson PLS-GLM is used

```
PLS.GLM.biplot(X=X2, y, algorithm=PLS.GLM_SIMPLS, ax.tickvec.X=
                  rep(2,ncol(X2)),ax.tickvec.y=3, ax.tickvec.b=3)

$D.hat
    soil.dry  bare.sand    .      .       . Expected_Trocterr
S1     1.919     1.745     .      .       .   27.64
S2     2.716     0.756     .      .       .   46.58
        .
                         .
                                          .
S28    0.934     3.308     .      .       .  -25.49
```

```
$bvec
                  Trocterr
b1:soil.dry        2.556
b2:bare.sand      -2.276
b3:fallen.leaves  -1.205
b4:moss           -2.367
b5:herb.layer      3.677
b6:reflection     -0.097
```

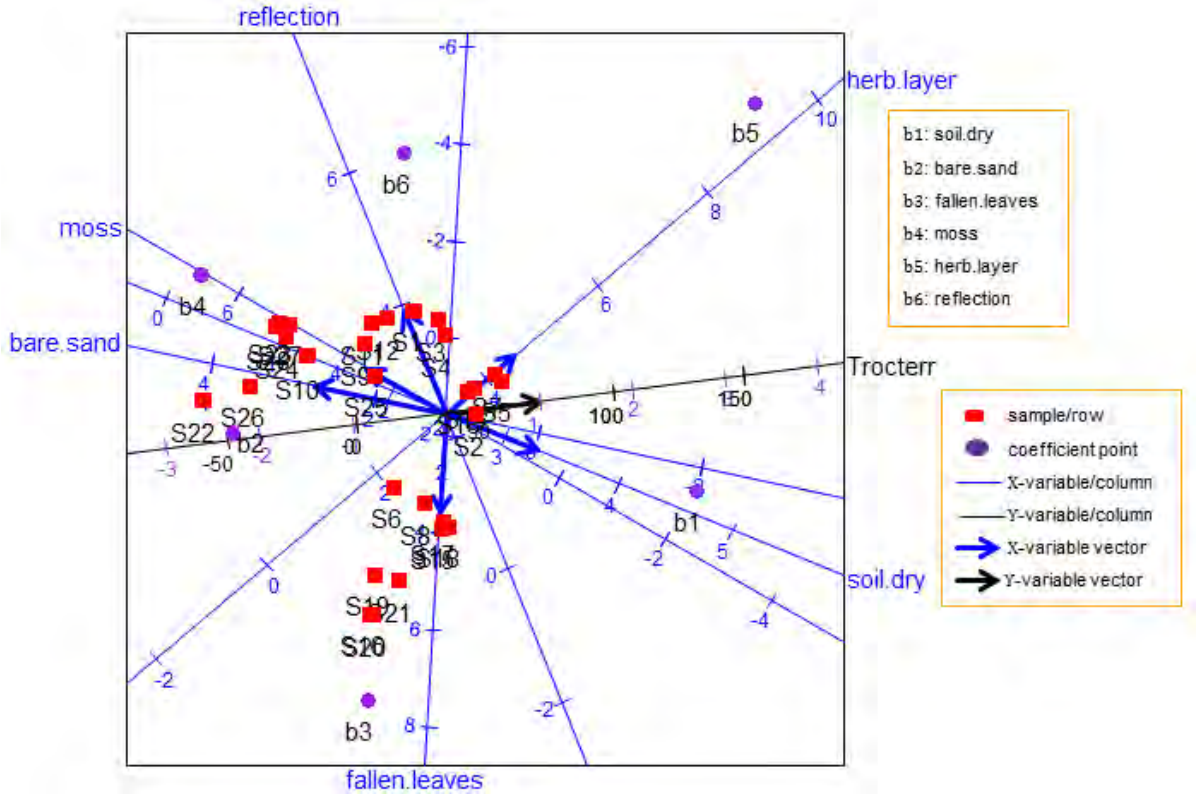

Figure 8.27 The PLS biplot for a Poisson PLS-GLM for species Trocterr of the spider data, fitted using the SIMPLS algorithm (Algorithm 6.3).

Comparing this PLS biplot (Figure 8.27) to that shown in Figure 8.25, slightly similar deductions can be observed. For example, in Figure 8.27, the relation between species Trocterr and environmental characteristics bare.sand, soil.dry and moss can be seen. This can also be observed in Figure 8.25. However, the positions of the coefficients points $\mathrm{b}i$, for $i = 1, 2, 3, 4, 5, 6$, and sample points are different in both biplots. This is due to the method used in constructing the X-weights matrix $\mathbf{R}_*$ in both algorithms, see Algorithms 6.2 and 6.3. The predicted coefficient values are shown under `bvec`, in their respective output.

In the **PLSbiplot1** package, only the Poisson PLS-GLM (`PLS.GLM`) and the Binomial PLS-GLM (`PLS.binomial.GLM`) were developed, as discussed in Chapter 6. To obtain a PLS biplot for a Binomial PLS-GLM, use

```
PLS.GLM.biplot(X, y, algorithm=PLS.binomial.GLM, ax.tickvec.X,
                     ax.tickvec.y, ax.tickvec.b)
```

and for the SIMPLS-fitted Binomial PLS-GLM, use

```
PLS.GLM.biplot_SIMPLS(X, y, algorithm=PLS.binomial.GLM,
                          ax.tickvec.X, ax.tickvec.y, ax.tickvec.b)
```

An application of the Binomial PLS-GLM function can be seen in Figures 8.6 to 8.8 of Section 8.3, obtained using the following.

```
data(Pima.tr, package="MASS")
X = as.matrix(cbind(Pima.tr[,1:7]))
dimnames(X) = list(1:nrow(X), colnames(X))
y = as.matrix(as.numeric(Pima.tr$type)-1, ncol=1)
#0=No and 1=Yes
dimnames(y) = list(1:nrow(y), paste("type"))
#Figure 8.6
PLS.GLM.biplot(X, y, algorithm=PLS.binomial.GLM, ax.tickvec.X=
                c(3,3,8,7,8,5,2), ax.tickvec.y=3, ax.tickvec.b=3)
#Figure 8.7
PLS.GLM.biplot_no.SN(X, y, algorithm=PLS.binomial.GLM,
                      ax.tickvec.X=c(3,3,8,7,8,5,2),
                       ax.tickvec.y=3, ax.tickvec.b=3)
#A zoomed-in display of the coefficient points (Figure 8.8)
PLS.GLM.biplot_bvec(X, y, algorithm=PLS.binomial.GLM,
                     ax.tickvec.b=10)
```

To apply the SIMPLS-fitted Binomial PLS-GLM function to the Pima.tr data, use

```
PLS.GLM.biplot_SIMPLS(X, y, algorithm=PLS.binomial.GLM,
                        ax.tickvec.X=c(3,5,8,4,3,7,2),
                         ax.tickvec.y=2, ax.tickvec.b=3)
#no sample names
PLS.GLM.biplot_SIMPLS_no.SN(X, y, algorithm=PLS.binomial.GLM,
                             ax.tickvec.X=c(3,5,8,4,3,7,2),
                              ax.tickvec.y=2, ax.tickvec.b=3)
```

8.5.8 *Biplot for SPLS*

The PLS biplot of a SPLS analysis of a data set is obtained using

```
SPLS.biplot(X, Y, algorithm=mod.SPLS, lambdaX, lambdaY, eps,
              ax.tickvec.X, ax.tickvec.Y)
```

Here, `algorithm=mod.SPLS` is the SPLS algorithm, `eps` is for the convergence step in the SPLS algorithm (see Algorithm 7.1), `lambdaX` and `lambdaY` are the desired penalty parameters for the soft-thresholding penalization function for the PLS X-weights and Y-weights respectively. The `SPLS.biplot` function also gives the overall quality value of the biplot (`overall.quality`), the axis predictivity (`axis.pred`), the approximated values of the data (`D.hat`) and the SPLS coefficients values (`Bmat`).

Consider the ash data. Prior to the SPLS analysis, the eight elemental compositions were subjected to a log-transformation, due to their skewed distribution. For this result, the log-transformed elements are used in the SPLS analysis. To choose a value for the penalty

parameters (λ_X and λ_Y) for this data, the experiment described in Subsection 7.4.1 is conducted using Algorithm 7.1. Since $M = 1$, there is no need to perform variables selection on the $M = 1$ Y-variable. As a result, $\lambda_Y = 0$ in this experiment, and much attention is given to the value of λ_X that gives the lowest RMSEP value. Here, different pairs of $\lambda_X \in (0, 500)$ and $\lambda_Y = 0$ are used as follows.

```
X1 = as.matrix(ash[,10:17], nc=8)
Y1 = as.matrix(ash$SOT)
colnames(Y1) = paste("SOT")
#choosing a value for the penalty parameters lambdaX and lambdaY
#for the ash data
main2 = opt.penalty.values(X=scale(X1), Y=scale(Y1), A=2,
                           algorithm=mod.SPLS, eps=1e-5,
                             from.value.X=0, to.value.X=500,
                              from.value.Y=0, to.value.Y=0,
                               lambdaY.len=1, lambdaX.len=100)
main2
$RMSEP.values
    lambdaY    lambdaX RMSEP.value
1         0   0.000000   0.7723511
2         0   5.050505   0.7710647
3         0  10.101010   0.7697669
              .
              .
              .
100       0 500.000000   0.9949367

$min.RMSEP.value
[1] 0.7697669

$lambdaY.to.use
[1] 0

$lambdaX.to.use
[1] 10.10101
```

As discussed in Subsection 7.4.1, the value for λ_X and λ_Y is chosen so that a minimum RMSEP is obtained. To achieved this, 100 different λ_X values and $\lambda_Y = 0$ were paired together, to form the $L = 100$ pairs (λ_X, λ_Y). Using the $L = 100$ pairs (λ_X, λ_Y) and the `opt.penalty.values` function, the value for λ_X and λ_Y, to use in the final SPLS analysis, is obtained. The `opt.penalty.values` function performs the experiment described in Subsection 7.4.1, for the data under consideration. In this function, `from.value.X` and `to.value.X` are the upper and lower limits for λ_X. Likewise, `from.value.Y` and `to.value.Y` are the upper and lower limits for λ_Y. In addition, `lambdaX.len` and `lambdaY.len` are the total number of λ_X and λ_Y to use in the experiment. Therefore,

`from.value.X=0, to.value.X=500` and `lambdaX.len=100` means 100 (different) values of $\lambda_X \in (0, 500)$, while `from.value.Y=0, to.value.Y=0` and `lambdaY.len=1` means $\lambda_Y = 0$.

Furthermore, in the `opt.penalty.values` function, the RMSEP values obtained from the L SPLS analyses are recorded under `RMSEP.values`, the minimum RMSEP value is recorded under `min.RMSEP.value`, while the value of λ_X and λ_Y having the minimum RMSEP value is recorded under `lambdaX.to.use` and `lambdaY.to.use` respectively. Note, `algorithm` in `opt.penalty.values` can be the SPLS algorithm (`mod.SPLS`) or the SPLS-GLM algorithm (`SPLS.GLM` or `SPLS.binomial.GLM`) and `eps` is for the convergence step of the SPLS/SPLS-GLM algorithm. For the ash data, a plot of the obtained RMSEP values is shown below in Figure 8.28, along with their respective λ_X value.

```
main2$RMSEP.values
plot(main2$RMSEP.values[,2], main2$RMSEP.values[,3], type='l',
      xlab="lambdaX", ylab="RMSEP values")
```

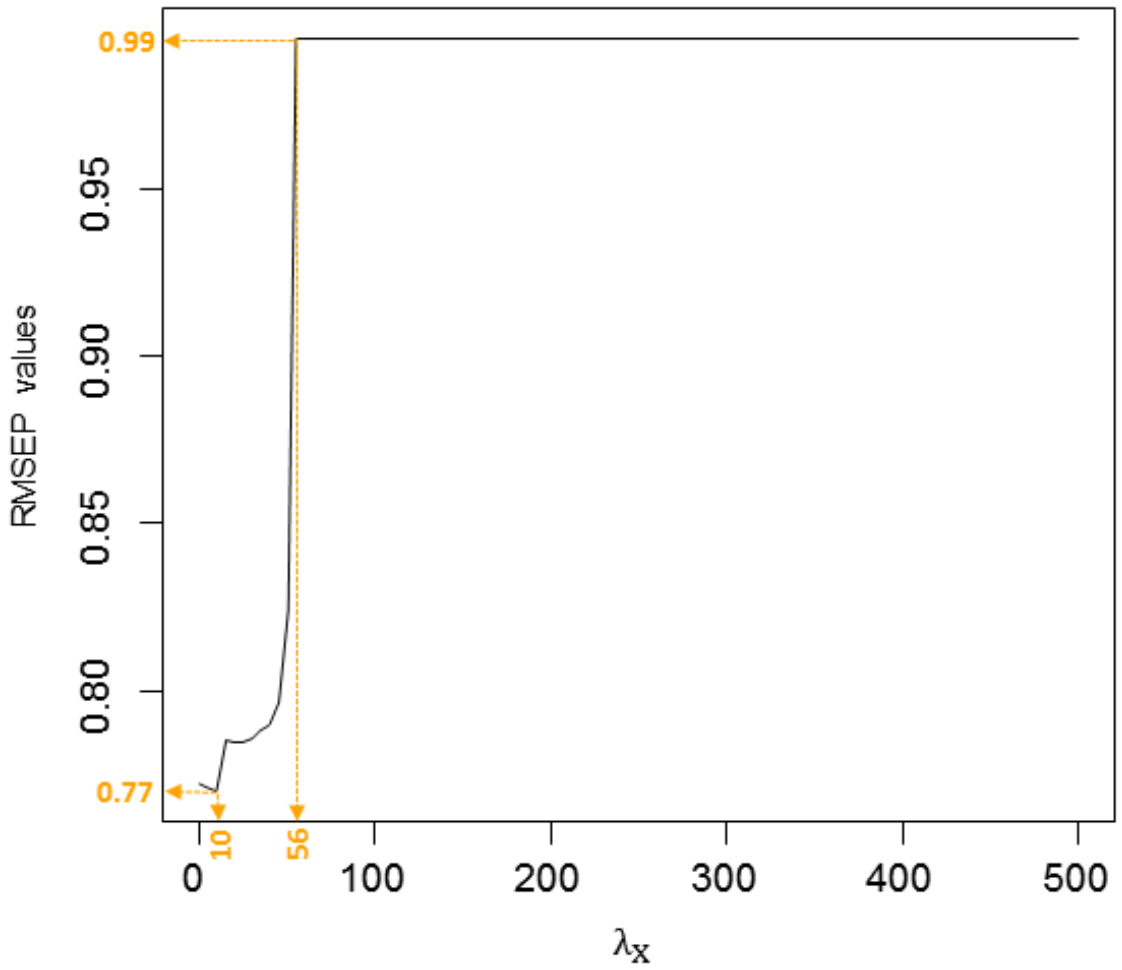

Figure 8.28 A plot of $\lambda_X \in (0, 500)$ values and their respective RMSEP value, for the ash data.

In Figure 8.28, the RMSEP value starts off at a value of 0.77, when $\lambda_X = 0$, and decreases to 0.77, at $\lambda_X = 10$. As the value for λ_X further increases, so does the RMSEP value until $\lambda_X = 56$ when it becomes constant. At this point, the RMSEP value is 0.99 and stays constant until the last value of λ_X (i.e., $\lambda_X = 500$) is reached in this experiment. From this plot, the minimum RMSEP value of 0.77 is observed at $\lambda_X = 10$. Thus, for the SPLS analysis of the ash data, $\lambda_X = 10$ and $\lambda_Y = 0$. Alternatively, from the `opt.penalty.values` output below, the minimum RMSEP value of 0.76977 occurred when $\lambda_X = 10.101$ and $\lambda_Y = 0$. Thus, these penalty values will be used for the SPLS analysis of the ash data.

```
min.RMSEP.value = main2$min.RMSEP.value
lambdaY.to.use = main2$lambdaY.to.use
lambdaX.to.use = main2$lambdaX.to.use
list(lambdaY.to.use=lambdaY.to.use, lambdaX.to.use=
      lambdaX.to.use, min.RMSEP.value=min.RMSEP.value)
$lambdaY.to.use
[1] 0
$lambdaX.to.use
[1] 10.101
$min.RMSEP.value
[1] 0.76977
```

With SPLS/SPLS-GLM also performing variables selection, besides components extraction, in the final SPLS analysis, all X-variables having a non-zero weight value are selected and used in the analysis. Likewise, all Y-variables having a non-zero weight value are selected and used in the analysis. However, if the number of Y-variables is small, then there is no need to perform variables selection on the Y-variables. Thus, $\lambda_Y = 0$. Similarly, if the number of X-variables is small, there is no need to perform variables selection on these X-variables, therefore, $\lambda_X = 0$. When $\lambda_X = 0 = \lambda_Y$, then no variables selection is needed and no thresholding is applied in the SPLS analysis. For the ash data, using the obtained penalty parameters ($\lambda_X = 10$ and $\lambda_Y = 0$), the variables selection procedure is obtained as follows.

```
#which X- and Y-variables to use for the SPLS analysis
main3 = mod.SPLS(X=scale(X1), Y=scale(Y1), A=2, lambdaY=
                  lambdaY.to.use, lambdaX=lambdaX.to.use, eps=
                   1e-5)

X.to.use = main3$X.select
Y.to.use = main3$Y.select  #not necessary for the ash data
X.new = as.matrix(X1[,X.to.use])
colnames(X.new) #P=6
[1] "log(P2O5)"  "log(Fe2O3)" "log(Al2O3)" "log(CaO)"
[5] "log(Na2O)"  "log(K2O)"

colnames(X1) #P=8
[1] "log(P2O5)" "log(SiO2)" "log(Fe2O3)" "log(Al2O3)"
[5] "log(CaO)" "log(MgO)" "log(Na2O)" "log(K2O)"

Y.new = as.matrix(Y1[,Y.to.use])
colnames(Y.new) = colnames(Y1)
```

```
colnames(Y.new)
[1] "SOT"
```

From $P = 8$ X-variables, in the final SPLS analysis, this value was reduced to $P = 6$, as seen above under `colnames(X.new)`. With $M = 1$, there is no need to perform variables selection on the Y-variable, thus, `Y.new=Y1`. Using the obtained penalty parameters ($\lambda_X = 10$ and $\lambda_Y = 0$), the resulting PLS biplot is obtained as follows.

```
SPLS.biplot(X.new, Y.new, algorithm=mod.SPLS, lambdaY=
            lambdaY.to.use, lambdaX=lambdaX.to.use, eps=1e-5,
             ax.tickvec.X=rep(1,ncol(X.new)), ax.tickvec.Y=
              rep(5,ncol(Y.new)))
```

The biplot display is shown below in Figure 8.29.

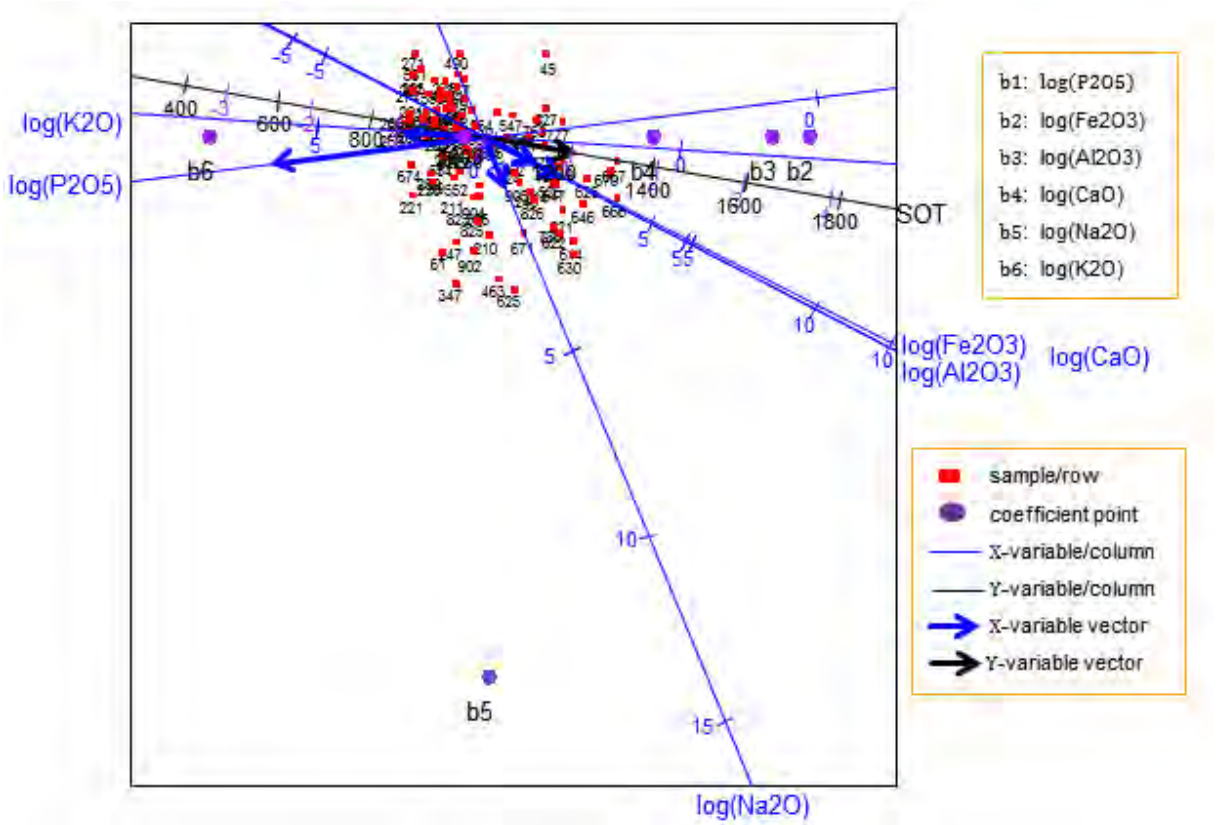

Figure 8.29 The PLS biplot for a SPLS of the ash data, with $\lambda_X = 10$ and $\lambda_Y = 0$.

Below is the resulting output from the `SPLS.biplot` function.

```
$overall.quality
 [1] 0.983

$axis.pred
  log(P2O5)          .       .       .   SOT
   0.736             .       .       .   0.982

$D.hat
       log(P2O5)   log(Fe2O3)       .       .       .   SOT
13      1.674        -1.3368       .       .       .   935
14      1.611        -1.2965       .       .       .   953
          .
                          .
                                                  .
```

```
905    1.323          0.9434        .        .        .    1136

$Bmat
                    SOT
b1:log(P2O5)   -0.239
b2:log(Fe2O3)   3.640
b3:log(Al2O3)   3.228
b4:log(CaO)     1.893
b5:log(Na2O)    1.096
b6:log(K2O)    -3.089
```

In this biplot display (Figure 8.29), a representation of the variance of each variable is provided. Observing the length of the thicker arrows (vectors), log(P2O5) can be said to have a large standard deviation. In addition, looking at the coefficient values under `$Bmat` above, all the compositions can be seen to have a positive effect on SOT, except for log(P2O5) and log(K2O). Composition log(Fe2O3) can be said to have a high effect on SOT, followed by log(Al2O3) and log(K2O). However, composition log(P2O5) can be said to have a low effect on SOT. With an overall quality (`overall.quality`) of 0.983, the relation between SOT and log(K2O) and log(P2O5); as well as between log(CaO), log(Fe2O3) and log(Al2O3) can be observed in the biplot (Figure 8.29). On the other hand, log(Na2O) can be seen to have no clear relation with the others.

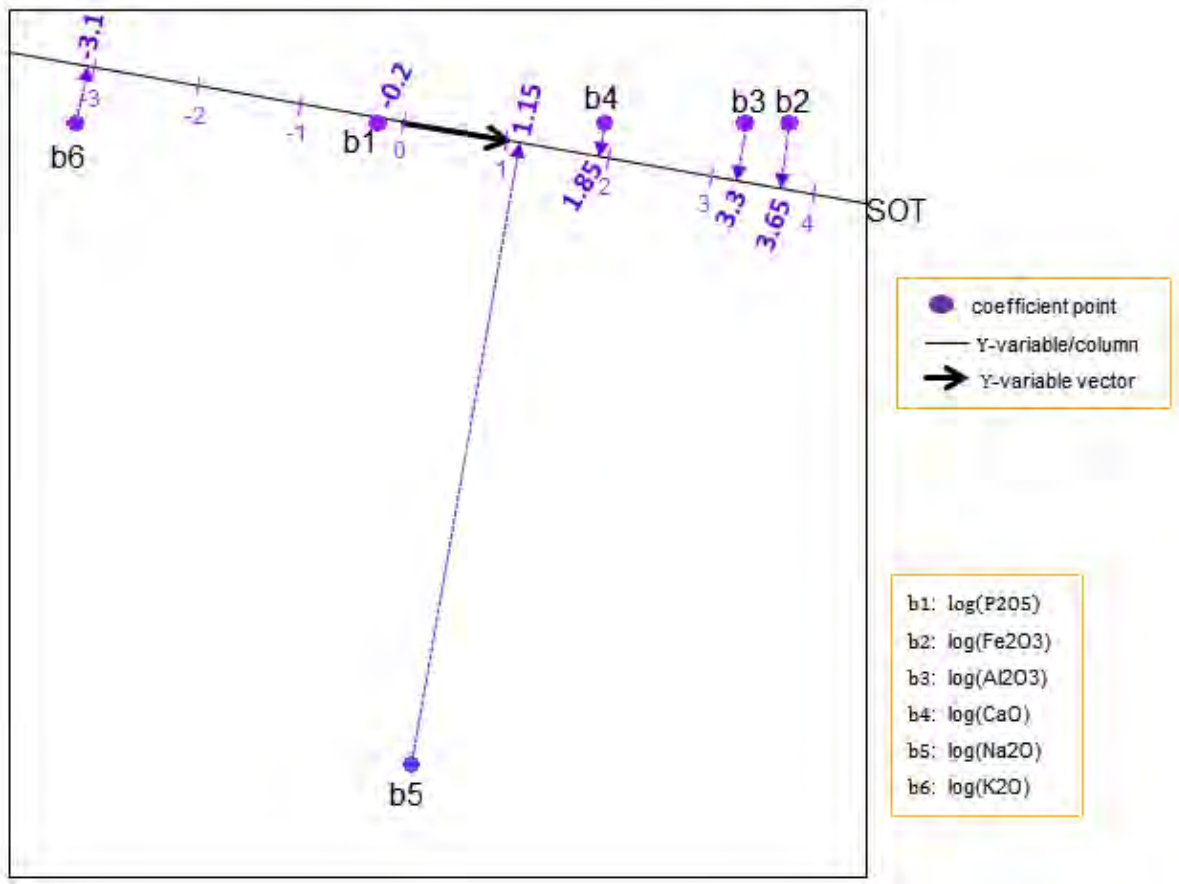

Figure 8.30 A display of the coefficient points in the PLS biplot for a SPLS of the ash data, with $\lambda_X = 10$ and $\lambda_Y = 0$.

Orthogonally projecting each of the coefficient points bi in Figure 8.29, for $i = 1, 2, \cdots, 6$, onto the SOT axis yields the coefficient values. As discussed in Subsection 5.2.3 and Section 7.6, *the purple markers on the SOT axis* are used to read off these values. A display of the coefficient points in Figure 8.29 is shown above in Figure 8.30. This display can be used for easier orthogonal projections of the coefficient points bi, for $i = 1, 2, \cdots, 6$, onto the SOT axis, to get their respective values. For example, coefficient points b1, b2, b3, b4, b5, and b6 projected orthogonally onto the SOT axis yields -0.2, 3.65, 3.3, 1.85, 1.15 and -3.1 respectively, as shown in Figure 8.30. The obtained coefficient values are shown under `$Bmat`, in the `SPLS.biplot` output above. This display (Figure 8.30) is obtained using

```
SPLS.biplot_Bmat(X.new, Y.new, algorithm=mod.SPLS, lambdaY=
                 lambdaY.to.use, lambdaX=lambdaX.to.use,
                 eps=1e-5, ax.tickvec.B=5)
```

Here, `lambdaX.to.use=10` and `lambdaY.to.use=0` is obtained from the `opt.penalty.values` output above.

For a display where the labels of the samples, coefficient points and tick markers have been excluded (Figure 8.31), use

```
SPLS.biplot_no_labels(X.new, Y.new, algorithm=mod.SPLS, lambdaY=
                      lambdaY.to.use, lambdaX=lambdaX.to.use,
                      eps=1e-5, ax.tickvec.X=rep(1,ncol(X.new)),
                      ax.tickvec.Y=rep(5,ncol(Y.new)))
```

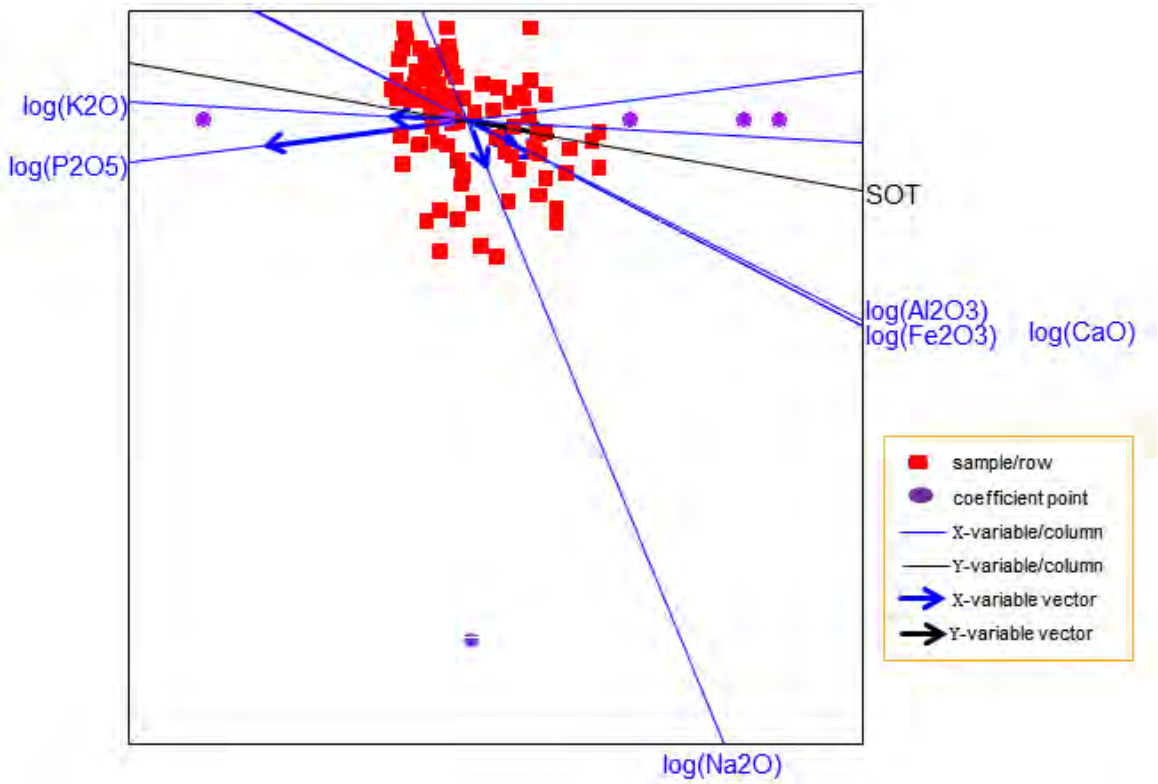

Figure 8.31 The PLS biplot for a SPLS of the ash data, with no sample, coefficients points and tick markers labels, for $\lambda_X = 10$ and $\lambda_Y = 0$.

Moreover, looking at the ash data, with the small number of X- and Y-variables ($P = 8$ and $M = 1$ respectively), performing variables selection on these variables is unnecessary. Thus, a PLS analysis can be performed on this data, rather than a SPLS analysis.

Furthermore, the PLS biplot of a SPLS-GLM of a data set can be obtained using

```
SPLS.GLM.biplot(X, y, algorithm, eps=1e-3, lambdaY, lambdaX,
                ax.tickvec.X, ax.tickvec.y, ax.tickvec.b)
```

where `algorithm` is any of the SPLS-GLM algorithms, "`SPLS.binomial.GLM`" for a Binomial $\mathbf{y}$ and "`SPLS.GLM`" for a Poisson $\mathbf{y}$. This function `SPLS.GLM.biplot` gives a PLS biplot display with the labels of the coefficient points and tick markers inclusive. An application of this function can be seen in Figure 7.7, see Chapter 7. Since the data is Poisson-distributed, `algorithm=SPLS.GLM` is used. The following is used in obtaining the display in Figure 7.7. The accompanying explanations of the resulting display are given in Subsection 7.7.2.

```
library(robustbase)
possum.mat
y = as.matrix(possum.mat[,1], nc=1)
dimnames(y) = list(paste("S", 1:nrow(possum.mat), seq=""),
                   "Diversity")
X = as.matrix(possum.mat[,2:14], nc=13)
dimnames(X) = list(paste("S", 1:nrow(possum.mat), seq=""),
                   colnames(possum.mat[,2:14]))
#choosing a value for the penalty parameters lambdaY and lambdaX
#for this data
main2 = opt.penalty.values(X=scale(X), Y=scale(y), A=2,
                           algorithm=SPLS.GLM, eps=1e-3,
                            from.value.X=0, to.value.X=48,
                             from.value.Y=0, to.value.Y=0,
                              lambdaY.len=1, lambdaX.len=100)
min.RMSEP.value = main2$min.RMSEP.value
lambdaY.to.use = main2$lambdaY.to.use
lambdaX.to.use = main2$lambdaX.to.use
list(lambdaY.to.use=lambdaY.to.use, lambdaX.to.use=
      lambdaX.to.use, min.RMSEP.value=min.RMSEP.value)
#2D plot of the obtained RMSEP values (Figure 7.5)
plot(main2$RMSEP.values[,2], main2$RMSEP.values[,3], type='l',
      xlab="lambdaX", ylab="RMSEP values")
main2B = opt.penalty.values(X=scale(X), Y=scale(y), A=2,
                           algorithm=SPLS.GLM, eps=1e-3,
                            from.value.X=1, to.value.X=4,
                             from.value.Y=0, to.value.Y=0,
                              lambdaY.len=1, lambdaX.len=100)
min.RMSEP.value = main2B$min.RMSEP.value
lambdaY.to.use = main2B$lambdaY.to.use
lambdaX.to.use = main2B$lambdaX.to.use
list(lambdaY.to.use=lambdaY.to.use, lambdaX.to.use=
      lambdaX.to.use, min.RMSEP.value=min.RMSEP.value)
main2B$RMSEP.values
#2D plot of the obtained RMSEP values (Figure 7.6)
plot(main2B$RMSEP.values[,2], main2B$RMSEP.values[,3], xlim=
      c(0,4.5), type='l', xlab="lambdaX", ylab="RMSEP values")
```

```
#which X-variables to use for the SPLS-GLM analysis #which X-
#variables to use for the SPLS-GLM analysis
main3 = SPLS.GLM(scale(X), scale(y), A=2, lambdaY=lambdaY.to.use,
                 lambdaX=lambdaX.to.use, eps=1e-3)
X.to.use = main3$X.select
X.new = as.matrix(X[,names(X.to.use)])
colnames(X.new)
main3$Y.select #note
#SPLS-GLM biplot (Figure 7.7)
SPLS.GLM.biplot(X.new, y, algorithm=SPLS.GLM, eps=1e-3,
                lambdaY=lambdaY.to.use, lambdaX=lambdaX.to.use,
                 ax.tickvec.X=c(10,5,5,5,5,5,5,5,5,5,5,5,5),
                  ax.tickvec.y=8, ax.tickvec.b=12)
#no sample point names (Figure 7.8)
SPLS.GLM.biplot_no.SN(X.new, y, algorithm=SPLS.GLM, eps=1e-3,
                      lambdaY=lambdaY.to.use, lambdaX=
                       lambdaX.to.use, ax.tickvec.X=
                        c(10,5,5,5,5,5,5,5,5,5,5,5,5),
                        ax.tickvec.y=8, ax.tickvec.b=12)
#zoomed-in display of the coefficient points in the PLS-GLM
#biplot (Figure 7.9)
SPLS.GLM.biplot_bvec(X.new, y, algorithm=SPLS.GLM, eps=1e-3,
                     lambdaY=lambdaY.to.use, lambdaX=
                      lambdaX.to.use, ax.tickvec.b=30)
```

To obtain a PLS biplot for a Binomial SPLS-GLM, use

```
SPLS.GLM.biplot(X, y, algorithm=SPLS.binomial.GLM, lambdaY,
                lambdaX, ax.tickvec.X, ax.tickvec.y,
                 ax.tickvec.b)
```

An example of this biplot can be seen in Figure 8.10 of Section 8.4, using

```
data(Colon, package="plsgenomics")
X = as.matrix(cbind(Colon$X))
dimnames(X) = list(1:nrow(X), colnames(X))
y = as.matrix(as.numeric(Colon$Y)-1, ncol=1)
#0=normal and 1=tumor
dimnames(y) = list(1:nrow(y), paste("tissue"))
#SPLS-GLM biplots
#choosing a value for the penalty parameters lambdaY and lambdaX
#for this data
main2 = opt.penalty.values(X=scale(X), Y=scale(y), A=2,
                           algorithm=SPLS.binomial.GLM, eps=
                            1e-3, from.value.X=0, to.value.X=
                             8.5, from.value.Y=0, to.value.Y=0,
                             lambdaY.len=1, lambdaX.len=500)
min.RMSEP.value = main2$min.RMSEP.value
lambdaY.to.use = main2$lambdaY.to.use
lambdaX.to.use = main2$lambdaX.to.use
list(lambdaY.to.use=lambdaY.to.use, lambdaX.to.use=
      lambdaX.to.use, min.RMSEP.value=min.RMSEP.value)
main2$RMSEP.values
#2D plot of the obtained RMSEP values (Figure 8.9)
plot(main2$RMSEP.values[,2], main2$RMSEP.values[,3], type='l',
      xlab="lambdaX", ylab="RMSEP values")
#which X-variables to use for the SPLS-GLM analysis
main3 = SPLS.binomial.GLM(scale(X), scale(y), A=2, lambdaY=
                          lambdaY.to.use, lambdaX=
```

```
                                lambdaX.to.use, eps=1e-3)
X.to.use = main3$X.select
X.new = as.matrix(X[,names(X.to.use)])
#there are still some X-variables with zero coefficient values
#that need to be dropped.
main.rerun = function(X.new)
{
  repeat{
          P.previous = ncol(X.new)
          Rmat = SPLS.binomial.GLM(scale(X.new), scale(y), A=2,
                                   lambdaY=lambdaY.to.use,
                                   lambdaX=lambdaX.to.use,
                                   eps=1e-3)$X.weights.trans
          X.select = which(!Rmat[,1]==0, arr.ind=TRUE)
          X.new = as.matrix(X[,names(X.select)])
          P.now = ncol(X.new)
          if(P.now==P.previous){
            break
            }
            else{
                  Rmat = SPLS.binomial.GLM(scale(X.new), scale(y), A=2,
                                           lambdaY=lambdaY.to.use,
                                           lambdaX=lambdaX.to.use,
                                           eps=1e-3)$X.weights.trans
                  X.select = which(!Rmat[,1]==0, arr.ind=TRUE)
                  X.new = as.matrix(X[,names(X.select)])
             }
       }
  list(X.to.use=colnames(X.new), P.final=P.now, X.final=X.new,
        Rmat=Rmat)
}
main.final = main.rerun(X.new)
main.final$P.final  #final number of selected X-variables
X.new.to.use = main.final$X.final   #final selected X-variables
#Figure 8.10
SPLS.GLM.biplot(X.new.to.use, y, algorithm=SPLS.binomial.GLM,
                eps=1e-3, lambdaY=lambdaY.to.use, lambdaX=
                 lambdaX.to.use, ax.tickvec.X=rep(1,ncol(
                  X.new.to.use)),ax.tickvec.y=5, ax.tickvec.b=1)
#zoomed-in display of the coefficient points in the PLS-GLM
#biplot (Figure 8.11)
SPLS.GLM.biplot_bvec(X.new.to.use, y, algorithm=
                     SPLS.binomial.GLM, eps=1e-3, lambdaY=
                     lambdaY.to.use, lambdaX=lambdaX.to.use,
                      ax.tickvec.b=75)
```

8.5.9 *Summary*

The developed **PLSbiplot1** R package can be used for six main purposes, among others. These main purposes are: (i) for the Principal Component Analysis (PCA) biplots, (ii) the covariance monoplots and biplots, (iii) Partial Least Squares (PLS) biplots, (iv) Partial Least Squares for Generalized Linear Model (PLS-GLM) biplots, (v) Sparse Partial Least Squares (SPLS) biplots, and (vi) Sparse Partial Least Squares for Generalized Linear Model (SPLS-GLM) biplots.

To obtain the PCA biplot of a data set, function

```
PCA.biplot(D, method=mod.PCA, ax.tickvec.D)
```

is used, or

```
PCA.biplot_no.SN(D, method=mod.PCA, ax.tickvec.D)
```

for a display with no sample names.

In this package, the PLS parameters of a data set are found using any of these functions

```
mod.SIMPLS(X, Y, A=A.final) #SIMPLS algorithm
mod.NIPALS(X, Y, A=A.final) #NIPALS algorithm
mod.KernelPLS_R(X, Y, A=A.final) #Kernel algorithm by
                                        #Rännar et al.(1994)
mod.KernelPLS_L(X, Y, A=A.final) #Kernel algorithm by
                                        #Lindgren et al.(1993)
```

where `A.final` is the final number of PLS components to use in each analysis. This number is determined with the aid of the respective RMSEP plot of each analysis:

```
plot(t(mod.SIMPLS(X, Y, A)$RMSEP), type="b",
       xlab="Number of components", ylab="RMSEP values")
                  #SIMPLS algorithm (Algorithm 3.4)
plot(t(mod.NIPALS(X, Y, A)$RMSEP), type="b",
       xlab="Number of components", ylab="RMSEP values")
                  #NIPALS algorithm (Algorithm 3.1)
plot(t(mod.KernelPLS_R(X, Y, A)$RMSEP), type="b",
        xlab="Number of components", ylab="RMSEP values")
                  #Kernel algorithm by Rännar et al.(1994)
                  #(Algorithm 3.2)
plot(t(mod.KernelPLS_L(X, Y, A)$RMSEP), type="b",
        xlab="Number of components", ylab="RMSEP values")
                  #Kernel algorithm by Lindgren et al.(1993)
                  #(Algorithm 3.3)
```

For the VIP analysis of a data use

```
mod.VIP(X, Y, algorithm=mod.SIMPLS, A, cutoff)
```

In addition, for the MMLR and PCR analyses of a data set, functions

```
mod.MMLR(X, Y)
mod.PCR(X, Y, r)
```

are used respectively. Here, `r` is the desired number of PCA components.

For the covariance monoplot of one set of variables and the covariance biplot of the data,

```
cov.monoplot(X)  #for the X-variables
cov.biplot(X, Y)
```

are used respectively.

Furthermore, to obtain various displays of the PLS biplot of a data set, functions

```
PLS.biplot(X, Y, algorithm, ax.tickvec.X, ax.tickvec.Y)
PLS.biplot_no.SN(X, Y, algorithm, ax.tickvec.X, ax.tickvec.Y)
      #with no sample names
```

```
PLS.biplot_no_labels(X, Y, algorithm, ax.tickvec.X, ax.tickvec.Y)
      #with no labels for the samples, coefficient points
      #and tick markers
PLS.biplot.area(X, Y, algorithm, ax.tickvec.X, ax.tickvec.Y,
                  base.tri, bi.value)
      #with area biplot idea
```

are used. In these functions, `algorithm` is any of the PLS algorithms "`mod.NIPALS`", "`mod.KernelPLS_R`", "`mod.KernelPLS_L`" and "`mod.SIMPLS`".

Likewise, for the PLS-GLM, the five different displays of the PLS biplot are found by using functions

```
PLS.GLM.biplot(X, y, algorithm, ax.tickvec.X, ax.tickvec.y,
                 ax.tickvec.b)
      #(Algorithm 6.2)
PLS.GLM.biplot_no.SN(X, y, algorithm, ax.tickvec.X,
                       ax.tickvec.y, ax.tickvec.b)
      #with no labels for the samples
PLS.GLM.biplot_bvec(X, y, algorithm, ax.tickvec.b)
      #Zoomed-in display of the coefficient points
PLS.GLM.biplot_SIMPLS(X, y, algorithm=PLS.GLM_SIMPLS,
                        ax.tickvec.X, ax.tickvec.y,
                         ax.tickvec.b)
      #fitted using the SIMPLS algorithm (Algorithm 6.3)
PLS.GLM.biplot_SIMPLS_no.SN(X, y, algorithm, ax.tickvec.X,
                                ax.tickvec.y, ax.tickvec.b)
      #with no labels for the samples
```

Here, `algorithm` is any of the PLS-GLM algorithms "`PLS.GLM`" and "`PLS.binomial.GLM`".

For the SPLS and SPLS-GLM, various displays of the PLS biplot can be obtained using functions

```
SPLS.biplot(X, Y, algorithm=mod.SPLS, lambdaY, lambdaX,
              ax.tickvec.X, ax.tickvec.Y)
      #(Algorithm 7.1)
SPLS.biplot_no_labels(X, Y, algorithm=mod.SPLS, lambdaY, lambdaX,
                        ax.tickvec.X, ax.tickvec.Y)
      #with no labels for the samples, coefficient points
      #and tick markers
SPLS.biplot_Bmat(X, Y, algorithm=mod.SPLS, lambdaY, lambdaX,
                   ax.tickvec.B)
      #Zoomed-in display of the coefficient points
SPLS.GLM.biplot(X, y, algorithm, lambdaY, lambdaX, ax.tickvec.X,
                   ax.tickvec.y, ax.tickvec.b)
      #PLS biplot for the SPLS-GLM
      #(Algorithm 7.2)
SPLS.GLM.biplot_no.SN(X, y, algorithm, lambdaY, lambdaX,
                        ax.tickvec.X, ax.tickvec.y, ax.tickvec.b)
      #PLS biplot for the SPLS-GLM, with no sample names
SPLS.GLM.biplot_bvec(X, y, algorithm, lambdaY, lambdaX,
                        ax.tickvec.b)
      #Zoomed-in display of the coefficient points
```

In the `SPLS.GLM.biplot`, `SPLS.GLM.biplot_no.SN` and `SPLS.GLM.biplot_bvec` functions, `algorithm` is any of the SPLS-GLM algorithms "`SPLS.GLM`" and "`SPLS.binomial.GLM`".

Moreover, the **PLSbiplot1** package is compatible with both the 32 and 64 bits of the RGui of the 3.1.0 version of R (R Core Team, 2014). Thus, it is suggested that the R version 3.1.0 be used.

CHAPTER 9

CONCLUSIONS AND FUTURE RESEARCH

9.1 Introduction

At the core of multivariate statistics is the investigation of relationships between different sets of variables. Biplots like the CA biplot, PCA biplot and CVA biplot already existed for the inter-variable relationships revelation. The PLS biplot provides for this revelation as well as the casual relationships between two sets of variables (predictors and responses) in terms of the matrix of regression coefficients. In addition, the PLS biplot provides a single graphical representation for displaying results from the PLSR analysis of a data set. In Chapter 5, the biplot theory was applied in the PLS context, to form the PLS biplot. Further extensions were discussed in Chapter 6 (for the GLM framework) and Chapter 7 (for the SPLS and SPLS-GLM). The PLS biplot software (**PLSbiplot1**) was developed for executing these applications. The **PLSbiplot1** package is utilized for extensive applications of the PLS biplot to different data sets, namely, the olive oil data from Mevik & Wehrens (2007), the SOVR data from Umetrics MKS (2013), the Pima.tr data from Smith *et al*. (1988), the cereal data from Varmuza & Filzmoser (2009), the possum diversity data from Lindenmayer *et al*. (1991), the bio-env data from Greenacre (2010) and the colon data from Alon *et al.* (1999). A brief step by step illustration of the use of this package is given in Section 8.5, using the ash and glass data from Varmuza & Filzmoser (2009), the cocktail data from Husson *et al*. (2013), the nutrimouse data by Martin *et al*. (2007) and the spider data from Van der Aart & Smeenk-Enserink (1975).

The main aim of this dissertation was to construct a PLS biplot to display multivariate data graphically. This was accomplished and tested using both small and large data sets. In the PLS biplot, besides simultaneously representing both the samples and the predictor and response variables on the same plot, it was found that the matrix of PLSR coefficients could be added to the plot. For this reason, two different PLS biplot styles were discussed in this dissertation to make it easier to represent these coefficients. The first style involved fitting a second set of markers (in purple) on the prediction biplot axes representing the response variables, and thereafter obtaining the coefficient values directly from the orthogonal projections onto these biplot axes. *The coefficient values are read off using the purple markers on these axes*. The second style used twice the areas of triangles spanned at the origin and the PLS vectors $\mathbf{r}_i$ and $\mathbf{q}_j$ to approximate the $(i,j)^{th}$ element of the coefficients matrix. It is not intrinsic to estimate the

exact area of a triangle visually, but it can be used as an exploratory tool, as larger and smaller coefficients can be easily distinguished. By means of this approach, the exact coefficient values can be obtained by printing out the matrix $\widehat{\mathbf{B}}_{\mathrm{PLSR}}$. Furthermore, the PLS biplot was implemented within the framework of (univariate) PLS-GLMs. This was tested using a small Binomial-distributed dataset (Pima.tr data) and three Poisson-distributed data sets (bio-env, possum diversity and spider data). In addition, the PLS biplot was employed in the SPLS and (univariate) SPLS-GLM frameworks and the resulting outcomes were tested using the possum diversity data from Lindenmayer *et al*. (1991), the cereal data from Varmuza & Filzmoser (2009) and the colon data from Alon *et al.* (1999).

9.2 Conclusions

The discussed methods and developed biplots have only been tested on nine small data sets (8-, 9-, 11-, 13-, 10-, 14-, 17- and 18-dimensions) and three large data sets (141-, 151- and 2001-dimensions). After applying the developed PLS biplot to these data sets, variables and inter-variable relationships were revealed. Considering PLS as an approximation technique, it was found that the PLS biplot approximates the data sets quite well, using $\mathrm{A} = 2$ components.

Furthermore, comparing the PLS biplot with the PCA biplot, it was found that both biplots approximate a large-dimensional data matrix using only a few components, to be exact, 2 components. However, the method of approximation differs across these biplots, hence, the different biplot displays. The PCA biplot treats the two sets of variables as one set, while the PLS biplot treats them as two separate sets of variables. Also, when comparing the PLS biplot with the covariance biplot, different graphical displays were observed. This was due to the approximation method used in approximating the covariance matrix in each technique.

In addition, comparing the PLS biplots obtained under the standard PLS, PLS-GLMs, SPLS and SPLS-GLMs frameworks, different graphical displays were observed. This was attributed to the different PLS computation under each framework.

Seeing that the PLS biplot approximates a large data matrix and that it helps to reveal variables and inter-variable relationships, it can be regarded as another useful graphical tool for displaying large data sets, alongside the PCA biplot. The PLS biplot can also be used as a graphical tool for displaying the approximated covariances between two sets of variables.

For extremely large-dimensional data sets, the SPLS, SPLS-GLM and their resulting biplots are recommended, since the number of variables can be reduced to a minimum. However, with the

main aim of this dissertation being to construct a biplot for large-dimensional data in the form of the PLS biplot, after applying the constructed biplot to several large-dimensional data sets, it was found that the PLS biplot can handle a fairly large data, but not extremely large data sets. Since no biplot such as the PLS biplot currently exists, this dissertation serves as a pioneer work on which others can build upon. The following are proposed, and hopefully, can result in a PLS biplot that can handle extremely large data sets.

9.3 Recommendations for future work

This work initiates another biplot into the biplot family, namely, the PLS biplot. However, some areas of this work could be further expanded. The following are possible areas.

(i) *Data sets*:

The PLS biplot developed in this dissertation was applied to eleven data sets, in sensory analysis, mineral sorting production, chemometrics, gene expression, ecology and biological sciences. It should also be applicable to wider scope of data, e.g., astronomy data and geographical data, whose foremost aim is to investigate the associations between variables and/or samples.

(ii) *Interactive PLS biplot*:

Sample names can be excluded in the PLS biplot, especially, when large samples are involved and the emphasis is more on analysing the relationships between the variables. An interactive PLS biplot can provide a more modest tool, allowing the user to click on a sample point to reveal its name and other information associated with it.

(iii) *Other GLM frameworks*:

Besides the Binomial and Poisson frameworks, PLS can be implemented into other members of the exponential family frameworks, such as the Negative Binomial, Logistic and Multinomial frameworks. This implementation can be explored.

(iv) *Multivariate GLMs*:

With the PLS biplot proposed as a graphical tool for univariate PLS-GLMs and SPLS-GLMs, it can be proposed for a multivariate PLS-GLM and SPLS-GLM. However, more work is still needed to develop the general framework for multivariate GLMs and PLS-GLMs, as well as SPLS-GLMs.

(v) *PLS algorithms*:

Since there is no general rule as to which PLS algorithm is the best, one can use the time factor, i.e., the computational time, to choose which algorithm to use for the data. This rules out the NIPALS algorithm, especially, for large data sets. Nonetheless, there is still other PLS algorithms, such as the kernel and SIMPLS algorithms, to choose from. Alternatively, one can compute the PLS analysis of the data using more than one algorithms, and afterwards, choose the algorithm that gives the minimum MSE or RMSEP value for the response variables of the data. However, this method mostly depends on the size of the data set.

(vi) *Number of PLS components*:

In this dissertation, the number of components to use in the PLS analysis is suggested based on the optimistically biased error rate, in the form of the RMSEP value. Several authors, such as Chong & Jun (2005), Lê Cao *et al*. (2008), Mevik & Wehrens (2007), Wakeling & Morris (1993) and Yoshida *et al*. (2013), suggested different guidelines. Nevertheless, the exact choice of the number of PLS component to use in a PLS analysis still remains an open question.

(vii) *Identifying irrelevant variables*:

The constructed PLS biplot can be applied to fairly large-dimensional data sets, after variables selection technique, such as the VIP technique, has been applied on the data, prior to the PLS biplot analysis. In the VIP technique, one has to apply discretion when it comes to choosing the cut-off value for selecting the relevant X-variables. In Section 3.9 of this dissertation, any X-variables having VIP $\geq$ 0.8 are selected for analysis. With this rule, one cannot help but wonder what about those variables with a VIP value between 0.75 and 0.79 inclusive? Thus, the question remains, how close to 0.8 is close? Chong & Jun (2005) suggests choosing variables having VIP value close to 1.0. Then again, the question still remains, how close to 1.0 is close?

(viii) *PLS/PLS-GLM versus SPLS/SPLS-GLM*:

For the PLS and SPLS, it is assumed that a linear form of relationship exist between the X- and Y-variables, whereas for the PLS-GLM/SPLS-GLM, it is assumed that the relationship follows some non-linear function. However, for the PLS/PLS-GLM analysis, variables selection is only performed on the X-variables,

while for the SPLS/SPLS-GLM analysis, variables selection is performed on both the X- and Y-variables. The latter comes in handy when both the number of X- and Y-variables in the data set are (very) large. Also, the variables selection process in SPLS/SPLS-GLM is based on the penalty parameters λ_X and λ_Y. Though if $\lambda_X = 0$ and $\lambda_Y = 0$, i.e., there is no need to perform variables selection on the X- and Y-variables, then PLS (or PLS-GLM) is more appropriate for the data, than SPLS (or SPLS-GLM). When $\lambda_X \neq 0$ and $\lambda_Y = 0$ (i.e., variables selection on the X-variables only), how does one choose between the PLS (or PLS-GLM) analysis and the SPLS (or SPLS-GLM) analysis, especially, for fairly large data sets? Hence, a rule to determine when to use PLS or SPLS for fairly large data sets can be looked into, especially, if variables selection is only needed on the X-variables. This also applies to their respective GLM framework.

(ix) *Mixed models*:

Since the relationship between the X- and Y-variables of a data set can also follow a mixture of both linear and non-linear functions, a future expansion of the PLS biplot and its software (**PLSbiplot1**) to such situation can be explored.

(x) *Penalization functions*:

Although applying the soft-thresholding LASSO penalty function (Sections 7.3 and 7.4) in the SPLS and SPLS-GLM algorithms did not reduce the very large number of X-variables substantially, as noted in Sections 7.7 and 8.4 for the cereal and colon data respectively, one can consider implementing other penalization functions, such as the Best Subset (hard-thresholding), Ridge (hard-thresholding) and Elastic Net (mixture of soft- and hard-thresholding) penalization functions (Hastie *et al*., 2009; Zou & Hastie, 2005), in the SPLS and SPLS-GLM algorithms.

(xi) *3D PLS biplot*:

Since the PLS biplot was constructed using $A = 2$ components, a future expansion of the PLS biplot software (**PLSbiplot1**) to $A = 3$ components can be explored.

APPENDIX A

```
#All functions used here are stored in the "PLSbiplot1" R package and
#this package (PLSbiplot1) needs to be installed in R, before running any of
#the code below.

#Installation of the PLSbiplot1 package
#First download the PLSbiplot1_0.1.tar.gz file from the dropbox link at
# https://www.dropbox.com/sh/wr66u07t1vjm9da/AACg_E4h8MvgOHuCXk69yDIya
#and install into R.
#This package is also available on CRAN http://cran.r-project.org/

#Loading the PLSbiplot1 package
library(PLSbiplot1)  #or  require(PLSbiplot1)

#The following packages are needed for the data sets used in this Appendix
#"chemometrics", "MASS", "mixOmics", "mvabund", "pls", "plsgenomics",
#"rgl", "robustbase" and "SensoMineR"


#Example sections of Chapter 2
  #Section 2.8
  #olive oil data
  data(oliveoil, package="pls")
  Dmat = as.matrix(oliveoil)
  dimnames(Dmat) = list(paste(c("G1","G2","G3","G4","G5","I1","I2","I3","I4",
                          "I5","S1","S2","S3","S4","S5","S6")),
                          paste(c("Acidity","Peroxide","K232","K270","DK",
                          "Yellow","Green","Brown","Glossy","Transp","Syrup")))
  PCA.biplot(D=Dmat, method=mod.PCA, ax.tickvec.D=c(8,5,5,7,6,4,5,5,8,7,7))

  #Section 2.12
  main = PCA.biplot(D=Dmat, method=mod.PCA, ax.tickvec.D=c(8,5,5,7,6,4,5,5,8,
                      7,7))
  main$overall.quality
  main$axis.pred
  options(digits=0)
  main$sample.pred


#Example sections of Chapter 3
  #Section 3.8
  #olive oil data already specified in "Example sections of Chapter 2"
  X = as.matrix(oliveoil$chemical, ncol=5)
  dimnames(X) = list(paste(c("G1","G2","G3","G4","G5","I1","I2","I3","I4",
                       "I5","S1","S2","S3","S4","S5","S6")),paste(c("Acidity",
                       "Peroxide","K232","K270","DK")))
  Y = as.matrix(oliveoil$sensory, ncol=6)
  dimnames(Y) = list(paste(c("G1","G2","G3","G4","G5","I1","I2","I3","I4",
                       "I5","S1","S2","S3","S4","S5","S6")),paste(c("Yellow",
                       "Green","Brown","Glossy","Transp","Syrup")))
  #final number of PLS components
  RMSEP = mod.SIMPLS(X, Y, A=min(ncol(X),ncol(Y)))$RMSEP #SIMPLS algorithm
  plot(t(RMSEP), type = "b", xlab="Number of components",
        ylab="RMSEP  values")
  A.final = 2 #from the RMSEP plot
  #PLS matrices R, P, T, Q, and Y.hat from SIMPLS algorithm
  options(digits=3)
  main = mod.SIMPLS(X, Y, A=A.final)
```

```
  main$X.weights.trans #matrix_R
  main$X.loadings #matrix_P
  main$X.scores #matrix_T
  main$Y.loadings #matrix_Q
  main$Y.hat[,,A.final] #Y.hat.PLSR

  #Sub-section 3.9.1
  #which PLSR coefficients are influential?
  Mag.Bmat.plot(X, Y, algorithm=mod.SIMPLS, A=A.final)

  #Sub-section 3.10.4
  options(digits=3)
  mod.MMLR(scale(X),scale(Y))$B.mmlr  ##B.hat.MMLR
  mod.PCR(scale(X),scale(Y),r=2)$B.pcr[,,r=2]  #B.hat.PCR


#Example section of Chapter 4
  #Section 4.4
  #X and Y already specified in "Example sections of Chapter 3"
  #covariance biplot
  cov.biplot(X,Y)
  #covariance monoplot
  #Y-variables only
  cov.monoplot(Y)


#Example sections of Chapter 5
  #Section 5.4
  #X and Y already specified in "Example sections of Chapter 3"
  #SIMPLS biplot
  PLS.biplot(X, Y, algorithm=mod.SIMPLS, ax.tickvec.X=c(8,5,5,5,5),
             ax.tickvec.Y=c(5,8,5,6,9,8))
  #Kernel PLS biplot
  PLS.biplot(X, Y, algorithm=mod.KernelPLS_R, ax.tickvec.X=c(3,3,4,5,2),
             ax.tickvec.Y=c(3,3,5,6,7,6))

  #Sub-section 5.5.1
  #SIMPLS biplots with 1 triangle
  PLS.biplot.area(X, Y, algorithm=mod.SIMPLS, ax.tickvec.X=c(8,5,5,5,5),
                  ax.tickvec.Y=c(5,10,5,6,7,10), base.tri=3, bi.value=4)
  #SIMPLS biplots with 4 triangles
  PLS.biplot.area(X, Y, algorithm=mod.SIMPLS, ax.tickvec.X=c(8,5,5,5,5),
                  ax.tickvec.Y=c(5,10,5,6,7,10), base.tri=2,
                  bi.value=c(1,2,3,4,5))


#Example sections of Chapter 6
  #Section 6.5
  #possum diversity data
  library(robustbase)
  possum.mat
  y = as.matrix(possum.mat[,1], nc=1)
  dimnames(y) = list(paste("S", 1:nrow(possum.mat), seq=""), "Diversity")
  X = as.matrix(possum.mat[,2:14], nc=13)
  dimnames(X) = list(paste("S", 1:nrow(possum.mat), seq=""),
                     colnames(possum.mat[,2:14]))
  #Poisson-fitted PLS-GLM
  PLS.GLM.biplot(X, y, algorithm=PLS.GLM, ax.tickvec.X=rep(5,ncol(X)),
                 ax.tickvec.y=10, ax.tickvec.b=7)$bvec
  #zoomed-in display of the coefficient points
  PLS.GLM.biplot_bvec(X, y, algorithm=PLS.GLM, ax.tickvec.b=10)
```

```
  #without samples names
  PLS.GLM.biplot_no.SN(X, y, algorithm=PLS.GLM, ax.tickvec.X=rep(5,ncol(X)),
                       ax.tickvec.y=10, ax.tickvec.b=7)$bvec
  #using the SIMPLS-GLM algorithm
  #with samples names
  PLS.GLM.biplot_SIMPLS(X, y, algorithm=PLS.GLM, ax.tickvec.X=rep(5,ncol(X)),
                        ax.tickvec.y=10, ax.tickvec.b=7)$bvec

  #Section 6.6
  #bio-env data
  bioenv.data = read.table(file.choose(), header=TRUE)
  #choose the file named "bioenv_data" from the "Data Sets" folder.
  rownames(bioenv.data) = paste("S", 1:nrow(bioenv.data))
  Y = as.matrix(bioenv.data[,1:5])
  dimnames(Y) = dimnames(bioenv.data[,1:5])
  sediment = bioenv.data[,9] #sediment variable
  S = (sediment=="S")  #locating sediment S
  Sediment_S = replace(S, S[TRUE],1)
  C = (sediment=="C")  #locating sediment C
  Sediment_C = replace(C, C[TRUE],1)
  G = (sediment=="G")  #locating sediment G
  Sediment_G = replace(G, G[TRUE],1)  #
  #leaving C as the reference category
  X = as.matrix(cbind(bioenv.data[,6:8],Sediment_S,Sediment_G))
  #Poisson-fitted PLS-GLM biplots
  for (i in 1:5)
  {
    which.y.variable = i  #specifying which y-variable to use
    y = as.matrix(Y[,which.y.variable])  #using only 1 Y-variable
    dimnames(y) = list(rownames(Y), colnames(Y)[which.y.variable])
    dev.new()  #opens an empty plotting window
    print(PLS.GLM.biplot(X, y, algorithm=PLS.GLM,
            ax.tickvec.X=rep(3,ncol(X)), ax.tickvec.y=5, ax.tickvec.b=3))
   }
     #for nicer axis markers in PLS biplot displays
      #for y=Y[,1]: use ax.tickvec.X=c(2,3,20,3,1), ax.tickvec.y=5, ax.tickvec.b=4
      #for y=Y[,2]: use ax.tickvec.X=c(2,3,10,3,2), ax.tickvec.y=7, ax.tickvec.b=5
      #for y=Y[,3]: use ax.tickvec.X=c(8,5,15,4,3), ax.tickvec.y=12, ax.tickvec.b=10
      #for y=Y[,4]: use ax.tickvec.X=c(8,5,15,3,3), ax.tickvec.y=7, ax.tickvec.b=10
      #for y=Y[,5]: use ax.tickvec.X=c(5,4,20,4,3), ax.tickvec.y=9, ax.tickvec.b=10


#Example sections of Chapter 7
  #Section 7.7.1
  #cereal data
  data(cereal, package="chemometrics")
  X1 = as.matrix(cbind(cereal$X))
  Y1 = as.matrix(cbind(cereal$Y))
  #choosing a value for the penalty parameters lambdaY and lambdaX for this
  #data
  main2 = opt.penalty.values(X=scale(X1), Y=scale(Y1), A=2,
            algorithm=mod.SPLS, eps=1e-5, from.value.X=0, to.value.X=100,
            from.value.Y=0, to.value.Y=100, lambdaY.len=10, lambdaX.len=100)
  #3D plot of the obtained RMSEP values
  library(rgl)
  x = seq(from=0, to=100, length.out=10)
  y = seq(from=0, to=100, length.out=100)
  z = main2$RMSEP.values[,3]
  persp3d(x,y,z, asp=1, col="red",alpha=0.9, xlab="lambdaY", ylab="lambdaX",
            zlab="RMSEP values")
```

```
main3 = opt.penalty.values(X=scale(X1), Y=scale(Y1), A=2,
          algorithm=mod.SPLS, eps=1e-5, from.value.X=0, to.value.X=500,
          from.value.Y=0, to.value.Y=0, lambdaY.len=1, lambdaX.len=50)
min.RMSEP.value = main3$min.RMSEP.value
lambdaY.to.use = main3$lambdaY.to.use
lambdaX.to.use = main3$lambdaX.to.use
list(lambdaY.to.use=lambdaY.to.use, lambdaX.to.use=lambdaX.to.use,
          min.RMSEP.value=min.RMSEP.value)
main3$RMSEP.values
#2D plot of the obtained RMSEP values
plot(main3$RMSEP.values[,2], main3$RMSEP.values[,3], type='l',
          xlab="lambdaX", ylab="RMSEP values")
#which X- and Y-variables to use for the SPLS analysis
main4 = mod.SPLS(scale(X1), scale(Y1), A=2, lambdaY=lambdaY.to.use,
                      lambdaX=lambdaX.to.use, eps=1e-5)
X.to.use = main4$X.select
Y.to.use = main4$Y.select
X.new = as.matrix(X1[,names(X.to.use)])
ncol(X.new) #108
#no axes labels
SPLS.biplot_no_ax.labels(X.new, Y1, algorithm=mod.SPLS,
          lambdaY=lambdaY.to.use, lambdaX=lambdaX.to.use, eps=1e-5,
          ax.tickvec.X=rep(1,ncol(X.new)), ax.tickvec.Y=rep(8,ncol(Y1)))
#zoomed-in display of the coefficient points
SPLS.biplot_Bmat(X.new, Y1, algorithm=mod.SPLS, lambdaY=lambdaY.to.use,
          lambdaX=lambdaX.to.use, eps=1e-5, ax.tickvec.B=c(8,10,5,5,3,3))

#Section 7.7.2
#possum.mat data
library(robustbase)
possum.mat
y = as.matrix(possum.mat[,1], nc=1)
dimnames(y) = list(paste("S", 1:nrow(possum.mat), seq=""), "Diversity")
X = as.matrix(possum.mat[,2:14], nc=13)
dimnames(X) = list(paste("S", 1:nrow(possum.mat), seq=""),
                      colnames(possum.mat[,2:14]))
#choosing a value for the penalty parameters lambdaY and lambdaX for this
#data
main2 = opt.penalty.values(X=scale(X), Y=scale(y), A=2, algorithm=SPLS.GLM,
          eps=1e-3, from.value.X=0, to.value.X=48, from.value.Y=0,
          to.value.Y=0, lambdaY.len=1, lambdaX.len=100)
min.RMSEP.value = main2$min.RMSEP.value
lambdaY.to.use = main2$lambdaY.to.use
lambdaX.to.use = main2$lambdaX.to.use
list(lambdaY.to.use=lambdaY.to.use, lambdaX.to.use=lambdaX.to.use,
          min.RMSEP.value=min.RMSEP.value)
main2$RMSEP.values
#2D plot of the obtained RMSEP values
plot(main2$RMSEP.values[,2], main2$RMSEP.values[,3], type='l',
          xlab="lambdaX", ylab="RMSEP values")
main2B = opt.penalty.values(X=scale(X), Y=scale(y), A=2,
          algorithm=SPLS.GLM, eps=1e-3, from.value.X=1, to.value.X=4,
          from.value.Y=0, to.value.Y=0, lambdaY.len=1, lambdaX.len=100)
min.RMSEP.value = main2B$min.RMSEP.value
lambdaY.to.use = main2B$lambdaY.to.use
lambdaX.to.use = main2B$lambdaX.to.use
list(lambdaY.to.use=lambdaY.to.use, lambdaX.to.use=lambdaX.to.use,
          min.RMSEP.value=min.RMSEP.value)
main2B$RMSEP.values
#2D plot of the obtained RMSEP values
```

```
  plot(main2B$RMSEP.values[,2], main2B$RMSEP.values[,3], xlim=c(0,4.5),
            type='l', xlab="lambdaX", ylab="RMSEP values")
  #which X-variables to use for the SPLS-GLM analysis
  main3 = SPLS.GLM(scale(X), scale(y), A=2, lambdaY=lambdaY.to.use,
                        lambdaX=lambdaX.to.use, eps=1e-3)
  X.to.use = main3$X.select
  X.new = as.matrix(X[,names(X.to.use)])
  colnames(X.new)
  main3$Y.select #note
  #SPLS-GLM biplot
  SPLS.GLM.biplot(X.new, y, algorithm=SPLS.GLM, eps=1e-3,
            lambdaY=lambdaY.to.use, lambdaX=lambdaX.to.use,
            ax.tickvec.X=c(10,5,5,5,5,5,5,5,5,5,5,5,5), ax.tickvec.y=8,
            ax.tickvec.b=12)
  #no sample point names
  SPLS.GLM.biplot_no.SN(X.new, y, algorithm=SPLS.GLM, eps=1e-3,
            lambdaY=lambdaY.to.use, lambdaX=lambdaX.to.use,
            ax.tickvec.X=c(10,5,5,5,5,5,5,5,5,5,5,5,5), ax.tickvec.y=8,
            ax.tickvec.b=12)
  #zoomed-in display of the coefficient points
  SPLS.GLM.biplot_bvec(X.new, y, algorithm=SPLS.GLM, eps=1e-3,
            lambdaY=lambdaY.to.use, lambdaX=lambdaX.to.use, ax.tickvec.b=30)


#Chapter 8
  #Section 8.2
  #SOVR data
  SOVR.data = read.table(file.choose(), header=TRUE)
  #choose the file named "SOVR_data" from the "Data Sets" folder.
  X = as.matrix(SOVR.data[,1:12])
  dimnames(X) = dimnames(SOVR.data[,1:12])
  Y = as.matrix(SOVR.data[,13:18])
  dimnames(Y) = dimnames(SOVR.data[,13:18])

  #Section 8.2.2
  #which X-variables are important/relevant?
  main2 = mod.VIP(X, Y, algorithm=mod.SIMPLS, A=3, cutoff=0.75)
  main2$VIP.values
  main2$X.impor
  X.new = X[,c(main2$X.impor)]  #important X-variables

  #Section 8.2.3
  #PLS biplot
  main3 = PLS.biplot_no.SN(X.new, Y, algorithm=mod.SIMPLS,
            ax.tickvec.X=rep(1,ncol(X.new)), ax.tickvec.Y=rep(3,ncol(Y)))
  #overall quality
  main3$overall.quality
  #approximated PLSR coefficients values
  main3$Bmat
  #SIMPLS biplots with 9 triangles
  PLS.biplot.area(X.new, Y, algorithm=mod.SIMPLS,
            ax.tickvec.X=rep(1,ncol(X.new)), ax.tickvec.Y=rep(2,ncol(Y)),
            base.tri=4, bi.value=1:9)
  #axis predictivity
  main3$axis.pred
  #which PLS coefficients are influential?
  #using components 1 and 2
  main = mod.SIMPLS(scale(X.new),scale(Y),A=2)
  Qmat = main$Y.loadings
  Rmat = main$X.weights.trans
  Bmat = Rmat %*% t(Qmat)  #(PxM) estimated PLS coefficients matrix
  dimnames(Bmat) = list(colnames(X.new), colnames(Y))
```

```
Mag.Bmat.plot(X.new, Y, algorithm=mod.SIMPLS, A=2)
list(Bmat=Bmat)
#using all three components
main.new = mod.SIMPLS(scale(X.new),scale(Y),A=A.final)
Qmat = main.new$Y.loadings
Rmat = main.new$X.weights.trans
Bmat = Rmat %*% t(Qmat)  #(PxM) estimated PLS coefficients matrix
dimnames(Bmat) = list(colnames(X.new), colnames(Y))
Mag.Bmat.plot(X.new, Y, algorithm=mod.SIMPLS, A=A.final)
list(Bmat=Bmat)

#Section 8.3
#Pima.tr data
data(Pima.tr, package="MASS")
X = as.matrix(cbind(Pima.tr[,1:7]))
dimnames(X) = list(1:nrow(X), colnames(X))
y = as.matrix(as.numeric(Pima.tr$type)-1, ncol=1)
#0=No and 1=Yes
dimnames(y) = list(1:nrow(y), paste("type"))
#PLS-GLM biplots
#with samples names
PLS.GLM.biplot(X, y, algorithm=PLS.binomial.GLM,
          ax.tickvec.X=c(3,3,8,7,8,5,2), ax.tickvec.y=3, ax.tickvec.b=3)
#zoomed-in display of the coefficient points
PLS.GLM.biplot_bvec(X, y, algorithm=PLS.binomial.GLM, ax.tickvec.b=10)
#without samples names
PLS.GLM.biplot_no.SN(X, y, algorithm=PLS.binomial.GLM,
          ax.tickvec.X=c(3,3,8,7,8,5,2), ax.tickvec.y=3, ax.tickvec.b=3)

#Section 8.4
#colon data
data(Colon, package="plsgenomics")
X = as.matrix(cbind(Colon$X))
dimnames(X) = list(1:nrow(X), colnames(X))
y = as.matrix(as.numeric(Colon$Y)-1, ncol=1)
#0=normal and 1=tumor
dimnames(y) = list(1:nrow(y), paste("tissue"))
#Section 8.4.1
#choosing a value for the penalty parameters lambdaY and lambdaX for this
#data
main2 = opt.penalty.values(X=scale(X), Y=scale(y), A=2,
          algorithm=SPLS.binomial.GLM, eps=1e-3, from.value.X=0,
          to.value.X=8.5, from.value.Y=0, to.value.Y=0, lambdaY.len=1,
          lambdaX.len=500)
min.RMSEP.value = main2$min.RMSEP.value
lambdaY.to.use = main2$lambdaY.to.use
lambdaX.to.use = main2$lambdaX.to.use
list(lambdaY.to.use=lambdaY.to.use, lambdaX.to.use=lambdaX.to.use,
           min.RMSEP.value=min.RMSEP.value)
main2$RMSEP.values
#2D plot of the obtained RMSEP values
plot(main2$RMSEP.values[,2], main2$RMSEP.values[,3], type='l',
          xlab="lambdaX", ylab="RMSEP values")
#which X-variables to use for the SPLS-GLM analysis
main3 = SPLS.binomial.GLM(scale(X), scale(y), A=2, lambdaY=lambdaY.to.use,
                          lambdaX=lambdaX.to.use, eps=1e-3)
main3$Y.select #note
X.to.use = main3$X.select
X.new = as.matrix(X[,names(X.to.use)])
colnames(X.new) #857
#there are still some X-variables with zero coefficient values
```

```
#that need to be dropped.
main.rerun = function(X.new)
{
  repeat{
          P.previous = ncol(X.new)
          Rmat = SPLS.binomial.GLM(scale(X.new), scale(y), A=2,
                           lambdaY=lambdaY.to.use, lambdaX=lambdaX.to.use,
                           eps=1e-3)$X.weights.trans
          X.select = which(!Rmat[,1]==0, arr.ind=TRUE)
          X.new = as.matrix(X[,names(X.select)])
          P.now = ncol(X.new)
          if(P.now==P.previous){
            break
           }
          else{
                Rmat = SPLS.binomial.GLM(scale(X.new), scale(y), A=2,
                           lambdaY=lambdaY.to.use, lambdaX=lambdaX.to.use,
                           eps=1e-3)$X.weights.trans
                X.select = which(!Rmat[,1]==0, arr.ind=TRUE)
                X.new = as.matrix(X[,names(X.select)])
           }
      }
  list(X.to.use=colnames(X.new), P.final=P.now, X.final=X.new, Rmat=Rmat)
 }
main.final = main.rerun(X.new)
main.final$P.final  #final number of selected X-variables
X.new.to.use = main.final$X.final  #final selected X-variables
#SPLS-GLM biplot
SPLS.GLM.biplot(X.new.to.use, y, algorithm=SPLS.binomial.GLM, eps=1e-3,
                lambdaY=lambdaY.to.use, lambdaX=lambdaX.to.use,
                ax.tickvec.X=rep(1,ncol(X.new.to.use)), ax.tickvec.y=5,
                ax.tickvec.b=1)
#zoomed-in display of the coefficient points
SPLS.GLM.biplot_bvec(X.new.to.use, y, algorithm=SPLS.binomial.GLM,
                     eps=1e-3, lambdaY=lambdaY.to.use,
                     lambdaX=lambdaX.to.use, ax.tickvec.b=20)


#Section 8.5.3: PCA biplot
#glass data
data(glass, package="chemometrics")
Dmat = matrix(glass,nc=13)
dimnames(Dmat) = list(1:180, paste(c("Na2O", "MgO", "Al2O3", "SiO2",
                     "P2O5", "SO3", "Cl", "K2O", "CaO", "MnO", "Fe2O3",
                     "BaO", "PbO")))
PCA.biplot(D=Dmat, method=mod.PCA, ax.tickvec.D=rep(5,ncol(Dmat)))
PCA.biplot_no.SN(D=Dmat, method=mod.PCA, ax.tickvec.D=rep(5,ncol(Dmat)))

#Section 8.5.4: PLSR
#nutrimouse data
data(nutrimouse, package="mixOmics")
X1 = as.matrix(nutrimouse$lipid, ncol=21)
Y1 = as.matrix(nutrimouse$gene, ncol=120)
main = mod.SIMPLS(X=X1, Y=Y1, A=17) #using the SIMPLS algorithm
main
#RMSEP
RMSEP = main$RMSEP
plot(t(RMSEP), type = "b", xlab="Number of components",
          ylab="RMSEP  values")
A.final = 9 #from the RMSEP plot
```

```
#Final PLSR
mod.SIMPLS(X=X1, Y=Y1, A=A.final)
#VIP
main2 = mod.VIP(X=X1, Y=Y1, algorithm=mod.SIMPLS, A=A.final, cutoff=0.8)
main2
#which X-variables are important/relevant?
main2$X.impor
X.new = X1[,c(main2$X.impor)]  #important X-variables
ncol(X.new)
colnames(X.new)
main2B = mod.VIP(X=X1, Y=Y1, algorithm=mod.SIMPLS, A=A.final, cutoff=0.75)
main2B$X.impor
colnames(X1[,c(main2B$X.impor)])
#influential coefficients
Mag.Bmat.plot(X=X.new, Y1, algorithm=mod.SIMPLS, A=A.final)
#alternatively
X.scal = scale(X.new, center=TRUE, scale=TRUE)
Y.scal = scale(Y1, center=TRUE, scale=TRUE)
main3 = mod.SIMPLS(X.scal, Y.scal, A.final)
#PLSR coefficients matrix
Bmat = main3$X.weights.trans %*% t(main3$Y.loadings)
dimnames(Bmat) = list(colnames(X.new), colnames(Y1))
Abs.Bmat = abs(Bmat) #absolute values of the coefficients
rowMeans(Abs.Bmat)

#Section 8.5.5: Covariance monoplot and biplot
data(cocktail, package="SensoMineR")
X3 = as.matrix(compo.cocktail, nc=4)
Y3 = as.matrix(senso.cocktail, nc=13)
cov.monoplot(Y3)
cov.biplot(X3,Y3)

#Section 8.5.6: PLS biplot
PLS.biplot(X=X3, Y3, algorithm=mod.SIMPLS, ax.tickvec.X=rep(2,ncol(X3)),
                ax.tickvec.Y=rep(3,ncol(Y3)))
#No sample names
PLS.biplot_no.SN(X=X3, Y3, algorithm=mod.SIMPLS,
                ax.tickvec.X=rep(2,ncol(X3)), ax.tickvec.Y=rep(3,ncol(Y3)))
#No sample and tick markers names
PLS.biplot_no_labels(X=X3, Y3, algorithm=mod.SIMPLS,
                ax.tickvec.X=rep(2,ncol(X3)), ax.tickvec.Y=rep(3,ncol(Y3)))
#With area biplot idea
PLS.biplot.area(X=X3, Y3, algorithm=mod.SIMPLS,
                ax.tickvec.X=rep(2,ncol(X3)), ax.tickvec.Y=rep(3,ncol(Y3)),
                base.tri=6, bi.value=c(1:ncol(X3)))

#Section 8.5.7: PLS biplot for GLM
#spider data
data(spider, package="mvabund")
X2 = as.matrix(cbind(spider$x))
rownames(X2) = paste( c("S1", "S2", "S3", "S4", "S5", "S6", "S7", "S8",
                        "S9", "S10", "S11", "S12", "S13", "S14", "S15",
                        "S16", "S17", "S18", "S19", "S20", "S21", "S22",
                        "S23", "S24", "S25", "S26", "S27", "S28") )
Y2 = as.matrix(cbind(spider$abund))
rownames(Y2) = rownames(X2)
which.y.variable = 11  #specifying which y-variable to use
y = as.matrix(Y2[,which.y.variable])  #using only 1 Y-variable
dimnames(y) = list(rownames(Y2), colnames(Y2)[which.y.variable])
PLS.GLM.biplot(X=X2, y, algorithm=PLS.GLM, ax.tickvec.X=rep(2, ncol(X2)),
                ax.tickvec.y=3, ax.tickvec.b=3)
```

```
#zoomed-in display of the coefficient points
PLS.GLM.biplot_bvec(X=X2, y, algorithm=PLS.GLM, ax.tickvec.b=15)
#using SIMPLS
PLS.GLM.biplot(X=X2, y, algorithm=PLS.GLM_SIMPLS, ax.tickvec.X=rep(2,
                 ncol(X2)), ax.tickvec.y=3, ax.tickvec.b=3)

#Pima.tr data
data(Pima.tr, package="MASS")
X = as.matrix(cbind(Pima.tr[,1:7]))
dimnames(X) = list(1:nrow(X), colnames(X))
y = as.matrix(as.numeric(Pima.tr$type)-1, ncol=1)
#0=No and 1=Yes
dimnames(y) = list(1:nrow(y), paste("type"))
#Figure 8.6
PLS.GLM.biplot(X, y, algorithm=PLS.binomial.GLM,
          ax.tickvec.X=c(3,3,8,7,8,5,2), ax.tickvec.y=3, ax.tickvec.b=3)
#Figure 8.7
PLS.GLM.biplot_no.SN(X, y, algorithm=PLS.binomial.GLM,
          ax.tickvec.X=c(3,3,8,7,8,5,2), ax.tickvec.y=3, ax.tickvec.b=3)
#A zoomed-in display of the coefficient points #(Figure 8.8)
PLS.GLM.biplot_bvec(X, y, algorithm=PLS.binomial.GLM,ax.tickvec.b=10)
#using the SIMPLS-GLM algorithm
PLS.GLM.biplot_SIMPLS(X, y, algorithm=PLS.binomial.GLM,
          ax.tickvec.X=c(3,5,8,4,3,7,2), ax.tickvec.y=2, ax.tickvec.b=3)
#no sample names
PLS.GLM.biplot_SIMPLS_no.SN(X, y, algorithm=PLS.binomial.GLM,
          ax.tickvec.X=c(3,5,8,4,3,7,2), ax.tickvec.y=2, ax.tickvec.b=3)

#Section 8.5.8: Biplot for SPLS
#ash data
data(ash, package="chemometrics")
X1 = as.matrix(ash[,10:17], nc=8)
Y1 = as.matrix(ash$SOT)
colnames(Y1) = paste("SOT")
#choosing a value for the penalty parameters lambdaY and lambdaX for this
#data
main2 = opt.penalty.values(X=scale(X1), Y=scale(Y1), A=2,
          algorithm=mod.SPLS, eps=1e-5, from.value.X=0, to.value.X=500,
          from.value.Y=0, to.value.Y=0, lambdaY.len=1, lambdaX.len=100)
min.RMSEP.value = main2$min.RMSEP.value
lambdaY.to.use = main2$lambdaY.to.use
lambdaX.to.use = main2$lambdaX.to.use
list(lambdaY.to.use=lambdaY.to.use, lambdaX.to.use=lambdaX.to.use,
          min.RMSEP.value=min.RMSEP.value)
main2$RMSEP.values
#2D plot of the obtained RMSEP values
plot(main2$RMSEP.values[,2], main2$RMSEP.values[,3], type='l',
          xlab="lambdaX", ylab="RMSEP values")
#which X- and Y-variables to use for the SPLS analysis
main3 = mod.SPLS(X=scale(X1), Y=scale(Y1), A=2, lambdaY=lambdaY.to.use,
                      lambdaX=lambdaX.to.use, eps=1e-5)
X.to.use = main3$X.select
Y.to.use = main3$Y.select #not necessary for the ash data
X.new = as.matrix(X1[,X.to.use])
colnames(X.new)  #P=6
colnames(X1)  #P=8
Y.new = as.matrix(Y1[,Y.to.use])
colnames(Y.new) = colnames(Y1)
colnames(Y.new)
```

```
#SPLS biplot
SPLS.biplot(X.new, Y.new, algorithm=mod.SPLS, lambdaY=lambdaY.to.use,
          lambdaX=lambdaX.to.use, eps=1e-5,
          ax.tickvec.X=rep(1,ncol(X.new)), ax.tickvec.Y=rep(5,ncol(Y.new)))
#no labels
SPLS.biplot_no_labels(X.new, Y.new, algorithm=mod.SPLS,
          lambdaY=lambdaY.to.use, lambdaX=lambdaX.to.use, eps=1e-5,
          ax.tickvec.X=rep(1,ncol(X.new)), ax.tickvec.Y=rep(5,ncol(Y.new)))
#A display of the coefficient points in the PLS biplot
SPLS.biplot_Bmat(X.new, Y.new, algorithm=mod.SPLS, lambdaY=lambdaY.to.use,
          lambdaX=lambdaX.to.use, eps=1e-5, ax.tickvec.B=5)
```

REFERENCES

Abdi, H. (2010). Partial Least Squares Regression and Projection on Latent Structure Regression (PLS Regression). John Wiley & Sons, Inc., *WIREs Computational Statistics*, **2**, 97-106.

Abdi, H. and Williams, L.J. (2010). Principal Component Analysis. John Wiley & Sons, Inc., *WIREs Computational Statistics*, **2**, 433-459.

Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D. and Levine, A.J. (1999). Broad Patterns of Gene Expression revealed by Clustering Analysis of Tumor and Normal Colon Tissues probed by Oligonucleotide Arrays. *Proceedings of the National Academy of Sciences of the United States of America*, **96**(12), 6745-6750.

Barnett, V. (1981). *Interpreting Multivariate Data*. Wiley Series in Probability and Mathematical Statistics. Wiley: New York, USA.

Barr, G.D.I. and Affleck-Graves, J.F. (1987). The Covariance Biplot and Stock Market Data: An Alternative Relative Strength Chart. *South African Journal of Business Management*, **18**, 46-50.

Barr, G.D.I., Underhill, L.G. and Kahn, B.S. (1990). The Covariance Biplot for Graphical Display of Multivariate Time Series Data. *American Journal of Mathematical and Management Sciences*, **10**, 1-15.

Bastien, P., Esposito Vinzi, V., and Tenenhaus, M. (2005). PLS Generalized Linear Regression. *Computational Statistics and Data Analysis*, **48**, 17-46.

Bertrand, F., Meyer, N., Beau-Faller, M., Bayed, K.E., Namer, I.J. and Maumy-Bertrand, M. (2013). Régression Bêta PLS. *Journal de la Société Française de Statistique*, **154**(3), 143-159.

Boulesteix, A. and Strimmer, K. (2005). Partial Least Squares: A Versatile Tool for the Analysis of High-Dimensional Genomic Data. *Seminar for Applied Stochastic*, *Department of Statistics*, *University of Munich*, *Akademiestrasse 1*, *D-80799 Munich*, *Germany*.

Bradu, D. and Gabriel, K.R. (1978). The Biplot as a Diagnostic Tool for Models of Two-Way Tables. *Technometrics*, **20**, 47-68.

Chong, I.G. and Jun, C.H. (2005). Performance of Some Variable Selection Methods when Multicollinearity is Present. *Chemometrics and Intelligent Laboratory Systems*, **78**, 109-112.

Chung, D. and Keles, S. (2010). Sparse Partial Least Squares Classification for High Dimensional Data. *Statistical Applications in Genetics and Molecular Biology*, **9**(1).

Constantine, A.G. and Gower, J.C. (1978). Graphical Representation of Asymmetry Matrices. *Journal of the Royal Statistical Society*, **27**(3), 297-304.

Cooper, M. and De Lacy, H. (1994). Relationships among Analytical Methods used to Study Genotypic Variation and Genotype by Environment Interaction in Plant Breeding Multi-Environment Experiments. *Theoretical and Applied Genetics*, **88**, 561-572.

De Jong, S. (1993). SIMPLS: An Alternative Approach to Partial Least Squares Regression. *Chemometrics and Intelligent Laboratory Systems*, **18**, 251-263.

Ding, B. and Gentleman, R. (2004). Classification Using Generalized Partial Least Squares. *Bioconductor Project Working Papers*, *hosted by The Berkeley Electronic Press*, **5**, 1-29.

Dobson, A. (2002). *An Introduction to Generalized Linear Models*, 2nd Edition. Chapman & Hall/CRC: Boca Raton, FL, USA.

Eckart, C. and Young, G. (1936). The Approximation of One Matrix by Another of Lower Rank. *Psychometrika*, **1**, 211-218.

Fahrmeir, L. and Tutz, G. (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models*. Springer: New York, USA.

Gabriel, K.R. (1971). The Biplot Graphic Display of Matrices with Application to Principal Component Analysis. *Biometrika*, **58**, 453-467.

Gabriel, K.R. (1981). Biplot Display of Multivariate Matrices for Inspection of Data and Diagnosis. In Barnett, V. (ed.), *Interpreting Multivariate Data*, pp. 147-173. Wiley: Chicester, UK.

Gardner, S., Le Roux, N.J. and Aldrich, C. (2005). Process Data Visualisation with Biplots. *Minerals Engineering*, **18**, 955-968.

Gardner-Lubbe, S., Le Roux, N.J. and Gower, J.C. (2008). Measures of Fit in Principal Component and Canonical Variate Analyses. *Journal of Applied Statistics*, **35**(9), 947-965.

Gardner-Lubbe, S., Le Roux, N.J., Maunders, H., Shah, V. and Patwardhan, S. (2009). Biplots for Exploratory Analysis of Gene Expression Data. *Statistical Analysis and Data Mining*, **2**, 135-145.

Gersende, F. and Lambert-Lacroix, S. (2004). Ridge-Partial Least Squares for Generalized Linear models with Binary Response. *Proceedings of Computational Statistics section*, 1-8.

Golub, G.H. and Kahan, W. (1965). Calculating the Singular Values and Pseudo-Inverse of a Matrix. *SIAM Journal of Numerical Analysis*, **2**, 205-224.

Gower, J.C. and Dijksterhuis, G.B. (2004). *Procrustes Problems*. Oxford University Press: New York, USA.

Gower, J.C., Groenen, P.J.F. and Van de Velden, M. (2010). Area Biplots. *Journal of Computational and Graphical Statistics*, **19**(1), 46-61.

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

Gower, J.C. and Harding, S.A. (1988). Nonlinear Biplots. *Biometrika*, **75**, 445-455.

Gower, J.C., Lubbe, S. and Le Roux, N.J. (2011). *Understanding Biplots*. John Wiley & Sons: Chicester, UK.

Greenacre, M.J. (1993). Biplots in Correspondence Analysis. *Journal of Applied Statistics*, **20**, 251-269.

Greenacre, M.J. (2009). The Standard Biplot. *Social Science Research Network*, 1-30.

Greenacre, M.J. (2010). *Biplots in Practice*. Fundación BBVA: Barcelona, Spain.

Greenacre, M.J. (2012). Biplots: The Joy of Singular Value Decomposition. *WIREs Computational Statistics*, 1-8.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd Edition. Springer Series in Statistics.

Helland, I. (1988). On the Structure of Partial Least Squares Regression. *Communications in Statistics*, *Simulation and Computation*, **17**(2), 581-607.

Holland, S.M. (2008). *Principal Components Analysis*. Athens: Department of Geology, University of Georgia.

Höskuldsson, A. (1988). PLS Regression Methods. *Journal of Chemometrics*, **2**, 211-228.

Husson, F., Le S. and Cadoret, M. (2013). *SensoMineR: Sensory Data Analysis with R*. An R package, version 1.17.

Johansson, D., Lindgren, P. and Berglund, A. (2003). A Multivariate Approach Applied to Microarray Data for Identification of Genes with Cell Cycle-Coupled Transcription. *Journal of Bioinformatics*, **19**(4), 467-473.

Jollife, I.T. (1986). *Principal Component Analysis*. Springer-Verlag: New York, USA.

Kempton, R.A. (1984). The Use of Biplots in Interpreting Variety by Environment Interaction. *Journal of Agricultural Science*, **103**, 123-135.

Kohler, U. and Luniak, M. (2005). Data Inspection using Biplots. *The Stata Journal*, **5**(2), 208-223.

Krishnan, A., Williams, L.J., McIntosh, A.R. and Abdi, H. (2011). Partial Least Squares (PLS) Methods for Neuroimaging: A Tutorial and Review. *Neuroimage*, **56**, 455-475.

Lattin, J., Carroll, J.D. and Green, P.E. (2003). *Analyzing Multivariate Data*. Thomson Learning, Inc., Canada.

Lê Cao, K.A., Rossouw, D., Robert-Granié, C. and Besse, P. (2008). A Sparse PLS for Variable Selection when Integrating Omics Data. *Statistical Applications in Genetics and Molecular Biology*, **7**(35).

Le Roux, N.J., Gardner, S. and Olivier, P. (2003). Biplots for Displaying Multidimensional Financial Performance Data Graphically. *South African of Accounting Research*, **17**(1), 41-64.

Lindenmayer, D. B., Cunningham, R. B., Tanton, M. T., Nix, H. A. and Smith, A. P. (1991). The Conservation of Arboreal Marsupials in the Montane Ash Forests of the Central Highlands of Victoria, South-East Australia III: The Habitat Requirements of Leadbeater's Possum Gymnobelideus Leadbeateri and Models of the Diversity and Abundance of Arboreal Marsupials. *Biological Conservation*, **56**, 295-315.

Lindgren, F., Geladi, P. and Wold, S. (1993). The Kernel Algorithm for PLS I. *Journal of Chemometrics*, **7**, 45-59.

Mardia, K.V., Kent, J.T. and Bibby, J.M. (1979). *Multivariate Analysis*. Academic Press: London, UK.

Martens, H. and Naes, T. (1989). *Multivariate Calibration*. John Wiley & Sons: New York, USA.

Martin, P. G. P., Guillou, H., Lasserre, F., Déjean, S., Lan, A., Pascussi, J.-M., San-Cristobal, M., Legrand, P., Besse, P. and Pineau, T. (2007). Novel Aspects of PPAR-Mediated Regulation of Lipid and Xenobiotic Metabolism revealed Through A Multrigenomic Study. *Hepatology*, **54**, 767-777.

Martins, J.P.A., Teófilo, R.F. and Ferreira, M.M.C. (2010). Computational Performance and Cross-Validation Error Precision of Five PLS Algorithms using Designed and Real Data Sets. *Journal of Chemometrics*, **24**, 320-332.

Marx, D. (1996). Iteratively Reweighted Partial Least Squares Estimation for Generalized Linear Regression. *Technometrics*, **38**(4), 374-381.

McCullagh, P. and Nelder, J.A. (1989). *Generalized Linear Models*, 2nd Edition. Chapman and Hall: London, UK.

McIntosh, A.R. and Lobaugh, N.J. (2004). Partial Least Squares Analysis of Neuroimaging Data: Applications and Advances. *Neuroimage*, **23**, 250-263.

Mevik, B.H. and Wehrens, R. (2007). The pls Package: Principal Component and Partial Least Squares Regression in R. *Journal of Statistical Software*, **2**(18), 1-24.

Meyer, N., Maumy-Bertrand, M. and Bertrand, F. (2010). Comparaison de Variantes de Regressions Logistiques PLS et de Régression PLS sur Variables Qualitatives: Application Dux Données d'Allélotypage. *Journal de la Société Française de Statistique*, **151**(2), 1-18.

Nash, M.S. and Lopez, R.D. (2010). Application of Partial Least Square (PLS) Regression to Determine Landscape-Scale Aquatic Resources Vulnerability in the Ozark Mountains. *Section on Statistics and the Environment*: *Joint Statistical Meetings of the American Statistical Association*, 3588-3597.

Osmond, C. (1985). Biplot Models Applied to Cancer Mortality Rates. *Applied Statistics*, **34**, 63-70.

Palermo, G., Piraino, P. and Zucht, H. (2009). Performance of PLS Regression Coefficients in Selecting Variables for Each Response of a Multivariate PLS for Omics-Type Data. Dove Medical Press Ltd. *Advances and Applications in Bioinformatics and Chemistry*, **2**, 57-70.

Park, P.J., Tian, L. and Kohane, I.S. (2002). Linking Gene Expression Data with Patient Survival Times using Partial Least Squares. *Bioinformatics*, **18**(1), 120-127.

Phatak, A. & De Jong, S. (1997). The Geometry of Partial Least Squares. *Journal of Chemometrics*, **11**, 311-338.

R Core Team (2014). R: A Language and Environment for Statistical Computing. *R Foundation for Statistical Computing*, *Vienna*, *Austria*. Available at http://www.R-project.org/.

Rännar, S., Lindgren, F., Geladi, P. and Wold, S. (1994). A PLS Kernel Algorithm for Data Sets with Many Variables and Fewer Objects. Part 1: Theory and Algorithm. *Journal of Chemometrics*, **8**, 111-125.

RStudio (2012). RStudio: Integrated Development for R. Version 0.97.551. Available at http://www.rstudio.org/.

Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C. and Johannes, R. S. (1988). Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus. In *Proceedings of the Symposium on Computer Applications in Medical Care, Greenes, R. A.(ed.): IEEE Computer Society Press*, 261-265.

Tenenhaus, M. (1998). *Lá regression PLS: Theorie et Pratique*. Technip: Paris, France.

Ter Braak, C.J.F. (1983). Principal Components Biplots and Alpha and Beta Diversity. *Ecology*, **64**, 454-462.

Ter Braak, C.J.F. and De Jong, S. (1998). The Objective Function of Partial Least Squares Regression. *Journal of Chemometrics*, **12**, 41-54.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society*, *Series B*, **58**(1), 267-288.

Tobias, D.C. (1997). *An Introduction to Partial Least Squares Regression*. SAS Institute Inc: Cary NC, USA.

Umetrics MKS. (2013). SIMCA-P+. Version 12. Available at http://www.umetrics.com.

Underhill, L.G. (1990). The Coefficient of Variation Biplot. *Journal of Classification*, **7**, 241-256.

Van der Aart, P.J.M. and Smeenk-Enserink, N. (1975). Correlations between Distributions of Hunting Spiders (Lycosidae, Ctenidae) and Environmental Characteristics in a Dune Area. *Netherlands Journal of Zoology*, **25**, 1-45.

Varmuza, K. and Filzmoser P. (2009). *Introduction to Multivariate Statistical Analysis in Chemometrics*. CRC Press: Boca Raton, FL, USA.

Wakeling, I.N. and Morris, J.J. (1993). A Test of Significance for Partial Least Squares Regression. *Journal of Chemometrics*, **7**, 291-304.

Wold, H. (1966). Estimation of Principal Components and Related Models by Iterative Least Squares. In Krishnaiah, P.R. (ed.), *Multivariate Analysis*, pp. 391-420. Academic Press: New York, USA.

Wold, S. (1994). PLS for Multivariate Linear Modelling. In van de Waterbeemd, H. (ed.), *QSAR: Chemometric Methods in Molecular Design. Methods and Principles in Medicinal Chemistry*. Verlag-Chemie: Weinheim, Germany.

Wold, S. (2001). Personal Memories of the Early PLS Development. *Chemometrics and Intelligent Laboratory Systems*, **58**, 83-84.

Wold, S., Ruhe, A., Wold, H. and Dunn, W.J. (1984). The Collinearity Problem in Linear Regression: The Partial Least Squares Approach to Generalized Inverses. *SIAM Journal of Scientific Statistics Computation*, **5**, 735-743.

Wold, S., Johansson, E. and Cocchi, M. (1993). PLS-Partial Least Squares Projections to Latent Structures. In H. Kubinyi (ed.), *3D QSAR in Drug Design, Theory, Methods, and Applications*, pp. 523-550. ESCOM Science Publishers: Leiden, Holland.

Wold, S., Sjöström, M. and Eriksson, L. (2001). PLS-Regression: A Basic Tool of Chemometrics. *Chemometrics and Intelligent Laboratory Systems*, **58**, 109-130.

Yan, W. and Kang, M.S. (2003). *GGE Biplot Analysis*. CRC Press: Boca Raton, FL, USA.

Yan, W. and Rajcan, I. (2002). Biplot Analysis of Test Sites and Trait Relations of Soybeans in Ontario. *Crop Science*, **42**, 11-20.

Yan, W. and Tinker, N.A. (2006). Biplot Analysis of Multi-Environment Trial Data: Principles and Applications. *Journal of Plant Science*, **86**, 623-645.

Yoshida, H., Kawaguchi, A. and Tsuruya, K. (2013). Radial Basis Function-Sparse Partial Least Squares for Application to Brain Imaging Data. *Computational and Mathematical Methods in Medicine*. Hindawi Publishing Corporation.

Zou, H. and Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society*, *Series B*, **67**(2), 301-320.