Regularized Statistical Analysis of Anatomy

Karl Sjöstrand

Kongens Lyngby 2007
IMM-PHD-2007-182

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-PHD: ISSN 0909-3192

Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling (IMM) of the Technical University of Denmark (DTU), in partial fulfillment of the requirements for acquiring a Ph.D. degree in mathematical modelling.

The thesis deals with the application and development of regularized statistical methods to the analysis of anatomical structures, predominantly in the human brain. The thesis consists of a review of methods for regularization in regression, classification and data decomposition, ranging from classic to current. This is followed by a collection of five research papers written during the period 2003–2007, and elsewhere published.

The work has been carried out in collaboration with the Danish Research Centre for Magnetic Resonance (DRCMR) of the Copenhagen University Hospital, Hvidovre, Denmark. Part of the research was conducted at the San Francisco Veterans Affairs Medical Center of the University of California San Francisco (UCSF), USA.

The project was supervised by Associate Professor Rasmus Larsen (IMM) and partly supervised by Dr. Colin Studholme (UCSF). Funding was provided by the Technical University of Denmark, and partly by the ITMAN graduate school, also of the Technical University of Denmark.

Kgs. Lyngby, June 2007

Karl Sjöstrand

Karl Sjöstrand

Acknowledgements

A doctoral thesis has a single author, but represents the collective outcome of time and energy spent by colleagues, family and friends. This section serves to acknowledge those who have contributed significantly to the work presented herein.

Pigmaei gigantum humeris impositi plusquam ipsi gigantes vident. My deepest gratitude is extended towards my principal mentors over the course of this project. These include my supervisor Rasmus Larsen (IMM) who introduced me to the world of modern statistics and made sure my research was going in a fruitful direction. Moreover, I highly appreciate the honest and open-hearted career advice that he has provided. Next, I acknowledge my supervisor during my stint as ”junior specialist” at UCSF, Colin Studholme, who showed me what it means to be a hard-working and thorough researcher. Finally, my role-model during the all-important first half of my Ph.D. project, Mikkel B. Stegmann, is greatly acknowledged. Mikkel is both a great colleague and friend, always suitably astonished over my attempts at action sports, and constantly educating me on the topic of soul and r&b classics. As implied by the saying above, the support of these people has allowed me to raise to a level where parts of the academic landscape have become visible that would otherwise be beyond my intellectual horizon.

The complete list of co-authors of the papers in the list of published papers shows the breadth of the collaborations that I have been blessed with. I thank you for your time and hard work towards getting these papers accepted for publication.

I am grateful to the people at the Danish Research Centre for Magnetic Resonance (DRCMR), who supplied most of the data used in this project, and who

invited me to come and work in an inspiring hospital environment. I regret that I did not find more time to work at and in collaboration with the DRCMR. My closest collaborators have been Egill Rostrup, Charlotte Ryberg and Lars G. Hanson.

The staff at the Center of Imaging of Neurodegenerative Diseases at the VA Medical Center, UCSF, San Francisco, USA are acknowledged for allowing me to take part in their research over a period of six months from March to August 2006. I know that it was a major undertaking to get the papers in order for my visa application, and I appreciate the effort. In particular, I would like to thank my closest UCSF co-workers Colin Studholme, Claudia Rodriguez-Carranza and Valerie Cardenas-Nicolson for excellent collaboration and camaraderie.

I also wish to thank Professor Trevor Hastie and Professor Paul Switzer at the Department of Statistics, Stanford University, USA, for inviting me to visit Stanford and allowing me to present my research.

Work in the Image Analysis and Computer Graphics group at IMM, DTU, has been an utterly pleasant experience, largely due to excellent work mates, both past and present. In particular, I am grateful to Hildur Ólafsdóttir, a first-rate colleague and friend, who agreed to proofread this thesis at a time where she herself was busy finishing her Ph.D. project. I can only hope to be able to return the favor. Thanks also to Søren G. Erbou who encrypted the abstract of this thesis into Danish.

Finally, I extend my gratitude to Karin Sjöstrand, not just for being my wife in general, but specifically for contributing to this thesis through discussions, structural comments and for allotting me the time I needed to finish in time, despite having a busy career of her own.

Abstract

This thesis presents the application and development of regularized methods for the statistical analysis of anatomical structures. Focus is on structure-function relationships in the human brain, such as the connection between early onset of Alzheimer's disease and shape changes of the corpus callosum. One of the comprehensive goals of this type of research is to use non-invasive imaging devices for the detection of diseases which are otherwise difficult to diagnose at an early stage. A more modest but equally interesting goal is to improve the understanding of the brain in relation to body and mind. Statistics represents a quintessential part of such investigations as they are preluded by a clinical hypothesis that must be verified based on observed data.

The massive amounts of image data produced in each examination pose an important and interesting statistical challenge, in that there are many more image features (variables) than subjects (observations), making an infinite number of solutions possible. To arrive at a unique and interesting answer, the analysis must be constrained, or regularized, in a sensible manner. This thesis describes such regularization options, discusses efficient algorithms which make the analysis of large data sets feasible, and gives examples of applications.

Resumé

Denne afhandling beskriver udvikling og anvendelse af regulariserede metoder til statistisk analyse af anatomiske strukturer. Fokus er på strukturer og funktionalitet i den menneskelige hjerne, såsom sammenhængen mellem tidlige tegn på Alzheimers sygdom og ændringer i formen af hjernebjælken (corpus callosum). Ofte er det overordnede mål for denne type forskning, at anvende ikke-invasive billeddannende tekniker til at, på et tidlig stadie, detektere sygdomme der ellers er svære at diagnosticere. Et mere beskedent men lige så interessant mål eri at forbedre forståelsen af hjernen i forhold til krop og sind. Statistik er et essentielt værktøj til sådanne undersøgelser da man har en klinisk hypotese der ønskes verificeret på baggrund af observerede data.

Den omfattende mængde af billeddata der produceres i hver undersøgelse udgør en vigtig og interessant statistisk udfordring, da den medfører mange flere variable end observationer, hvilket igen giver en uendelig mængde af mulige løsninger. For at udlede entydige og interessante svar må løsningen nødvendigvis regulariseres på en passende måde. Denne afhandling beskriver sådanne regulariseringsmuligheder, diskuterer effektive algoritmer som gør analysen af store datamængder mulig, samt giver eksempler på anvendelser af disse.

List of Published Papers

Listed here are peer-reviewed scientific papers and abstracts prepared during the course of the Ph.D. program. Before October 1st 2005, the author's surname was Skoglund, while the author's current surname is Sjöstrand. Publications exist in both names.

Journal papers:

- P.-H. Yeh, S. Gazdzinski, T.C. Durazzo, K. Sjöstrand, D.J. Meyerhoff. Hierarchical Linear Modeling of Longitudinal Brain Structural and Cognitive Changes in Alcohol-dependent Individuals during Sobriety. *Drug and Alcohol Dependence*, 2007. Accepted for publication.
- K. Sjöstrand, E. Rostrup, C. Ryberg, R. Larsen, C. Studholme, H. Baezner, J. Ferro, F. Fazekas, L. Pantoni, D. Inzitari, G. Waldemar. Sparse Decomposition and Modeling of Anatomical Shape Variation. *IEEE Transactions on Medical Imaging*, 2007. Accepted for publication.
- K. Sjöstrand, M.S. Hansen, H.B.W. Larsson, R. Larsen. A Path Algorithm for the Support Vector Domain Description and its Application to Medical Imaging. *Medical Image Analysis*, 2007. Accepted for publication.

Conference papers:

- M.S. Hansen, K. Sjöstrand, H. Ólafsdóttir, H.B.W. Larsson, M.B. Stegmann, R. Larsen. Robust Pseudo-Hierarchical Support Vector Clustering. *Scandinavian Conference on Image Analysis - SCIA*. 2007.

- H. Ólafsdóttir, M.S. Hansen, K. Sjöstrand, T.A. Darvann, N.V. Hermann, E. Oubel, B.K. Ersbøll, R. Larsen, A.P. Larsen, C.A. Perlynn, G.M. Morriss-Kay, S. Kreiborg. Sparse Statistical Deformation Model for the Analysis of Craniofacial Malformations in the Crouzon Mouse. *Scandinavian Conference on Image Analysis - SCIA*. 2007.

- M.S. Hansen, H. Ólafsdóttir, K. Sjöstrand, H.B.W. Larsson, M.B. Stegmann, R. Larsen. Ischemic Segment Detection using the Support Vector Domain Description. *International Symposium on Medical Imaging - SPIE*, 2007.

- K. Sjöstrand, R. Larsen. The entire regularization path for the support vector domain description. *Medical Image Computing and Computer-Assisted Intervention - MICCAI*. 2006.

- K. Sjöstrand, M.B. Stegmann, R. Larsen. Sparse Principal Component Analysis in Medical Shape Modeling. *International Symposium on Medical Imaging - SPIE*. 2006.

- K. Sjöstrand, A. Ericsson, R. Larsen. On the Alignment of Shapes Represented by Fourier Descriptors. *International Symposium on Medical Imaging - SPIE*. 2006.

- M.B. Stegmann, K. Sjöstrand, R. Larsen. Sparse Modeling of Landmark and Texture Variability using the Orthomax Criterion. *International Symposium on Medical Imaging - SPIE*. 2006.

- M.B. Stegmann, K. Skoglund. On Automating and Standardising Corpus Callosum Analysis in Brain MRI. *Proceedings, Svenskt Symposium i Bildanalys - SSBA*. 2005.

- M.B. Stegmann, K. Skoglund, C. Ryberg. Mid-sagittal plane and mid-sagittal surface optimization in brain MRI using a local symmetry measure. *International Symposium on Medical Imaging - SPIE*. 2005.

- R. Larsen, K.B. Hilger, K. Skoglund, S. Darkner, R.R. Paulsen, M.B. Stegmann, B. Lading, H. Thodberg, H. Eiriksson. Some Issues of Biological Shape Modelling with Applications. *13th Scandinavian Conference on Image Analysis - SCIA*. 2003.

Abstracts:

- C. Ryberg, M.B. Stegmann, K. Sjöstrand, E. Rostrup, F. Barkhof, F. Fazekas, G. Waldemar. Corpus Callosum Partitioning Schemes and Their Effect on Callosal Morphometry. *Proceedings, International Society of Magnetic Resonance In Medicine - ISMRM*. 2006.

- K. Sjöstrand, T.E. Lund, K.H. Madsen, R. Larsen. Sparse PCA, a new method for unsupervised analyses of fMRI data. *Proceedings, International Society of Magnetic Resonance In Medicine - ISMRM.* 2006.
- K. Skoglund, M.B. Stegmann, C. Ryberg, H. Ólafsdóttir, E. Rostrup. Estimation and Perturbation of the Mid-Sagittal Plane and its Effects on Corpus Callosum Morphometry. *Proceedings, International Society of Magnetic Resonance In Medicine - ISMRM.* 2005.

Contents

Chapter 1

Introduction

This thesis presents the application and development of regularized statistical methods to the analysis of anatomical structures, predominantly in the human brain. The presentation is divided into two parts.

The first part provides a review of a selection of statistical methods in which regularization is used to make the analysis of large and/or difficult data sets possible. This part consists of three chapters, dealing with regression, classification and data decomposition respectively, of which the latter is confined to a discussion of sparse and regular principal component analysis. In large, each chapter is organized such that each section leads up to the next, introducing increasingly involved methods.

The second part presents a few applications of these methods to the analysis of anatomy. This part of the thesis consists of original research papers published over the course of the Ph.D. project.

1.1 Scope

The range of subjects touched upon in this thesis is broad, ranging from classical (frequentist) statistics to modern statistical methods pertinent to e.g. pattern recognition and machine learning. The area of application, medical image analysis, draws upon results from e.g. shape analysis (morphometry) and image registration and segmentation. Most image data come from magnetic resonance (MR) imaging devices. Handling such data requires basic knowledge of MR physics. Other clinical and cognitive data may for instance arise from standardized test batteries of physical performance and psychiatric investigations of mental health, which overlaps with other areas of statistics such as behavioral science and epidemiology.

Working at the crossroads of these fields of research is both a trying and a rewarding experience. A simple analysis of an MR data set may take years to perfect, taking the breadth of necessary knowledge into account. For the same reason, this thesis does not attempt to cover more than fragments of the topics listed above. The discussion is mainly focused on a small set of regularized statistical methods for regression, classification and data decomposition, most of them developed within the last decade. Some earlier work is also reviewed to provide a sound basis for the rest of the thesis.

The use of these statistical methods in the analysis of anatomical structures is deferred to the contributions in Part II, where short introductions to image acquisition, clinical and cognitive data, shape analysis etc. are given.

1.2 Purpose

The purpose of this thesis, besides being a mandatory part of a Ph.D. degree, is to provide a comprehensive reference to useful statistical methods and to show how these can be applied in the area of medical image analysis, with focus on morphometric analysis of brain anatomy. For some methods, we have attempted to give alternative derivations and interpretations to complement the original publications. Most methods are summarized using pseudo-code, facilitating and encouraging implementation by the reader. Matlab code for most of these methods have been made available and can be found on the home page of the author, `www.imm.dtu.dk/~kas`.

1.3 Definitions

Throughout this thesis the following notation is used, unless otherwise stated.

The number of observations The number of observations is denoted by n.

The number of variables The number of variables are denoted p, unless a change of variables is performed, in which case the number of variables is denoted k.

Scalars Scalars are denoted using lower-case lightface letters such as x or y.

Vectors Vectors are denoted using lower-case boldface letters such as $\mathbf{x}$ and $\mathbf{y}$.

Matrices Matrices are denoted using upper-case boldface letters such as $\mathbf{X}$ and $\mathbf{Y}$.

Random variables Random variables are represented by capital italic lightface letters such as X or Y. Random model coefficients are denoted using Greek letter such as β_i, where the index i usually refers to the coefficient corresponding to the i^{th} variable.

Observed variables Realizations of random variables are denoted using the same letters as those for random variables, but in lower-case boldface formatting for vectors and lightface for scalars. Observed model coefficients are denoted using the Latin letter corresponding to the Greek letter used to denote a random variable. For instance, the vector of observations pertaining to the random coefficient β_i is denoted $\mathbf{b}_i$.

Errors and residuals The error ε is a measure of the difference between an observation and its expected value. The term is therefore a misnomer. A residual r denotes the difference between an observation and its average value in a sample, and is therefore an estimate of ε. In regression analyses, ε denotes the error term when random variables are regarded while the residual vector $\mathbf{r}$ is the estimated ditto.

Expectation The expectation of a random variable or the mean of a vector is denoted $E(X)$ and $E(\mathbf{x}) = n^{-1}\sum_i x_i$ respectively.

Variance The variance of a random variable and the sample variance of a vector is denoted $\text{var}(X) = E\left((X - E(X))^2\right)$ and $\text{var}(\mathbf{x}) = (n-1)^{-1}\sum_i (x_i - E(\mathbf{x}))^2$ respectively.

Standard deviation The standard deviation is denoted $\text{std}(X) = \sqrt{\text{var}(X)}$ or equivalently $\text{std}(\mathbf{x}) = \sqrt{\text{var}(\mathbf{x})}$.

Covariance The covariance between two random variables X and Y is denoted $\text{cov}(X, Y) = E\left((X - E(X))(Y - E(Y))\right)$. For vectors $\mathbf{x}$ and $\mathbf{y}$ of observations, the covariance is estimated by

$$\text{cov}(\mathbf{x}, \mathbf{y}) = \frac{1}{n-1}\sum_i \left((x_i - E(\mathbf{x}))(y_i - E(\mathbf{y}))\right) = \frac{(\mathbf{x} - \bar{\mathbf{x}})^T(\mathbf{y} - \bar{\mathbf{y}})}{n-1}.$$

When cov is applied to a centered matrix of observations, it denotes the variance-covariance matrix $\text{cov}(\mathbf{X}) = (n-1)^{-1}\mathbf{X}^T\mathbf{X}$.

Correlation The correlation coefficient between two variables $\mathbf{x}$ and $\mathbf{y}$ is denoted

$$\mathrm{corr}(\mathbf{x},\mathbf{y}) = \frac{\mathrm{cov}(\mathbf{x},\mathbf{y})}{\mathrm{std}(\mathbf{x})\mathrm{std}(\mathbf{y})} = \frac{\sum_i (x_i - E(\mathbf{x}))(y_i - E(\mathbf{y}))}{\sqrt{\sum_i (x_i - E(\mathbf{x}))^2}\sqrt{\sum_i (x_i - E(\mathbf{x}))^2}}.$$

For a centered and normalized matrix $\mathbf{X}$ the correlation matrix is simply $\mathrm{corr}(\mathbf{X}) = \mathbf{X}^T\mathbf{X}$.

The identity matrix The identity matrix is denoted $\mathbf{I}$, or $\mathbf{I}_k$ indicating its size $(k \times k)$.

The unit and zero vector. A vector of ones is denoted $\mathbf{1}$ or $\mathbf{1}_k$ indicating its size $(k \times 1)$. Similarly for the zero vector $\mathbf{0}$ and $\mathbf{0}_k$.

Hadamard product The Hadamard (element-wise) product of two matrices $\mathbf{A}$ and $\mathbf{B}$ is denoted $\mathbf{A} \circ \mathbf{B}$. The notation $\mathbf{A}^k$ denotes the Hadamard product of k matrices $\mathbf{A}$. Further, $\mathbf{A}^{-k}$ is convenient notation for $(\mathbf{A}^{-1})^k$, the Hadamard product of k matrices $\mathbf{A}^{-1}$, where $\mathbf{A}^{-1}$ represents the standard matrix inverse.

Vector norms A norm is a measure of the size of a vector or matrix. We adopt the vector norm definitions of the MATLAB software. Norms are denoted using the symbol ℓ, where for instance ℓ_2 is called the "two-norm".

$$\begin{array}{rclcl}
\ell_p(\mathbf{a}) & = & \|\mathbf{a}\|_p & = & (\sum_{i=1}^n |a_i|^p)^{\frac{1}{p}} \\
\ell_2(\mathbf{a}) & = & \|\mathbf{a}\| & = & \sqrt{\sum_{i=1}^n a_i^2} \\
\ell_\infty(\mathbf{a}) & = & \|\mathbf{a}\|_\infty & = & \max_i |a_i| \\
\ell_{-\infty}(\mathbf{a}) & = & \|\mathbf{a}\|_{-\infty} & = & \min_i |a_i|
\end{array}$$

The two-norm is of course just a special case of the p-norm but is usually written without subscript.

Matrix norms Matrix norms are similar in notation to vector norms but differ in calculation.

$$\begin{array}{rclcl}
\ell_2(\mathbf{A}) & = & \|\mathbf{A}\| & = & \max_i d_i \\
\ell_1(\mathbf{A}) & = & \|\mathbf{A}\|_1 & = & \max_i \sum_{j=1}^n |a_{ji}| \\
\ell_\infty(\mathbf{A}) & = & \|\mathbf{A}\|_\infty & = & \max_i \sum_{j=1}^p |a_{ij}| \\
\ell_f(\mathbf{A}) & = & \|\mathbf{A}\|_f & = & \sqrt{\sum_{i=1}^p \mathbf{x}_i^T\mathbf{x}_i}
\end{array}$$

Here, d_i is the i^{th} singular value of $\mathbf{A}$. The norm $\ell_1(\mathbf{A})$ is the largest column sum of $\mathbf{A}$ while $\ell_\infty(\mathbf{A})$ is the largest row sum. The norm ℓ_f is called the *Frobenius* norm.

Diagonal matrices The function diag($\mathbf{a}$) turns a length-n vector $\mathbf{a}$ into an $(n \times n)$ diagonal matrix $\mathbf{A}$ which has the elements of $\mathbf{a}$ along its diagonal. Rarely, we also use the notation diag($\mathbf{A}$) which produces a vector $\mathbf{a}$ consisting of the diagonal elements of $\mathbf{A}$, regardless of whether $\mathbf{A}$ is diagonal or not.

1.4 Formulae

$$(\mathbf{ABC})^{-1} = \mathbf{C}^{-1}\mathbf{B}^{-1}\mathbf{A}^{-1} \tag{1.1}$$

$$\text{var}(X) = E\left((X - E(X))^2\right) = E(X^2) - E(X)^2 \tag{1.2}$$

$$\text{cov}(X, Y) = E\left((X - E(X))(Y - E(Y))\right) = E(XY) - E(X)E(Y) \tag{1.3}$$

$$E(a + bX) = a + bE(X) \tag{1.4}$$

$$\text{var}(a + bX) = b^2\text{var}(X) \tag{1.5}$$

$$E(\mathbf{A} + \mathbf{B}X\mathbf{C}) = \mathbf{A} + \mathbf{B}E(X)\mathbf{C} \tag{1.6}$$

$$\text{var}(\mathbf{A} + \mathbf{B}X) = \mathbf{B}\text{var}(X)\mathbf{B}^T \tag{1.7}$$

Part I

Regularized Statistical Methods

Chapter 2

Regression

2.1 Introduction

Use of the word *regression* was first reported in the 14th century, stemming from the Latin word *regressus*, describing the act of going back to a previous place or state. This is a reasonably accurate description of the word in its statistical sense. It is assumed that there is a *true* model describing the relation between sets of variables which has been perturbed by random noise, thus forming the observed data. The term is accurate in the sense that a regression analysis represents an attempt of going back from the observed data to the true model.

The mathematical formulation of a regression model under squared error loss is contained in the *regression equation*,

$$Y = E(Y|X = x) = f(x) + \varepsilon, \tag{2.1}$$

where Y is the random variable we wish to characterize, X is a (possibly multivariate) random variable denoting the input data, f is an arbitrary function of X, and ε is an error term. The regression equation is conditioned on X, that is, the observed input data is considered fixed. The variables X and Y have a range of different names, largely depending on the field of research. As medical image analysis is an interdisciplinary field, there is a variety of terms used in research papers. Some of the most common ones are presented in Table 2.1. The terms *response variable* (Y) and *predictor variable* (x) are used here. We

Y	x
dependent variable	independent variable
response variable	predictor variable
explained variable	explanatory variable
regressand	regressor
endogenous variable	exogenous variable
output variable	input variable
criterion variable	
	covariate
outcome variable	

Table 2.1: Common terms for the output variable Y and the input variable(s) x. In this thesis, we refer to these as *response* and *predictor* variables.

will refer to a clinical or cognitive variable, for instance stemming from tests of muscle strength (clinical) or a rating of depression (cognitive), as an *outcome* variable, regardless of whether it enters the model as a response or a predictor variable. Predictor variables which relate both to other predictors and the response variable and which must be included in the model to obtain reliable results are called *confounding* variables, but may be referred to as *covariates* elsewhere.

This chapter has the following layout. In Section 2.2 we introduce the linear model which is the basis for all methods described in this thesis, and provide a short review of what the model represents and what information it may provide. We also remind the reader that linear modeling does not limit the analysis to linear regression functions. The following sections concern the estimation of the regression coefficients. Section 2.3 describes the standard non-regularized approach to this end from a geometrical viewpoint. Section 2.4 introduces the terms bias and variance and discusses how these interact. We conclude the section with a review of the Gauss-Markov theorem, which serves to introduce and motivate the use of regularization in a statistical method. Section 2.5 provides a brief explanation of the use of univariate, or *pointwise*, regression for high-dimensional problems. Section 2.6 introduces ridge regression and exposes pointwise regression as a strongly regularized form of this technique. In Section 2.7 we introduce two classic variable selection methods. The following sections presents more recent methods that combine regularization and variable selection in a single framework. Section 2.8 discusses least angle regression, a geometrically motivated method. The LASSO, a closely related method is then presented in Section 2.9. Section 2.10 presents the Elastic Net, a combination of ridge regression and the LASSO. The chapter is concluded with a similar method known as the non-negative garrote (Section 2.11).

2.2 The Linear Model

Throughout this thesis, we will assume that the phenomenon under study can be described using a linear model. This means that the relationship between the response and the predictors can be reasonably accurately formulated as

$$\begin{aligned} Y = E(Y|X=x) &= \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \varepsilon = \\ &= \beta_0 + X\beta + \varepsilon, \end{aligned} \tag{2.2}$$

where the fixed predictor variables are denoted x_i, Y is the random response, β are the regression coefficients (β_0 is known as the *intercept*), and the random errors are denoted ε. Most often, we will write this equation using vectors of observations $\mathbf{y}$ and $\mathbf{x}_i$,

$$\begin{aligned} \mathbf{y} &= b_0 + b_1\mathbf{x}_1 + \ldots + b_p\mathbf{x}_p + \mathbf{r} = \\ &= b_0 + \mathbf{X}\mathbf{b} + \mathbf{r}, \end{aligned} \tag{2.3}$$

where $\mathbf{X}$ $(n \times p)$ is called the *data matrix* and the response $\mathbf{y}$ and the residuals $\mathbf{r}$ are $(n \times 1)$ vectors. Performing a regression analysis amounts to the calculation of the regression coefficients $\mathbf{b}$ which preferably are as close as possible to the real (unknown) coefficients β.

Assume, for instance, that we wish to measure the dependence between height of a group of (adult) sons and the height of their fathers[1]. The hypothesis is that short fathers have short sons and vice versa, and we assume that this relationship is approximately linear. Figure 2.1 shows a synthetic collection of height measurements. The green line represents the hypothesis $Y = x$ while the red line represents the fitted regression function. As can be seen, there is strong correlation between the two variables, and a linear model seems appropriate. We conclude that the expected height of a son is closely related to the height of his father. The next section discusses why such conclusions should be drawn with caution.

2.2.1 What can be Inferred?

Assuming a linear model is appropriate, the observed data in Figure 2.1 deviate from the hypothesis $Y = x$ for two reasons. First, uncertainties in the measurements may perturb the variables. When measuring a person's height, this factor is assumed negligible but may be significant in more complex investigations. Second, the variance of the response variable can usually not be

[1] This was in fact one of the first regression analyses in history and was carried out by statistician Karl Pearson

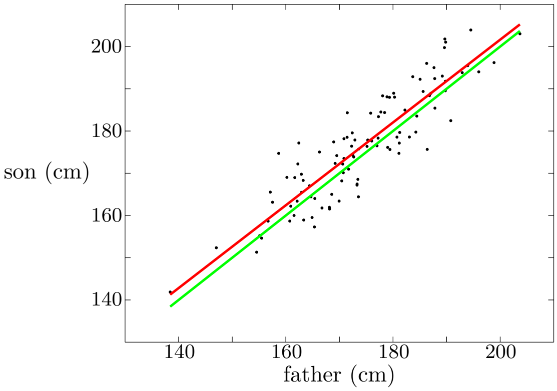


Figure 2.1: A linear fit of the height of sons onto the height of their fathers (synthetic data). The plot suggests that a linear model is appropriate and the fitted regression line (red) is close to the true function (green).

fully explained by the predictor variables. The remaining set of variables that relate to the response is either not known, or is deliberately excluded from the model to simplify the analysis. There are, however, reasons for including variables that are not of immediate interest. When such variables are significantly correlated with both the response and the predictors, their inclusion into the model may weaken, strengthen, or alter the significance of the results, giving a better understanding of the predictor variables of interest and their relation to the response. Variables that are not of primary interest but which must be included to obtain interpretable results are known as *confounding variables* (or simply *confounders*). In our example, such variables may for instance include environmental and genetic effects, and history of disease. When conducting an experiment, correct identification of confounding variables is an important part of the analysis to make sure that the results are correctly interpreted. A simple example gives more insight into the importance of including a suitable set of variables.

Imagine an investigation into the relationship between monthly ice-cream sales and drowning accidents. We don't expect these to be related, but to our surprise, a regression analysis points to a strong connection. Obviously, we failed to identify one or several important confounders. Assume one such variable is the monthly average temperature. Adding this variable to the analysis, the relationship between ice-cream sales and drowning accidents vanishes. The key point is that there is no *causal* relationship between the two. High temperature causes an increase in ice-cream sales and increased frequency of drowning accidents, but ice-cream sales does not cause drowning accidents.

The example shows that care must be taken when interpreting the results from a regression analysis. A strong mathematical connection between variables does not imply a causal relationship. There is no principled way of finding out whether an observed relationship between two variables is causal or due to unobserved variables. Instead, the analysis is commonly done the other way around. A hypothesis is made on the causal relationship between a response variable and one or more predictor variables, and we perform a regression analysis to see if the collected data support the hypothesis.

2.2.2 Linearity

The assumed linear model stated in Equation 2.2 is linear in terms of the regression coefficients but not necessarily its variables. This means that the models (excluding residual terms and intercept for brevity)

$$\mathbf{y} = b_1 \mathbf{x}_1^2 \qquad \mathbf{y} = b_1 e^{\mathbf{x}_1} \qquad \mathbf{y} = b_1 \log \mathbf{x}_1 + b_2 \sin(\sqrt{\mathbf{x}_2} + 5) \tag{2.4}$$

are also considered linear in this respect. This means that we are not necessarily restricted to regression functions that are straight lines. Suppose we suspect that the relationship between our measured response variable and a single independent variable is third order polynomial. This is modeled using a linear model by,

$$\mathbf{y} = b_0 + b_1 \mathbf{x}_1 + b_2 \mathbf{x}_1^2 + b_3 \mathbf{x}_1^3 + \mathbf{r}. \tag{2.5}$$

This is an important technique for generalizing linear statistical methods such that non-linear functions for regression, classification or clustering may be used. In Figure 2.2, a set of data points (black dots) has been created using Equation 2.5 with true parameters $\beta_0 = 5$, $\beta_1 = -2$, $\beta_2 = 9$, $\beta_3 = -8$, to which noise is added with $\mathbf{r}$ drawn from $N(0, 0.1)$. The green line represents the true function, while the red line represents a third order polynomial fit (using ordinary least squares fitting, see Section 2.3). The recovered parameters are $\mathbf{b} = [5.09 \quad -2.83 \quad 10.3 \quad -8.67]^T$. This technique, where a single variable is transformed and included in a model several times, is known as a *basis expansion.* We will return to this topic in Sections 3.3 and 3.4.

In the analysis of a real data set, the true form of the model is most often unknown. If a polynomial model is used, what is the appropriate order? If the chosen order is too low, the fitted regression function will not be able to capture the variance of the response. If the order is too high, the model will fit not only to the variance of interest, but also to the noise. This is known as *overfitting* and is avoided by careful *model selection.* Figure 2.3 shows an example of overfitting, where a tenth order polynomial is fitted to third-order data. Model selection is a central concept in regularized statistical analyses and thus, also in this thesis.

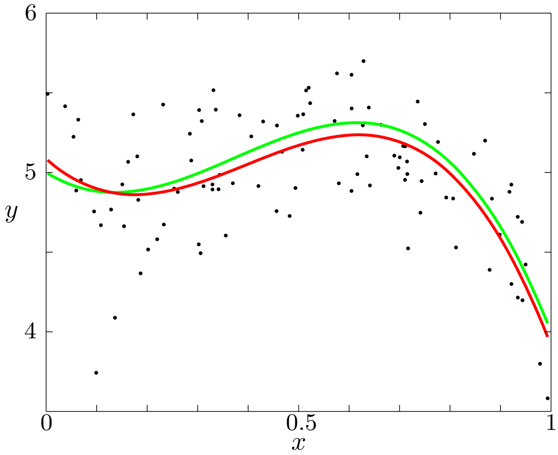


Figure 2.2: Example showing that linear modeling does not restrict the set of possible regression functions to straight lines. Here, a third-order polynomial (red) is fitted to a data set constructed from a noisy function of the same type (green).

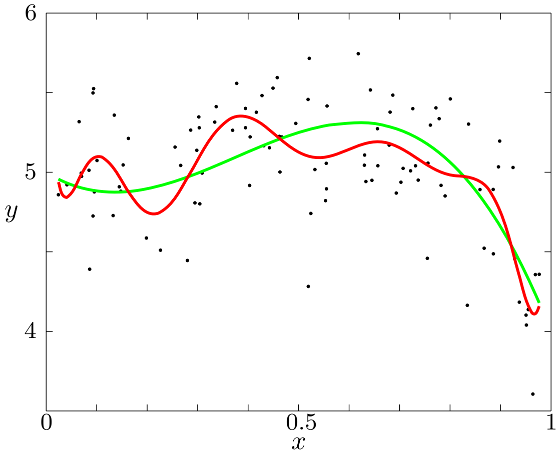


Figure 2.3: Example showing the importance of careful model selection. Here, a tenth-order polynomial (red) is fitted to perturbed third-order data (green), resulting in a poor match. Flexible models should be used with caution as they frequently suffer from *overfitting*, the ability to fit not only to the function of interest, but also to the noise.

2.2.3 Centering, Normalization and Standardization

In the remainder of this thesis, unless otherwise stated, all variables (predictors and responses) are assumed to be *mean centered.* For the observation of a variable $\mathbf{x}_i$ this means that

$$\bar{\mathbf{x}}_i = \sum_{j=1}^{n} x_{ji} = \mathbf{x}_i^T \mathbf{1}_n = 0. \tag{2.6}$$

In some cases, we further assume that the variables have been *normalized* or *standardized.* The meaning of these terms may differ slightly between researchers and research topics. Here, we define a normalized variable to be centered and of unit Euclidian length,

$$\sqrt{\sum_{j=1}^{n} x_{ji}^2} = \sqrt{\mathbf{x}_i^T \mathbf{x}_i} = 1, \tag{2.7}$$

and a standardized variable to be centered and with unit standard deviation,

$$\frac{1}{n}\sqrt{\sum_{j=1}^{n} x_{ji}^2} = \frac{1}{n}\sqrt{\mathbf{x}_i^T \mathbf{x}_i} = 1. \tag{2.8}$$

The difference between a normalized and a standardized variable is a simple scaling, which is sometimes chosen to be $1/(n-1)$ rather than $1/n$ as this leads to an unbiased estimate of the standard deviation (cf. Section 2.4). We usually settle for normalized variables as the inner products $\mathbf{x}_i^T \mathbf{x}_i = 1$ frequently simplify the expressions of which they are part.

Using the linear model, we can safely assume that the predictor variables have been centered and normalized and that the response has been centered. If the regression coefficients corresponding to the original variables are of interest, these can easily be obtained from the estimated coefficients. To see this[2], consider again the linear model in Equation 2.3,

$$\begin{aligned} \mathbf{y} &= b_0 + b_1\mathbf{x}_1 + \ldots + b_p\mathbf{x}_p + \mathbf{r} \Leftrightarrow \\ \mathbf{y} - \bar{\mathbf{y}} + \bar{\mathbf{y}} &= b_0 + \sum_{i=1}^{p} b_i \frac{\|\mathbf{x}_i - \bar{\mathbf{x}}_i\|}{\|\mathbf{x}_i - \bar{\mathbf{x}}_i\|}(\mathbf{x}_i - \bar{\mathbf{x}}_i + \bar{\mathbf{x}}_i) + \mathbf{r} \Leftrightarrow \\ \mathbf{y} - \bar{\mathbf{y}} &= \left[b_0 + \sum_{i=1}^{p} b_i \bar{\mathbf{x}}_i - \bar{\mathbf{y}} \right] + \sum_{i=1}^{p} b_i \frac{\mathbf{x}_i - \bar{\mathbf{x}}_i}{\|\mathbf{x}_i - \bar{\mathbf{x}}_i\|} \|\mathbf{x}_i - \bar{\mathbf{x}}_i\| + \mathbf{r}. \end{aligned} \tag{2.9}$$

[2] Here, we show the case where the predictors have been normalized. The proof for standardized variables proceeds in the same way.

Taking expectations of sides of this expression, we see that the equation inside the square brackets must equal zero. Therefore,

$$b_0 = \bar{\mathbf{y}} - \sum_{i=1}^{p} b_i \bar{\mathbf{x}}_i. \tag{2.10}$$

Performing a regression analysis using the linear model on centered and normalized predictors and a centered response corresponds to the model

$$\mathbf{y} - \bar{\mathbf{y}} = \sum_{i=1}^{p} \tilde{b}_i \frac{\mathbf{x}_i - \bar{\mathbf{x}}_i}{\|\mathbf{x}_i - \bar{\mathbf{x}}_i\|} + \mathbf{r}, \tag{2.11}$$

where the notation $\tilde{b}_i$ is used to emphasize that $\tilde{b}_i \neq b_i$. From the differences between this model and the original linear model, we infer that the transformation $b_i = \tilde{b}_i / \|\mathbf{x}_i - \bar{\mathbf{x}}_i\|$ can be used to obtain the untransformed regression coefficients for $i = 1 \dots p$. Thus, the intercept is obtained by,

$$b_0 = \bar{\mathbf{y}} - \sum_{i=1}^{p} \tilde{b}_i \frac{\bar{\mathbf{x}}_i}{\|\mathbf{x}_i - \bar{\mathbf{x}}_i\|} \tag{2.12}$$

Regardless of the method used to estimate the regression coefficients, the above exposition shows that we are free to center and normalize or standardize the variables as we see fit as long as a linear relationship between the response and the predictors is assumed. Again, the response and the predictors are assumed to be mean centered from this point onwards. This means that we can disregard the intercept and state the linear model,

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{r}. \tag{2.13}$$

2.3 Regression by Ordinary Least Squares

The linear model of Equation 2.13 describes the mathematical relationship between the variables of interest. To complete the description, we also need to specify how the regression coefficients are estimated. This amounts to defining an objective function measuring the merit of a particular solution. The most common approach is that of *ordinary least squares* (OLS), a method due to Carl Friedrich Gauss and dating back to the early 19th century. The formal description of OLS is,

$$\mathbf{b}_{\text{OLS}} = \arg\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 \tag{2.14}$$

We will motivate and derive an expression for the optimal solution to this minimization problem from a geometrical viewpoint. Consider a problem with three observations of a response variable $\mathbf{y}$ and two predictor variables $\mathbf{x}_1$ and $\mathbf{x}_2$. These variables can be visualized as vectors in a three-dimensional space, see Figure 2.4. The predictors span a plane $\mathcal{P}$ in $\mathbb{R}^3$. In the general case, the predictors span a p-dimensional hyperplane in $\mathbb{R}^n$. Each point in $\mathcal{P}$ defines a linear combination of the predictor variables and thus a solution $\hat{\mathbf{y}}$ for the regression equation. The OLS criterion states that the best solution is achieved where the squared length of the residual vector $\mathbf{r}$ is shortest. As can be realized from Figure 2.4, the residual vector is shortest when it is orthogonal to $\mathcal{P}$, yielding a unique point in $\mathcal{P}$. This means that the residual vector is orthogonal to both $\mathbf{x}_1$ and $\mathbf{x}_2$, or the columns of $\mathbf{X}$ in the general case. Two vectors are orthogonal if and only if their inner (dot) product is zero; we therefore have the following relationship,

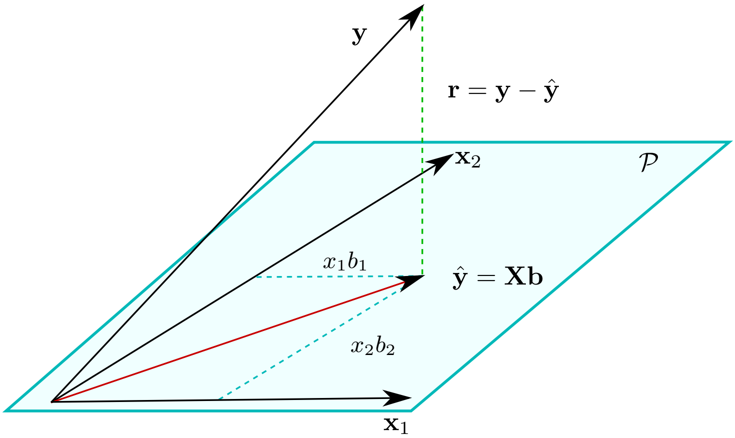

$\mathbf{y}$
$\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}}$
$\mathbf{x}_2$
$\mathcal{P}$
x_1b_1
$\hat{\mathbf{y}} = \mathbf{Xb}$
x_2b_2
$\mathbf{x}_1$

Figure 2.4: Geometry of the OLS solution for a problem with three observations in two dimensions. OLS attempts to minimize the (squared) length of the residual vector $\mathbf{r}$. The shortest such vector must be orthogonal to the plane $\mathcal{P}$ spanned by the predictor variables $\mathbf{x}_1$ and $\mathbf{x}_2$, ensuring a unique solution.

$$\mathbf{X}^T(\mathbf{y} - \mathbf{Xb}) = 0. \tag{2.15}$$

Solving this expression for $\mathbf{b}$ gives the minimizing parameters for OLS,

$$\mathbf{b}_{\mathrm{OLS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{2.16}$$

Using this expression, we can express the fit of the response variable as

$$\hat{\mathbf{y}} = \mathbf{Xb}_{\mathrm{OLS}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{Hy}, \tag{2.17}$$

where the matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is known as the "hat" matrix as it puts the hat on $\mathbf{y}$. Another term for $\mathbf{H}$ is the "projection" matrix as it projects $\mathbf{y}$ onto $\mathcal{P}$. A fitting method that can be written in this way is linear in terms of the random variable Y. Note the distinction here, all regression models considered in this chapter are linear, but the optimization problem used to establish $\mathbf{b}$ does not necessarily have a solution that can be written as an explicit function of $\mathbf{y}$, and obviously, even fewer functions have an optimal solution that is a linear function of $\mathbf{y}$. OLS is however linear in this regard, and another such regression method will be presented in Section 2.6.

For the OLS solution $\mathbf{b}_{\text{OLS}}$ to be defined, the $(p \times p)$ *gram* matrix $\mathbf{X}^T\mathbf{X}$ must be invertible. The rank of the gram matrix is at most $\min(n, p)$, which means that the number of observations n must be equal to or greater than the number of variables p for the gram matrix to be full rank. Further, the rank is reduced if there are linearly dependent columns (variables) of $\mathbf{X}$. This is known as *multicollinearity* in the regression setting. Regularization can be used to improve the condition of analyses where lack of observations and/or multicollinearity make OLS infeasible. Before regularized procedures for regression are regarded, we will go into more details about the OLS solution.

2.3.1 The Gram-Schmidt Procedure

To gain more insight into the machinery of OLS regression we will review the case where the predictor variables are orthogonal. Assuming this is the case, the gram matrix $\mathbf{X}^T\mathbf{X}$ is diagonal of the form

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T\mathbf{x}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{x}_p^T\mathbf{x}_p \end{bmatrix}. \tag{2.18}$$

Using this matrix to solve the OLS Equation (2.16), we see that each b_i is a function of the i^{th} variable $\mathbf{x}_i$ and $\mathbf{y}$ only such that,

$$b_i = (\mathbf{x}_i^T\mathbf{x}_i)^{-1}\mathbf{x}_i^T\mathbf{y}. \tag{2.19}$$

Such a regression equation, where a single independent variable is considered is called *univariate*, as opposed to the more general *multiple*[3] regression approach of Equation 2.13. It is also seen that univariate regression amounts to an orthogonal projection of $\mathbf{y}$ onto $\mathbf{x}_i$, where b_i is the length of the image of $\mathbf{y}$ on $\mathbf{x}_i$. The conclusion is that the OLS estimates are particularly simple to calculate in the univariate case. In the non-orthogonal case, we cannot regard the variables

[3] The term *multivariate* regression is perhaps more suitable here, but is reserved for procedures that regard multiple response variables at once.

one by one, as several variables contribute to the description of $\mathbf{y}$ in a particular direction. Figure 2.5 depicts these two situations. The left example has orthogonal predictors $\mathbf{z}_1$ and $\mathbf{z}_2$. Here, each multiple regression coefficient is given by univariate regression on each variable independently. In the right example, the predictors are linearly dependent. Trying to determine the multiple regression coefficients using univariate regression leads to overestimation of each b_i in this case as indicated by the dashed projection lines.

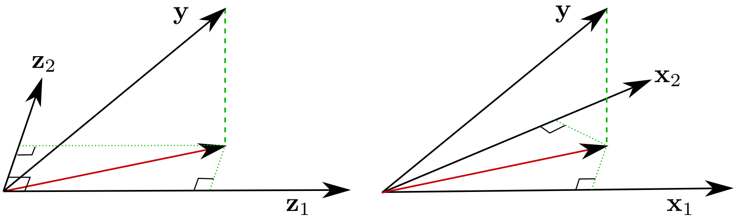


Figure 2.5: Regression on orthogonal predictors (left) versus the usual non-orthogonal case (right). In the orthogonal design, each multiple regression coefficient b_i can be obtained by regressing $\mathbf{y}$ on $\mathbf{z}_i$ alone, whereas in the non-orthogonal case, the sum of the vectors obtained through univariate regression will not give the correct $\hat{\mathbf{y}}$.

In cases where the predictor variables are non-orthogonal, new variables can be derived which span the same hyperplane $\mathcal{P}$ as the original variables and which are orthogonal. Since the same hyperplane is regarded, the fitted vector $\hat{\mathbf{y}}$ will be the same regardless of whether the original or derived orthogonal variables are used. The procedure that yields the orthogonal variables is simple. Let the first original variable be the first orthogonal direction, $\mathbf{z}_1 = \mathbf{x}_1$. Next, remove the presence of $\mathbf{z}_1$ in $\mathbf{x}_2$, forming the second orthogonal variable $\mathbf{z}_2 = \mathbf{x}_2 - \mathbf{z}_1(\mathbf{z}_1^T\mathbf{z}_1)^{-1}\mathbf{z}_1^T\mathbf{x}_2$. The third orthogonal variable is fashioned in the same way, through orthogonalization first with respect to $\mathbf{z}_1$, then to $\mathbf{z}_2$. The process is repeated until p orthogonal variables are obtained. This procedure is known as *Gram-Schmidt orthogonalization*, and we present it in more detail in Algorithm 2.1, where we also create a matrix $\mathbf{G}$, specifying the mapping between $\mathbf{X}$ and $\mathbf{Z}$ such that $\mathbf{X} = \mathbf{ZG}$.

Algorithm 2.1 Gram-Schmidt orthogonalization

```
Initialize $\mathbf{Z} = \mathbf{x}_1$
for $j = 2$ to $p$ do
    Add column $\mathbf{x}_j - \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{x}_j$ to $\mathbf{Z}$
end for
$\mathbf{G} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{X}$
Output $\mathbf{Z}$ and $\mathbf{G}$.
```

The i^{th} column of $\mathbf{G}$ specifies the linear combination of the derived variables $\mathbf{z}_1 \ldots \mathbf{z}_p$ that gives $\mathbf{x}_i$. By inspection of the Gram-Schmidt process, it is seen that this matrix is upper triangular. Inserting $\mathbf{X} = \mathbf{ZG}$ into Equation 2.16 and simplifying, the OLS solution can be written

$$\mathbf{b}_{\text{OLS}} = \mathbf{G}^{-1}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{y} = \mathbf{G}^{-1}\tilde{\mathbf{b}}, \tag{2.20}$$

where $\tilde{\mathbf{b}}$ represents the OLS regression coefficients of $\mathbf{y}$ with respect to $\mathbf{Z}$. This shows that we can obtain the multiple regression solution $\mathbf{b}_{\text{OLS}}$ by first deriving the orthogonal predictor variables $\mathbf{Z}$ by the Gram-Schmidt process, then finding $\tilde{\mathbf{b}}$ through univariate regression of $\mathbf{y}$ on each $\mathbf{z}_i$ and then transforming the obtained coefficients using $\mathbf{G}^{-1}$.

This process may appear cumbersome, but is computationally efficient for large problems, where inversion of the gram matrix $\mathbf{X}^T\mathbf{X}$ can be avoided as the entire process can be expressed in terms of univariate regression problems and a single inversion of the matrix $\mathbf{G}$, which can be done efficiently since $\mathbf{G}$ is triangular. This thesis contains several examples of techniques related to the Gram-Schmidt process.

The algorithm also sheds light on the meaning of the multiple regression coefficient $\mathbf{b}_i$. Noting that both $\mathbf{G}$ and $\mathbf{G}^{-1}$ are upper triangular with ones along the diagonal, Equation 2.20 shows that the last coefficient $b_p = \tilde{b_p}$. This finding leads to the following interpretation of b_i as described by Hastie et al. [59].

> The multiple regression coefficient b_i represents the additional contribution of $\mathbf{x}_i$ on $\mathbf{y}$, after $\mathbf{x}_i$ has been adjusted for $\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}$, $\mathbf{x}_{i+1}, \ldots, \mathbf{x}_p$.

The matrices $\mathbf{Z}$ and $\mathbf{G}$ can be efficiently calculated using the orthogonal-triangular (QR) decomposition of the data matrix $\mathbf{X}$,

$$\mathbf{X} = \mathbf{QR}, \tag{2.21}$$

where $\mathbf{Q}$ is an orthogonal matrix such that $\mathbf{Q}^T = \mathbf{Q}^{-1}$ and $\mathbf{R}$ is upper triangular. The matrices $\mathbf{Z}$ and $\mathbf{G}$ can be obtained from $\mathbf{Q}$ and $\mathbf{R}$ using the diagonal scaling matrix $\mathbf{D}$ with $\text{diag}(\mathbf{D}) = \text{diag}(\mathbf{R})$. The decomposition can then be written $\mathbf{X} = \mathbf{QDD}^{-1}\mathbf{R} = \mathbf{ZG}$. Using the QR representation of $\mathbf{X}$, the OLS solution can be simplified into

$$\mathbf{b}_{\text{OLS}} = \mathbf{R}^{-1}\mathbf{Q}^T\mathbf{y}, \qquad \hat{\mathbf{y}} = \mathbf{QQ}^T\mathbf{y}. \tag{2.22}$$

2.4 Bias and Variance

To this point, the only assumption about the model has been that the true relationship between Y and X is approximately linear. We now state two additional assumptions that are necessary for measuring the accuracy of a solution.

- $E(\varepsilon) = 0$
- The errors ε are independent. This implies that observations y_i of the response Y are also independent. If, for instance, measurements are made over time, there must be no dependence of the y_i on time.
- The errors must have finite constant variance $\text{var}(\varepsilon) = \sigma^2$ (homoscedasticity). This implies that Y is also homoscedastic.

Armed with these assumptions, we now discuss the accuracy of a general estimator, and the OLS estimator in particular.

To measure the accuracy of a solution, we are not only interested in the magnitude of the regression coefficients. One must also estimate the extent to which a regression coefficient is expected to vary between trials. In general, we have less faith in coefficients which vary enough to switch signs when an experiment is repeated. The variance-covariance matrix of $\mathbf{b}$ is easily derived (cf. Equation 1.7),

$$\begin{aligned}\text{var}(\mathbf{b}_{\text{OLS}}) &= \text{var}\left[(X^TX)^{-1}X^TY\right] \\ &= (X^TX)^{-1}X^TX(X^TX)^{-1}\text{var}(X\beta + \varepsilon) \\ &= (X^TX)^{-1}\text{var}(\varepsilon). \qquad (2.23)\end{aligned}$$

The error variance $\text{var}(\varepsilon) \equiv \sigma_\varepsilon^2$ can be approximated by the sum of squared residuals normalized by the number of degrees of freedom of the residuals,

$$\sigma_\varepsilon^2 \approx \hat{\sigma}_\varepsilon^2 = \frac{\mathbf{r}^T\mathbf{r}}{n-p}. \qquad (2.24)$$

The estimate of the variance of a regression coefficient is used to measure the reliability with which one can assume that the true coefficient is zero. In short, the procedure is as follows. The estimated coefficient b_i is normalized by its standard deviation, thus forming a standardized coefficient or z-score,

$$z_i = \frac{b_i}{\text{std}(b_i)}, \qquad (2.25)$$

where $\text{std}(b_i)$ is the i^{th} diagonal value of $\text{var}(\mathbf{b})$. Under the hypothesis $\beta_i = 0$, the z-score is approximately normal with mean zero and unit variance. Placing the z-score on this distribution, a measure of the probability that z_i comes from this distribution is given. The farther z_i is from zero, the smaller the probability and the greater the confidence with which the null hypothesis can be rejected. A test where both negative and positive values of z_i are regarded is called *two-sided* whereas a test where either positive of negative values are regarded is called *one-sided.*

The *bias* of an estimator is the difference between the average estimator over a large set of trials and the true regression function. Denote the (unknown) true regression function $f(X)$ and the estimated regression function $\hat{f}(X|D)$, where D denotes the data set of a single trial. Averaging over many trials amounts to taking the expectation of $\hat{f}$ over all possible data sets, $E_D\hat{f}(X|D)$. For the OLS estimator the expectation yields,

$$E_D(\hat{f}(X|D)) = E_D(X\mathbf{b}) = X(X^TX)^{-1}X^TE_D(Y|X) = X\beta = f(X). \tag{2.26}$$

The measure $E_D(\hat{f}(X|D)) - f(X)$ is the bias. The equation shows that OLS is an *unbiased* estimator. Obviously, this is a desirable property, but as we shall see, there are biased estimators which may be preferred.

Biased or not, the most wanted property of an estimated regression function is that it describes the phenomenon under study well and is able to do predictions with high accuracy. The expected prediction error (EPE) of a function $\hat{f}$ can be written $\text{EPE}_{\hat{f}}(X) = E_{D,Y}\left[(Y - \hat{f}(X|D))^2\right]$, where the expectation is taken over both all data sets D and all responses Y. The EPE can be decomposed into quantities that help us understand the ways in which an estimator can be improved. Augmenting the EPE by addition and subtraction of $f(X)$, we get,

$$\begin{aligned}
\text{EPE}_{\hat{f}}(X) =& E_{D,Y}\left[(Y - \hat{f}(X|D))^2\right] \\
=& E_{D,Y}\left[(Y - f(X) + f(X) - \hat{f}(X|D))^2\right] \\
=& E_Y\left[(Y - f(X))^2\right] + E_D\left[(f(X) - \hat{f}(X|D))^2\right] + \\
& 2E_{D,Y}\left[(Y - f(X))(f(X) - \hat{f}(X|D))\right] \\
=& \sigma_\varepsilon^2 + E_D\left[(f(X) - \hat{f}(X|D))^2\right].
\end{aligned} \tag{2.27}$$

The double product term vanishes using that $Y = f(X) + \varepsilon$, $f(X)$ constant, $E(\varepsilon) = 0$, and ε and $\hat{f}(X|D)$ are independent and hence $E_{D,Y}(\hat{f}(X|D)\varepsilon) = 0$. The derivation shows that the EPE is a function of the noise variance $\text{var}(\varepsilon) = \sigma_\varepsilon^2$ and the mean square error between the true and estimated regression functions.

Using the augmentation trick a second time on the latter term yields,

$$
\begin{aligned}
E_D\left[(f(X)-\hat{f}(X|D))^2\right] &= \\
E_D\left[(f(X)-E_D\hat{f}(X|D)+E_D\hat{f}(X|D)-\hat{f}(X|D))^2\right] &= \\
E_D\left[(f(X)-E_D\hat{f}(X|D))^2\right]+E_D\left[E_D\hat{f}(X|D)-\hat{f}(X|D))^2\right] &+ \\
2E_D\left[(f(X)-E_D\hat{f}(X|D))(E_D\hat{f}(X|D)-\hat{f}(X|D))\right] &= \\
(f(X)-E_D\hat{f}(X|D))^2+E_D\left[(E_D\hat{f}(X|D)-\hat{f}(X|D))^2\right]. & \quad (2.28)
\end{aligned}
$$

Again, the double product vanishes using that $E_D\hat{f}(X|D)$ is constant. Putting the pieces together, we get the following expression for the EPE,

$$
\begin{aligned}
\mathrm{EPE}_{\hat{f}}(X) &= \sigma_\varepsilon^2+(f(X)-E_D\hat{f}(X|D))^2+E_D\left[(E_D\hat{f}(X|D)-\hat{f}(X|D))^2\right] \\
&= \sigma_\varepsilon^2+\mathrm{bias}(\hat{f}(X|D))^2+\mathrm{var}(\hat{f}(X|D)), \qquad (2.29)
\end{aligned}
$$

that is, the EPE is a sum of the noise variance, the squared bias and the variance of the estimated function. The variance σ_ε^2 of the errors can only be lowered by changing the model, usually through the inclusion of confounding variables. Assuming the model is fixed, the only way to decrease prediction error is to work with the bias and variance terms. Below, we show that if we require that the estimator is unbiased, we cannot get lower variance and hence lower EPE than we get with OLS. The conclusion is that we must introduce bias to improve our estimator.

2.4.1 The Gauss-Markov Theorem

In Section 2.3, we established that the OLS estimator produces a reconstruction $\hat{\mathbf{y}}$ that is as close as possible to the response variable $\mathbf{y}$ in terms of the (squared) residual length. It is also optimal in another sense. If we repeat the regression analysis with new input data from the same experiment, we would prefer it if small differences among the predictor variables result in the smallest possible differences in the corresponding response variable. Translated into statistical terms, this corresponds to minimal variance of $\hat{f}(\mathbf{X})$. The Gauss-Markov theorem states that among all unbiased linear estimators, OLS is the one with minimal variance. In Figure 2.6, we attempt to provide an intuitive explanation of this property.

The green sphere has been positioned at $\mathbf{y}$ and represents the variance of Y. This representation is correct, since the variance of Y is the same in all directions according to the assumption of homoscedasticity above. In the plane spanned

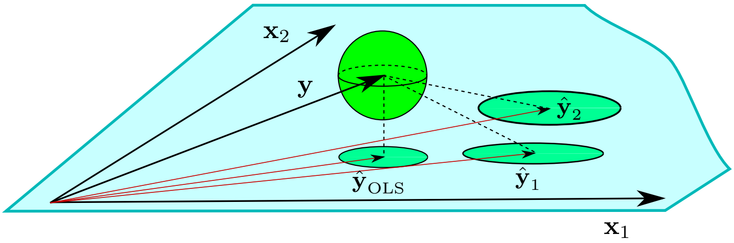


Figure 2.6: Geometry of the Gauss-Markov theorem. OLS represents the projection of $\mathbf{y}$ that gives minimal variance of the fitted vector $\hat{\mathbf{y}}$. The variance of $\mathbf{y}$ is represented by a green sphere, while the OLS projection and two non-orthogonal projections of this sphere onto the plane spanned by the predictors are represented by a circle and two ellipses respectively.

by the predictor variables, three solutions are shown, the OLS solution and two non-orthogonal alternatives. At each solution, the (parallel) projection of the variance of $\mathbf{y}$ shows the corresponding variance of $\hat{\mathbf{y}}$. This suggests the Gauss-Markov property of the OLS estimator, the smallest variance of $\hat{\mathbf{y}}$ is achieved at the OLS solution where the projection is a circle with the same radius as the variance sphere. At the other solutions, this projection becomes ellipses, all which are big enough to encompass the OLS variance circle. Minimal variance of the fitted vector $\hat{\mathbf{y}}$ implies that the variance is also minimal for the regression coefficients b_i since, according to Equation 2.23, the variance of b_i is $\text{var}(b_i) = (X_i^T X_i)^{-1}\text{var}(Y) = (X_i^T X_i)^{-1}\sigma_\varepsilon$.

2.5 Pointwise Regression

Previous sections serve to motivate the use of regularization in regression models. We have seen that no linear unbiased method outperforms OLS. However, if some bias is tolerated we can potentially lower the variance of the estimates and the prediction error considerably as well as handle cases where $p > n$ and/or data plagued by multicollinearity. The question is how bias is added to the model in a sensible way.

A simple but often used regularization approach is given by the assumption that the independent variables are uncorrelated. Under this assumption, the analysis is simplified into that of the orthogonal design described in Section 2.3.1. We will reiterate the computational implications of this here. The assumption represents a type of regularization that will result in an invertible gram matrix also in cases where $p > n$. Unless any of the variables have zero variance, the augmented gram matrix has positive values along its diagonal and zeros elsewhere,

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T\mathbf{x}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{x}_p^T\mathbf{x}_p \end{bmatrix}. \tag{2.30}$$

Such a matrix is positive definite and therefore invertible. Using this gram matrix to solve the ordinary least squares problem of Equation 2.16, we see that each regression coefficient b_i is a function of the i^{th} variable $\mathbf{x}_i$ and $\mathbf{y}$ such that,

$$b_i = (\mathbf{x}_i^T\mathbf{x}_i)^{-1}\mathbf{x}_i^T\mathbf{y}. \tag{2.31}$$

If the original ill-posed problem consists of, say, 900 variables and 100 observations, pointwise regression splits the analysis into 900 separate ordinary least squares analyses, each well-posed with 100 observations and a single variable.

In Section 2.3.1, we referred to Equation 2.31 as *univariate* while we opt for the term *pointwise* here. Multivariate analyses with $p \gg n$ usually occur in problems where the majority of variables are of the same type, such as spatial variables in image analysis, or gene expression measurements in microarray analysis. In addition to such variables, a small set of confounding variables may be included. In such cases, the analysis is split up such that each analysis contains a single variable of interest along with the full set of confounding variables. This makes each analysis a multivariate, albeit small, regression problem, making use of the term univariate misleading. To clarify, let $\tilde{\mathbf{X}}_i = [\mathbf{x}_i \quad \mathbf{c}_1 \ldots \mathbf{c}_{p_c}]$ be a data matrix which includes the i^{th} variable of interest and let $\mathbf{c}_j$ be the j^{th} confounding variable, $j = 1 \ldots p_c$. A single pointwise analysis is then performed using this matrix and the OLS approach of Section 2.3.

2.6 Ridge Regression

Many regularization methods use a technique called *coefficient shrinkage* to lower the variance of $\mathbf{y}$ (and $\mathbf{b}$). This means that the regression coefficients b_i are shrunk from their corresponding OLS estimates. Obviously, this will lower the variance of the estimates; if the coefficients are shrunk all the way to $\mathbf{b} = \mathbf{0}$, the variance is zero. Although such a model is pointless, the idea of coefficient shrinkage methods is that for a moderate amount of shrinkage, both lower variance and lower prediction error may be obtained. In the presence of multicollinearity, the OLS coefficients pertaining to two strongly correlated variables may differ wildly. For instance, a large coefficient on one variable can be cancelled by an equally large negative coefficient on the other. By restricting the size of the regression coefficients, this is prevented from happening. This is one reason why the OLS estimates, having the lowest variance of all unbiased estimators, have rather high variance compared to suitably biased estimators.

A simple form of coefficient shrinkage is implemented by the addition of a quadratic penalty term on the regression coefficients,

$$\mathbf{b}_{\text{ridge}} = \arg\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \lambda\|\mathbf{b}\|^2, \tag{2.32}$$

where $\lambda \geq 0$ is a parameter that controls the amount of regularization. A positive value of λ emphasizes solutions with regression coefficients of smaller magnitude. This shrinkage is strengthened as λ grows, and excessive regularization will force all coefficients to zero. The cost function above is specified in the *loss + penalty* form, where the loss function is the residual sum of squares term of OLS, and the penalty function is the squared ℓ_2-norm of the regression coefficients.

We derive the optimal $\mathbf{b}_{\text{ridge}}$ by differentiating Equation 2.32 and equaling zero,

$$\begin{aligned} \frac{\partial}{\partial \mathbf{b}}\left[\|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \lambda\|\mathbf{b}\|^2\right] &= \\ \frac{\partial}{\partial \mathbf{b}}\left[\mathbf{y}^T\mathbf{y} - 2\mathbf{b}^T\mathbf{X}^T\mathbf{y} + \mathbf{b}^T(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{b}\right] &= \\ -2\mathbf{X}^T\mathbf{y} + 2\mathbf{b}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}) &= 0, \end{aligned} \tag{2.33}$$

and solving for $\mathbf{b}$,

$$\mathbf{b}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}, \tag{2.34}$$

where $\mathbf{I}$ is the $p \times p$ identity matrix. The computational effect of ridge regression is evident; a small constant is added to the diagonal of the gram matrix. In cases where the gram matrix is not invertible, this augmentation gives the resulting matrix full rank which therefore can be inverted. Further, $\lambda = 0$ gives the ordinary least squares solution, in cases where the gram matrix can be inverted. Ridge regression represents a linear projection of $\mathbf{y}$, similar to OLS. The hat matrix is

$$\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T. \tag{2.35}$$

Figure 2.7(a) shows coefficient values obtained using ridge regression for a range of values of λ on a data set with 442 observations and 10 variables. This type of plot is known as a *ridge trace*. The data set is from a study of diabetes where the response variable measures disease progress one year after baseline[4]. The

[4] The word baseline refers to the start of a clinical investigation that runs over a certain time span. Such a study is known as a longitudinal study as opposed to a cross-sectional study where a data is gathered at a single occasion. This study falls somewhere in between, as data for each variable is gathered once, but at different time points.

predictor variables are gathered at baseline and consist of age, sex, body mass index (BMI), blood pressure and six blood serum measurements[5]. The goal is to construct a model that can be used to predict disease progression from baseline data. The data set was divided into three parts; a training set (the first 221 observations), a validation set (the next 111 observations) and a test set (the final 110 observations). The training set was used to estimate ridge regression coefficients for 100 values of the regularization parameters λ, ranging between 10^{-4} and 10^3 on a logarithmic scale. The validation set was then used to select a suitable model based on a measure of prediction error using the sum of squared residuals $\mathbf{r}^T\mathbf{r} = \|\mathbf{y} - \mathbf{X}\mathbf{b}\|$, where $\mathbf{y}$ and $\mathbf{X}$ represent the validation data set, and $\mathbf{b}$ is estimated from training data. Figure 2.7(b) shows the resulting plot of prediction errors. The plot shows that the coefficient shrinkage introduced by ridge regression indeed lowers the prediction error for certain values of λ. Finally, a fair estimate of the true prediction error can then be calculated using the test set. In this example, the prediction error was $3.33 \cdot 10^5$, only slightly higher than the corresponding validation error $3.30 \cdot 10^5$.

Similarly to the OLS estimates, an expression for the variance of the ridge regression coefficients can be derived. This proceeds in the same manner as in Equation 2.23, yielding,

$$\mathrm{var}(\mathbf{b}_{\mathrm{ridge}}) = (X^T X + \lambda\mathbf{I})^{-1} X^T X (X^T X + \lambda\mathbf{I})^{-1} \sigma_\varepsilon^2. \tag{2.36}$$

The estimation of the error variance σ_ε^2 requires some attention. In Equation 2.24, the sum of squared residuals are normalized by the number of degrees of freedom for the residuals, estimated by $n-p$. In ridge regression, the regularization parameter controls the complexity of the model, creating an increasingly fixed model for growing values of λ. The result is that the number of degrees of freedom of the residuals grows with λ. This makes clear the need for a better estimate of the number of parameters of the model than p. Moody [102] discusses a measure called *the effective number of parameters* for this purpose,

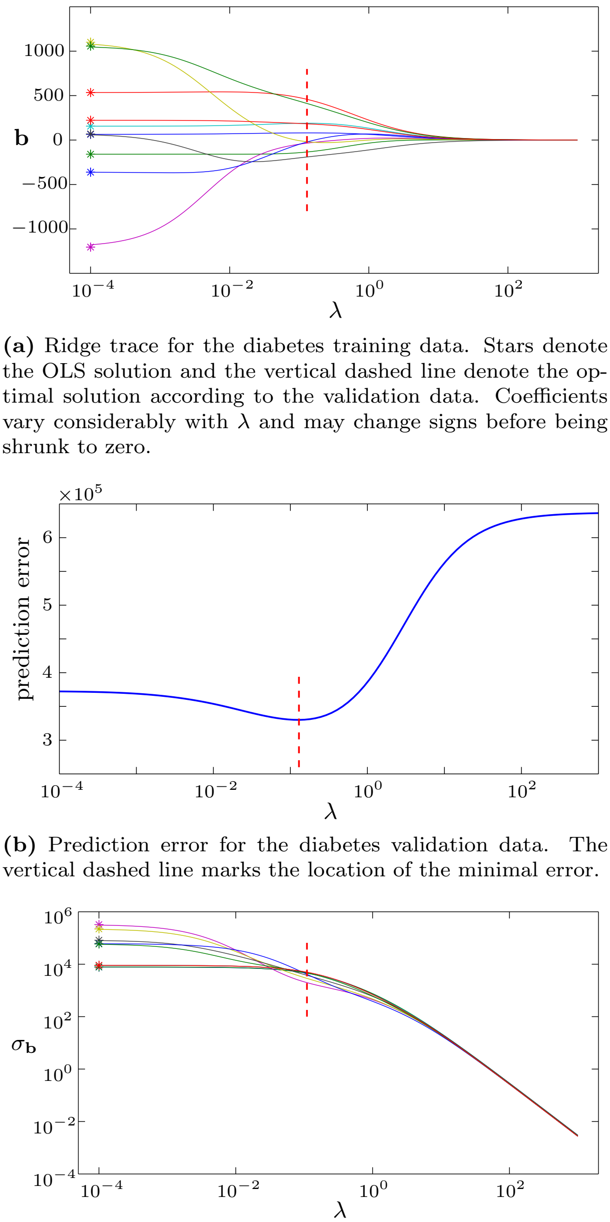
$$D_f(\lambda) = \mathrm{trace}(\mathbf{H}), \tag{2.37}$$

where $\mathbf{H}$ is the hat matrix defined in Equation 2.35. Working out the algebra, it is seen that $D_f(0) = p$, the number of parameters of the OLS solution. For $\lambda \to \infty$, $D_f(\lambda) = 0$. The estimate of error variance becomes

$$\hat{\sigma}_\varepsilon^2 = \frac{\mathbf{r}^T\mathbf{r}}{n - D_f(\lambda)}. \tag{2.38}$$

Figure 2.7(c) shows the variance of each b_i as a function of λ computed according to Equation 2.36. As expected, the variances shrinks for growing values of λ;

[5] This data set is distributed with the R statistical computing environment and is part of the `lars` package for least angle regression (cf. Efron et al. [40] and Section 2.8). The software is freely available from `www.r-project.org`.



(a) Ridge trace for the diabetes training data. Stars denote the OLS solution and the vertical dashed line denote the optimal solution according to the validation data. Coefficients vary considerably with λ and may change signs before being shrunk to zero.

(b) Prediction error for the diabetes validation data. The vertical dashed line marks the location of the minimal error.

(c) Variance of the ridge regression estimates versus the variance of the OLS estimates (stars). The variance is considerably lowered at the optimal value of λ.

Figure 2.7: Ridge regression on the diabetes data set. The parameter λ is defined by 100 equidistant points on a logarithmic scale.

the variance of the optimal model is considerably lower than the corresponding OLS estimate.

2.6.1 Computation

In Section 2.1, we saw that the QR decomposition could be used to simplify computation of the OLS estimates. Here, we present a similar technique, based on the singular value decomposition (SVD). The SVD of a matrix $\mathbf{X}$ expresses $\mathbf{X}$ in terms of two orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$, and a diagonal matrix $\mathbf{D}$,

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T, \tag{2.39}$$

where the diagonal elements of $\mathbf{D}$ are typically sorted in descending order and are called the *singular values* of $\mathbf{X}$. Using the SVD, the ridge estimate can be written

$$\begin{aligned}\mathbf{b}_{\text{ridge}} &= (\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{U}\mathbf{D}\mathbf{V}^T + \lambda\mathbf{I})^{-1}\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{y}\\ &= (\mathbf{V}(\mathbf{D}^2 + \lambda\mathbf{I})\mathbf{V}^T)^{-1}\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{y}\\ &= \mathbf{V}(\mathbf{D}^2 + \lambda\mathbf{I})^{-1}\mathbf{D}\mathbf{U}^T\mathbf{y},\end{aligned} \tag{2.40}$$

noting that $\mathbf{V}^{-1} = \mathbf{V}^T$ and using Equation 1.1. The advantage of this formulation is that $\mathbf{D}^2 + \lambda\mathbf{I}$ is a diagonal matrix and, as such, easy to invert. Using this formulation, the hat matrix can be written

$$\mathbf{H} = \mathbf{U}\mathbf{D}(\mathbf{D}^2 + \lambda\mathbf{I})^{-1}\mathbf{D}\mathbf{U}^T = \sum_{i=1}^{p} \mathbf{u}_i \frac{d_i^2}{d_i^2 + \lambda}\mathbf{u}_i^T, \tag{2.41}$$

and the effective number of parameters is simplified into

$$D_f(\lambda) = \sum_{i=1}^{p} \frac{d_i^2}{d_i^2 + \lambda}. \tag{2.42}$$

Finally, the variance of the parameters can be efficiently computed using the SVD transform,

$$\text{var}(\mathbf{b}_{\text{ridge}}) = \mathbf{V}^2\text{diag}\left[\mathbf{D}^2(\mathbf{D}^2 + \lambda\mathbf{I})^{-2}\right]\hat{\sigma}_\varepsilon^2. \tag{2.43}$$

So far, the advantage of using the SVD transform in the formulation of ridge regression is that $\mathbf{D}^2 + \lambda\mathbf{I}$ is a diagonal matrix which is straightforward to invert. In cases where $p > n$, another advantage arises. The corresponding data matrix in such cases has $\text{rank}(\mathbf{X}) = k \leq n$. The last $p - k$ singular values of $\mathbf{X}$ are therefore zero and we can express $\mathbf{X}$ using the *economy size* SVD.

$$\mathbf{X} = \underset{n\times k}{\mathbf{U}}\,\underset{k\times k}{\mathbf{D}}\,\underset{k\times p}{\mathbf{V}^T} \tag{2.44}$$

Using this decomposition of $\mathbf{X}$, Equation 2.40 still renders the ridge solution in Equation 2.34. The formulation does, however, result in a significant computational difference. While the diagonal matrix to be inverted has size $(p \times p)$ in Equation 2.34, it has size $(k \times k)$ in Equation 2.40.

2.6.2 Relation to Pointwise Regression

The variance estimates of Equation 2.43 can be used to turn the ridge regression coefficients into z-scores as described in Equation 2.25. Figure 2.8 shows the obtained z-scores for different values of λ. For low values of λ, the scores are rather stable and close to the z-scores of the OLS solution (stars). For high values of λ the scores again converge to stable values. A relevant question concerns the meaning of the ridge regression solutions in the limit $\lambda \to \infty$. It turns out that the solutions have a strong relation to those of pointwise regression. This result, presented in the following, is intuitive. Both methods focus on the variance of the variables rather than their covariance. Studying the ridge trace in Figure 2.7(a), it is seen that the coefficients are shrunk as λ grows, and depart from the coefficients of pointwise and OLS regression which have similar magnitudes. However, the corresponding z-scores behave differently, and turn out to be roughly equal to those of pointwise regression in the limit. The results from pointwise regression are shown in Figure 2.8 at the far right using the symbol ×. We investigate the equivalence between ridge regression and pointwise regression in the following theorem.

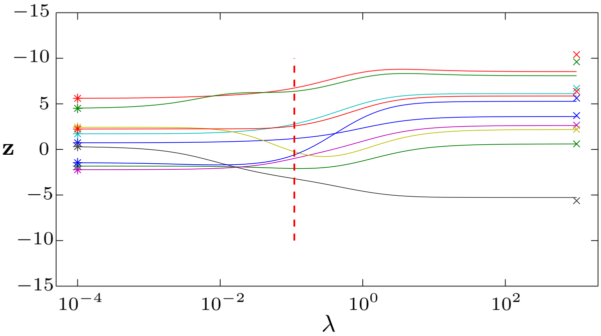


Figure 2.8: Plot showing the z-scores of each regression coefficient as a function of λ. Results seem to gain in significance as λ grows, and converges for λ sufficiently large.

Theorem 2.1 *In the limit $\lambda \to \infty$, the normalized coefficients of ridge regression and pointwise regression are equivalent.*

Proof. Let $\boldsymbol{\Omega} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}$ and let $\boldsymbol{\omega}_i$ be the i^{th} column of $\boldsymbol{\Omega}$. The i^{th} z-score resulting from ridge regression is then

$$z_i(\lambda) = \frac{\beta_i(\lambda)}{\sigma_{\beta_i}(\lambda)} = \frac{\boldsymbol{\omega}_i^T\mathbf{X}^T\mathbf{y}}{\sigma_{\boldsymbol{\varepsilon}}\sqrt{\boldsymbol{\omega}_i^T\mathbf{X}^T\mathbf{X}\boldsymbol{\omega}_i}} \tag{2.45}$$

We note that $\boldsymbol{\Omega} \to \mathbf{I}/\lambda$ as $\lambda \to \infty$. This means that $\boldsymbol{\omega}_i$ simplifies to a zero vector with the i^{th} entry equal to $1/\lambda$. The expression for $z_i(\lambda)$ becomes

$$\lim_{\lambda\to\infty} z_i(\lambda) = \frac{\frac{1}{\lambda}\mathbf{x}_i^T\mathbf{y}}{\frac{1}{\lambda}\sigma_{\boldsymbol{\varepsilon}}\sqrt{\mathbf{x}_i^T\mathbf{x}_i}} = \frac{(\mathbf{x}_i^T\mathbf{x}_i)^{-1}\mathbf{x}_i^T\mathbf{y}}{\sigma_{\boldsymbol{\varepsilon}}\sqrt{(\mathbf{x}_i^T\mathbf{x}_i)^{-1}}} \tag{2.46}$$

This expression is equivalent to the pointwise estimation of z_i.

This finding points to an important property of ridge regression; OLS regression and pointwise regression are special cases of ridge regression. Whereas OLS represents an unbiased ridge regression analysis, pointwise regression is severely biased. This motivates the use of ridge regression in cases where pointwise regression is normally used; it is unlikely that the amount of bias introduced in pointwise regression yields the lowest prediction error. Instead, consideration of cases with less bias, where correlation information is allowed to some extent, may be beneficial. The result also makes clear that values of λ over a certain threshold are uninteresting, as the ridge solutions have converged to pointwise regression at that point. In fact, we can derive an expression for an upper limit of λ. This limit can be defined in terms of the value $d_{max} = \max_i d_i$ as

$$\lambda_{max} = d_{max}^2\frac{1-\epsilon}{\epsilon}. \tag{2.47}$$

For this choice of λ, the elements of the matrix $(\mathbf{D}^2 + \lambda\mathbf{I})^{-1}$ will deviate at most 100ϵ % from the matrix $\mathbf{I}/\lambda$, where ϵ is a small number $0 < \epsilon < 1$.

The derivation in Theorem 2.1 assumes that the error variance σ_ε^2 is measured in the same way for both methods. This is usually not the case. Equation 2.36 gives this estimate for ridge regression while the estimate for pointwise regression is $\sigma_\varepsilon^2 = \mathbf{r}_i^T\mathbf{r}_i/(n-p)$ where $p = 1 + p_c$ (ignoring intercept) and p_c is the number of confounding variables, if any. However, performing the same asymptotic analysis as above for the error variance of ridge regression gives,

$$\lim_{\lambda\to\infty} \sigma_\varepsilon^2(\lambda) = \frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}(\lambda))^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}(\lambda))}{n - D_f(\lambda)} = \frac{\mathbf{y}^T\mathbf{y}}{n} = \text{var}(\mathbf{y}), \tag{2.48}$$

a measure that is known to be reasonably close to $\mathbf{r}_i^T\mathbf{r}_i/(n-p)$.

2.7 Variable Selection

In previous sections we have briefly mentioned the possibility of testing whether a single variable contributes significantly to the description of the response variable $\mathbf{y}$. This points to the fact that a subset of the variables included in a model may constitute a model that performs as well or better than the full model. In general, we prefer a more compact model over a redundant one for reasons relating to the principle of Occam's razor[6]. The principle is perhaps best paraphrased as "all things being equal, the simplest solution tends to be the best one". A more precise statistical motivation is that the error variance tends to be lower for a more parsimonious model than for a larger, redundant, model. Revisiting Equation 2.24 for the error variance,

$$\hat{\sigma}_\varepsilon^2 = \frac{\mathbf{r}^T\mathbf{r}}{n-p}, \tag{2.49}$$

we see that as p gets smaller, the error variance shrinks, assuming all models describe $\mathbf{y}$ equally well. This ultimately leads to a more powerful analysis with lower p-values for non-zero coefficients.

Another motivation of particular interest in this thesis is that smaller models are easier to interpret. A regression model is mainly used for two purposes; prediction and interpretation. A model solely used for prediction can be allowed to include a large number of variables, relevant or not, as long as the prediction accuracy is sufficiently high. If one also wishes to understand the factors governing the response, interpretation will become easier with fewer variables in the model. A model for disease progression that contains variables for body-mass index and age alone is easier to interpret and draw clinical conclusions from than a model with, say, ten variables. Furthermore, variable selection can be used to replace the standard statistical test of whether a regression coefficient is zero. Variable selection may have an edge over classic testing procedures as the latter places assumptions on the variables that do not necessarily hold. Instead, the variables included by the variable selection procedure may be reported as the important ones.

Out of a set of candidate models, we must define criteria for selecting the most appropriate model. In Section 2.6 we used one such criterion — prediction error measured on a validation data set which was separate from the training data set used to construct the various models. This is a simple criterion which works in cases where there are plenty of observations compared to the number of variables. A rule-of-thumb is that any data set (training, validation, test) should contain a number of observations that is at least 10–15 times the number

[6] The name and principle is due to the 14th-century English logician and Franciscan friar William of Ockham. The principle's Latin name *lex parsimoniae* is sometimes used.

of variables. This is not the case for most models; in fact, statistical analyses in e.g. image analysis frequently contain many more variables than observations. For such models, we cannot afford to split the data into separate training and test sets. Instead, one may attempt to estimate the prediction error directly from the training data. Many methods exist for this purpose. We will list some common ones here for reference, but refrain from explaining them in any detail. Refer to e.g. Hastie et al. [59] for an in-depth discussion of these and other criteria. The measures concern Gaussian models under squared-error loss.

Akaike's Information Criterion (AIC) [1]. The Akaike information criterion combines a measure of the training error (prediction error calculated from the training data set) with an estimate of the optimism of validating a model on training data,

$$AIC = \mathbf{r}^T\mathbf{r} + 2\sigma_\varepsilon^2 D_f, \tag{2.50}$$

where the training error $\mathbf{r}^T\mathbf{r}$ is adjusted by the optimism $2\sigma_\varepsilon^2 D_f$.

Mallows' C_p [96]. Mallows proposed a metric known as C_p with similar properties as the AIC,

$$C_p = \frac{\mathbf{r}^T\mathbf{r}}{\sigma_\varepsilon^2} - n + 2D_f. \tag{2.51}$$

This expression for C_p for a Gaussian model under squared-error loss can be seen to be a scaled and translated equivalent to AIC.

Bayesian Information Criterion (BIC). The Bayesian information criterion (also known as the Schwartz criterion [129]) takes a different approach to formulating an estimate of the prediction error, but arrives at an expression similar to the AIC.

$$BIC = \mathbf{r}^T\mathbf{r} + \log(n)\sigma_\varepsilon^2 D_f \tag{2.52}$$

Cross-validation. When the available data set is too small for creating separate training, validation and test sets, a measure of the prediction accuracy can nevertheless be established with a similar technique. The data set is split into equal parts, each containing roughly n/K observations where K is the number of subsets. The model is then fitted to $K-1$ of these subsets and the prediction accuracy is measured on the K^{th} set. Letting each of the subsets act as the validation set once, we obtain K measures of prediction accuracy which are then pooled, e.g. by taking their mean value, into a single estimate of prediction accuracy. Five or ten-fold cross-validation ($K = 5$ or $K = 10$) are common choices. With $K = n$, the procedure is known as leave-one-out cross validation, where a single observation is used to probe the prediction accuracy for each subdivision.

The AIC generally chooses better models when the true model is not among the candidates, while BIC often chooses the right model if the true model is present. The error variance σ_ε^2, if unknown, is commonly based on the largest model in question, usually the full OLS model. The measure of the number of parameters in the model D_f is simply p for OLS, but should be augmented for methods that include regularization. For ridge regression, an appropriate measure is given in Equation 2.37. For least angle regression and the LASSO, introduced in Sections 2.8 and 2.9, an unbiased estimate of D_f is the number of non-zero variables. Several different forms of the criteria listed above exist in the literature, but most are equal under scaling and translation.

In summary, the goal of variable selection is to select a subset of the available variables such that included variables contribute to the description of the response and such that no important variables are left out. The threshold defining which variables are interesting and which are redundant or irrelevant is tuned such that a suitably sparse model is obtained. In the following sections, several such methods will be presented. In the remainder of this section, we will outline two classic approaches.

2.7.1 All Subsets

All subsets regression is a particularly simple but computationally demanding approach to variable selection. As the name implies, all possible subsets of p variables are regarded. There are 2^p such combinations of variables, including the null model with zero variables. This follows since each variable takes on one out of two states — included or excluded, or *active/inactive* as we prefer to call them here. Since there are p variables, we have a total of 2^p combinations. For the diabetes data set, this is a feasible strategy as there are $2^{10} = 1024$ combinations to evaluate. However, assuming we accept to evaluate 10 000 models at most, the limit is a mere 13 variables (yielding 8192 models). For a full model of 64 variables, the number of combinations of variables is $1.8 \cdot 10^{19}$.

2.7.2 Stepwise Regression

A feasible strategy in cases with many variables is given by stepwise procedures. Variables are included in or excluded from the model according to some statistical criterion, and the variable to be included or excluded in a certain step depends on the set of currently active (included) variables. In this way, the set of solutions follow a path going from e.g. the null (empty) model to the full (OLS) model. *Forward selection* implements this case where a single variable is included in each step until the full model is reached. *Backward elimination* does the opposite; starting with the full model, variables are dropped one by one until the null model is reached. There are also algorithms that both include and exclude variables along the path. Here, we turn our attention to forward selection, as it bears close resemblance to the more recent methods of coming sections.

Following the theme of this thesis, we will present forward selection from a geometrical viewpoint. First, some notation is introduced. Variables are indexed from 1 to p. The set of such indices that are currently active (included in the model) is denoted $\mathcal{A}$, while the complement of this set contains the inactive variables and is denoted $\mathcal{I}$. A subset of the variables contained in the data matrix is denoted using a subscript set variable, e.g. $\mathbf{X}_{\mathcal{A}}$ for the currently active variables. The fitted response variable in each step is denoted $\boldsymbol{\mu}$, with $\boldsymbol{\mu} = \hat{\mathbf{y}}$ at the OLS solution. The residual, measured from the current position to the response variable, is denoted $\mathbf{r}_k = \mathbf{y} - \boldsymbol{\mu}_k$. The vector connecting the fitted variable in one step with the fitted variable in the next is denoted $\mathbf{d}_k = \boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k$. Examples of these variables are given in Figure 2.9, a case with two variables and three observations akin to Figure 2.4.

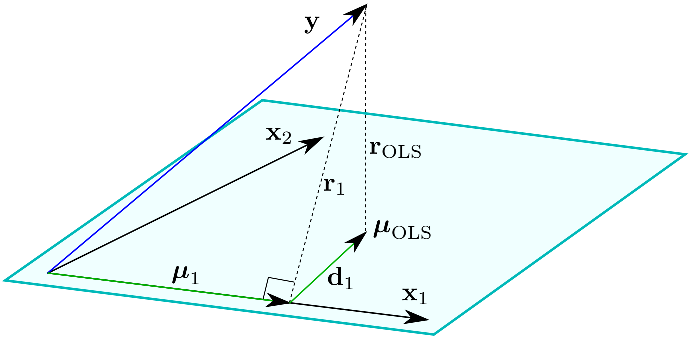


Figure 2.9: A small example of the forward selection procedure. The current position after one step of the algorithm is at $\boldsymbol{\mu}_1$, where $\mathbf{r}_1 = \mathbf{y} - \boldsymbol{\mu}_1$. The correlation between $\mathbf{r}_1$ and currently inactive variables is measured, resulting in the inclusion of $\mathbf{x}_2$ as this is the sole inactive variable. A step $\mathbf{d}_1$ taken from the current position to the full OLS solution concludes the process.

At each step of the algorithm, a variable is included according to some statistical criterion. There are a range of measures that are suitable here, including significance tests (t or F-tests), AIC, BIC and correlation, of which the latter is employed here. The correlation is measured between the current residual and each variable.

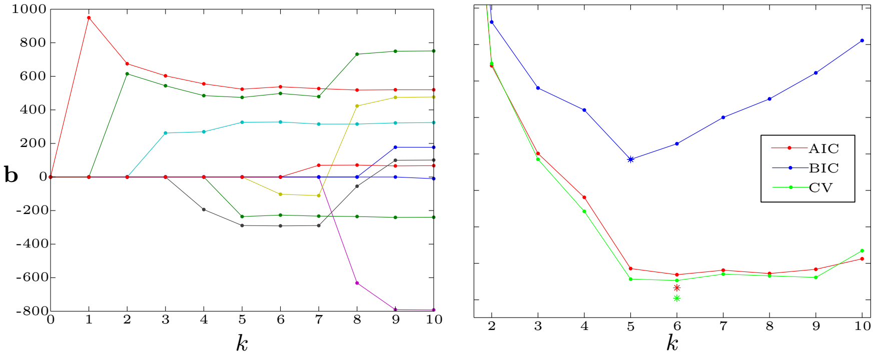
The algorithm starts at the origin representing the null model. The variable strongest correlated with $\mathbf{r}_1 = \mathbf{y}$ is the first variable to enter $\mathcal{A}$. A step is then taken in the direction of this variable such that the solution of the partial OLS regression of $\mathbf{y}$ onto the first variable is reached. The residual $\mathbf{r}_2$ is now measured from this point to $\mathbf{y}$. The process is then repeated. The correlation between $\mathbf{r}_k$ and inactive variables is measured, a second variable becomes active, and a new step is taken from the current position to the partial OLS solution including the two active variables. The process continues until the full OLS solution is reached. Algorithm 2.2 summarizes the procedure.

Algorithm 2.2 Stepwise Regression

```
Initialize the coefficient vector $\mathbf{b}_0 = \mathbf{0}_p$ and the fitted vector $\boldsymbol{\mu} = \mathbf{0}_n$,
Initialize the active set $\mathcal{A} = \emptyset$ and the inactive set $\mathcal{I} = \{1 \dots p\}$
for $k = 1$ to $p$ do
    Update residual $\mathbf{r} = \mathbf{y} - \boldsymbol{\mu}$
    Find maximal correlation $c = \max_{i \in \mathcal{I}} \mathbf{x}_i^T \mathbf{r}$
    Move variable corresponding to $c$ from $\mathcal{I}$ to $\mathcal{A}$.
    Calculate the partial OLS solution $\mathbf{b}_{\text{OLS}}^{\mathcal{A}} = (\mathbf{X}_{\mathcal{A}}^T \mathbf{X}_{\mathcal{A}})^{-1} \mathbf{X}_{\mathcal{A}}^T \mathbf{y}$
    Calculate current direction $\mathbf{d} = \mathbf{X}_{\mathcal{A}} \mathbf{b}_{\text{OLS}}^{\mathcal{A}} - \boldsymbol{\mu}$
    Save the regression coefficients $\mathbf{b}_k = \mathbf{b}_{\text{OLS}}^{\mathcal{A}}$
    Update the fitted vector $\boldsymbol{\mu} = \boldsymbol{\mu} + \mathbf{d}$
end for
Output the series of coefficients $\mathbf{B} = \{\mathbf{b}_0 \dots \mathbf{b}_p\}$
```

Figure 2.10 shows the results from applying forward selection to the diabetes data set. Figure 2.10(a) plots the values of the regression coefficients along the selection process from $k = 0$ to $k = p$ variables. The abscissa ("x-axis") runs in a direction of *less* regularization, opposite to the path of ridge regression. Although this type of variable selection is normally regarded as a discrete process, where variables are either included or excluded, we choose to plot the values of the coefficients as a trace, or regularization path. This provides a better basis for comparison with continuous methods such as ridge regression, and interesting solutions can indeed be found between steps. Figure 2.10(b) shows part of the corresponding error curves along the path using AIC (red), BIC (blue) and five-fold cross-validation (green). At the minimum of each curve, the minimal value of the corresponding all-subsets procedure is marked with a star. The BIC tends to pick a sparser model than AIC and cross-validation. Indeed, BIC retains five variables here, while AIC and cross-validation selects a model with six variables.

Forward selection is known to be very sensitive to small perturbations of the input variables. This is an effect of the greedy nature of the procedure. Once a variable has become active, all the information contained in the active variables is used to fit the response, frequently resulting in overfitting. Also, subtle combinations of variables are often overlooked. In the following sections, more careful variants of forward selection will be introduced which address such short-



(a) Regularization path of the forward selection procedure for variable selection.

(b) Estimates of prediction error along the path. The all-subsets minima are denoted by stars.

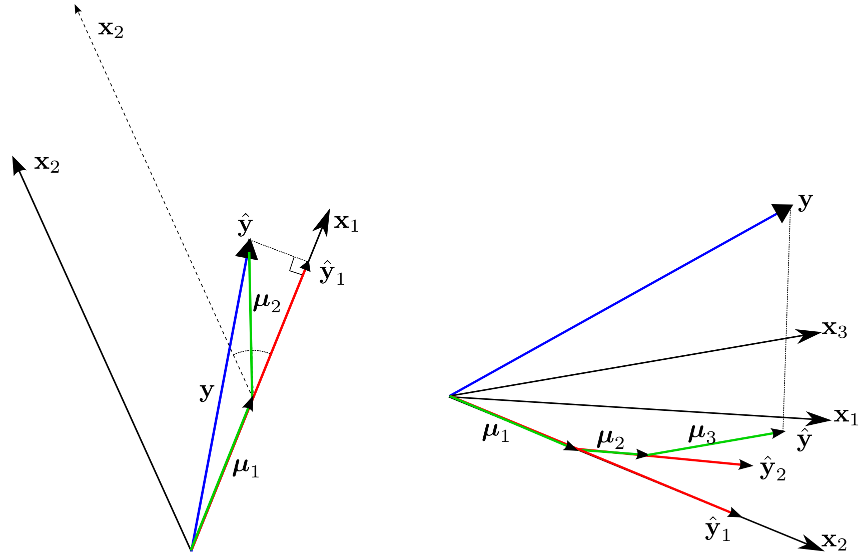
Figure 2.10: Results of applying forward selection to the diabetes data set. Although the solution set is usually considered discrete, the path is presented in a continuous fashion here. The models selected by each criterion contain 50-60 % of the total number of variables, creating a more manageable model which is easier to interpret.

comings.

2.8 Least Angle Regression

Least angle regression (LAR) is a geometrically motivated regression method that provides a gentler version of forward selection. Its implementation, described in detail below, requires some work while the conceptual outline of the algorithm is straightforward. As with forward selection, the algorithm proceeds from zero "active" variables and adds a single variable in each step until all variables are active and the full OLS solution is reached. In contrast to forward selection, variables are not either fully included or excluded in each step; instead they are gradually entering the model in a continuous fashion, producing a proper regularization path similar to that of ridge regression (cf. Figure 2.7(a)).

The name least angle regression describes the core idea of the algorithm. Starting with the empty set of active variables, the angle between each variable and the response is measured, and the variable with the smallest angle becomes the first variable included into the model. Walking along the direction of this variable, the angles between the variables and the residual vector are measured, where the residual is the vector from the current position along the walk to $\mathbf{y}$. Along this walk, the angles will change; in particular, the angle between the residual vector and the active variable will grow monotonically towards 90 degrees, a point where the partial OLS solution is reached as for forward selection. At some stage before this point, another variable will obtain the same angle with respect to the residual vector as the active variable. The walk is then halted and the new variable is added to the active set. The new direction of the walk is towards the partial least squares solution obtained through OLS regression of the response onto the two active variables. Again, before this walk reaches the partial OLS solution (where the two angles reaches 90 degrees), a new variable will obtain the same angle as the two active variables. This variable enters the model at this point and a new direction is calculated. After p steps, the full least squares solution will be reached ending the trace of the regularization path. A schematic overview of the algorithm is shown in Figure 2.11.



(a) Least angle regression of $\mathbf{y}$ (blue) onto two variables $\mathbf{x}_1$ and $\mathbf{x}_2$.

(b) Least angle regression of $\mathbf{y}$ (blue) onto three variables $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$.

Figure 2.11: Outline of the geometry of least angle regression with two and three variables. Starting at the origin, the fitted vector moves in a piecewise linear fashion along directions $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_p$ of minimal angle/maximal correlation between the residual vector and the variables. At each breakpoint, the next step is taken in the direction of the OLS solution $\hat{\mathbf{y}}_k$ using the currently active variables.

Figure 2.11(a) shows the process for $p = 2$ and $n = 3$, showing the variables $\mathbf{x}_1$ and $\mathbf{x}_2$ "from above", where the response vector $\mathbf{y}$, shown in blue, is sticking out of the page. At the start of the algorithm, the current position is at the origin and $\mathbf{r}_1 = \mathbf{y}$. The smallest angle is between $\mathbf{r}$ and $\mathbf{x}_1$ at this point; $\mathbf{x}_1$ therefore becomes the first active variable. Walking along $\mathbf{x}_1$, we reach a point

$\boldsymbol{\mu}_1$ where $\angle(\mathbf{x}_1, \mathbf{r}) = \angle(\mathbf{x}_2, \mathbf{r})$, where $\mathbf{r} = \mathbf{y} - \boldsymbol{\mu}_1$. At this point $\mathbf{x}_2$ enters the model. The next direction is towards the OLS projection of $\mathbf{y}$ onto $\mathbf{x}_1$ and $\mathbf{x}_2$. Since there are only two variables in this model, the new direction is towards the full OLS solution where the algorithm is terminated. The walk from $\boldsymbol{\mu}_1$ to the full OLS solution is denoted $\boldsymbol{\mu}_2$.

A schematic setup with three variables is shown in Figure 2.11(b). Variables are added in the order $\mathbf{x}_2$, $\mathbf{x}_1$ and $\mathbf{x}_3$. The variables $\hat{\mathbf{y}}_1$, and $\hat{\mathbf{y}}_2$ represent the partial OLS solutions on $\mathbf{x}_2$ and $\{\mathbf{x}_1, \mathbf{x}_2\}$ respectively, while $\hat{\mathbf{y}}$ represents the full OLS solution.

A number of questions arise from this description of the LAR algorithm. First, what is the rationale of measuring angles between variables? The cosine of the angle between two vectors $\mathbf{a}$ and $\mathbf{b}$ can be expressed using inner products as

$$\cos\angle(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^T\mathbf{b}}{\sqrt{\mathbf{a}^T\mathbf{a}}\sqrt{\mathbf{b}^T\mathbf{b}}}. \tag{2.53}$$

Treating $\mathbf{a}$ and $\mathbf{b}$ as variables rather than vectors, and assuming that $\mathbf{a}$ and $\mathbf{b}$ have been mean centered, the correlation between $\mathbf{a}$ and $\mathbf{b}$ is

$$\operatorname{corr}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^T\mathbf{b}}{\sqrt{\mathbf{a}^T\mathbf{a}}\sqrt{\mathbf{b}^T\mathbf{b}}}, \tag{2.54}$$

the exact same expression. Angles therefore have a direct correspondence to correlation, where small angles correspond to high correlation and vice versa. An equivalent name for LAR could therefore be ”strongest correlation regression”, where the current fitted vector always points in the direction of maximal correlation.

A more involved question concerns the directions calculated at each breakpoint. As a breakpoint is reached and a new variable enters the model, the angles between the response and each active variable are equal. The new direction is taken towards the least squares solution defined by $\mathbf{y}$ and the active variables. At this OLS solution, the angles are also equal, as the residual is at right angles with all active variables. The question is whether the angles remain equal as we travel from the breakpoint towards the partial OLS solution. Since the endpoints of this vector are equiangular, it is sufficient to show that the angles change linearly along the walk. Figure 2.12 shows a view of the problem with two active variables where the current position is at $\boldsymbol{\mu}_1$, $\mathbf{x}_2$ just entered the model, and we wish to estimate $\boldsymbol{\mu}_2$. The current residual vector is denoted $\mathbf{r}$. The partial OLS solution involving $\mathbf{x}_1$ and $\mathbf{x}_2$ is denoted $\boldsymbol{\mu}_{\text{OLS}}$. The walk along $\boldsymbol{\mu}_2$ towards $\boldsymbol{\mu}_{\text{OLS}}$ can be formulated $\gamma(\boldsymbol{\mu}_{\text{OLS}} - \boldsymbol{\mu}_1) + \boldsymbol{\mu}_1$ where $0 \leq \gamma \leq 1$; estimating $\boldsymbol{\mu}_2$ then amounts to estimating γ. Denoting the set of currently active variables $\mathcal{A}$ and assuming normalized variables, the correlation (or cosine of the angle) between the active variables and the residual vector as we walk along $\boldsymbol{\mu}_2$ is $\mathbf{x}_{i\in\mathcal{A}}^T(\mathbf{y}-\gamma(\boldsymbol{\mu}_{\text{OLS}}-\boldsymbol{\mu}_1)-\boldsymbol{\mu}_1)$. This is a linear function of γ, meaning that the angles between the residual and the variables in $\mathcal{A}$ will change, but remain equal along each direction.

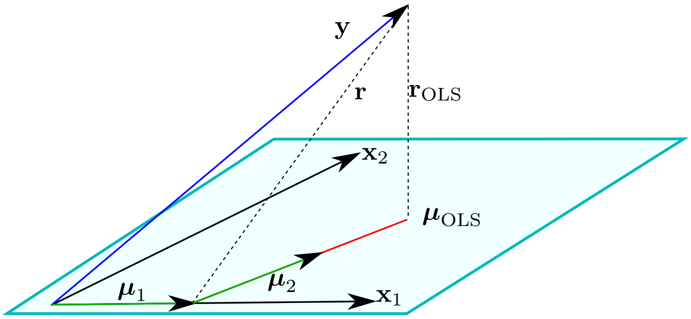


Figure 2.12: Least angle regression case showing the solution obtained after a single step of the algorithm. The second active variable $\mathbf{x}_2$ has just been included and the next direction will be towards the OLS solution $\boldsymbol{\mu}_{\text{OLS}}$ representing the regression of $\mathbf{y}$ onto $\mathbf{x}_1$ and $\mathbf{x}_2$. The residual vector at the current position is denoted $\mathbf{r}$ while $\mathbf{r}_{\text{OLS}}$ is the residual vector at the OLS solution.

We have now come a long way towards describing the entire algorithm formally. For some value of γ, a new variable will enter the set of active variables. This happens when the angles corresponding to the active variables (which are all equal) become equal to one the angles corresponding to an inactive variable. Denoting the set of inactive variables $\mathcal{I}$, we seek the smallest γ such that

$$\mathbf{x}_{i\in\mathcal{I}}^T(\mathbf{y}-\gamma(\boldsymbol{\mu}_{\text{OLS}}-\boldsymbol{\mu}_1)-\boldsymbol{\mu}_1)=\mathbf{x}_{j\in\mathcal{A}}^T(\mathbf{y}-\gamma(\boldsymbol{\mu}_{\text{OLS}}-\boldsymbol{\mu}_1)-\boldsymbol{\mu}_1). \tag{2.55}$$

Solving this expression for γ, we get

$$\gamma=\frac{(\mathbf{x}_i-\mathbf{x}_j)^T(\mathbf{y}-\boldsymbol{\mu}_1)}{(\mathbf{x}_i-\mathbf{x}_j)^T(\boldsymbol{\mu}_{\text{OLS}}-\boldsymbol{\mu}_1)}=\frac{(\mathbf{x}_i-\mathbf{x}_j)^T\mathbf{r}}{(\mathbf{x}_i-\mathbf{x}_j)^T\mathbf{d}} \tag{2.56}$$

where $\mathbf{d}=\boldsymbol{\mu}_{\text{OLS}}-\boldsymbol{\mu}_1$ is the direction of the walk. Now, $\mathbf{d}$ is the orthogonal projection of $\mathbf{r}$ onto the plane spanned by the variables in $\mathcal{A}$, therefore we have $\mathbf{x}_j^T\mathbf{r}=\mathbf{x}_j^T\mathbf{d}\equiv c$, representing the angle at the current breakpoint ($\boldsymbol{\mu}_1$). Furthermore, the sign of the correlation between variables is irrelevant in LAR. Therefore, we must also check where the correlations of opposite signs become equal to the correlation of the active variables. In terms of angles, we also have this sign distinction, as the angles are defined within the interval $[-90,90]$ degrees. Working through the derivation above for correlations of opposite signs, it is seen that the next active variable and the step length γ can be found by

$$\gamma=\min_{i\in\mathcal{I}}\left\{\frac{\mathbf{x}_i^T\mathbf{r}-c}{\mathbf{x}_i^T\mathbf{d}-c},\quad\frac{\mathbf{x}_i^T\mathbf{r}+c}{\mathbf{x}_i^T\mathbf{d}+c}\right\},\quad 0<\gamma\leq 1, \tag{2.57}$$

where the two terms are for correlations/angles of equal and opposite sign respectively.

Keeping track of the LAR regression coefficients at each breakpoint, the coefficients at the next step is given by γ,

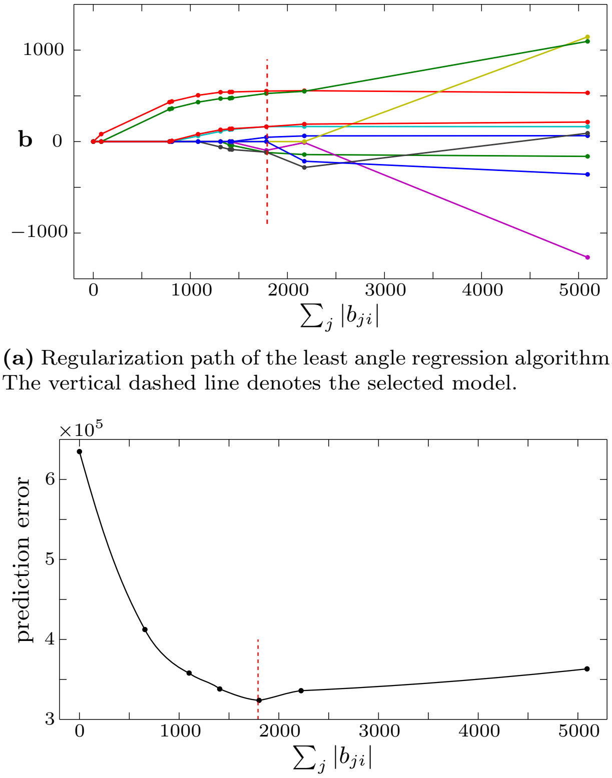
$$\mathbf{b}_k = \gamma(\mathbf{b}^{\mathcal{A}}_{\text{OLS}} - \mathbf{b}_{k-1}) + \mathbf{b}_{k-1} \tag{2.58}$$

Now that the key pieces of the LAR puzzle are in place, we state the entire LAR regression process in Algorithm 2.3.

Algorithm 2.3 Least Angle Regression

```
Initialize the coefficient vector $\mathbf{b}_0 = \mathbf{0}_p$ and the fitted vector $\boldsymbol{\mu} = \mathbf{0}_n$,
Initialize the active set $\mathcal{A} = \emptyset$ and the inactive set $\mathcal{I} = \{1 \dots p\}$
for $k = 1$ to $p - 1$ do
    Update residual $\mathbf{r} = \mathbf{y} - \boldsymbol{\mu}$
    Find maximal correlation $c = \max_{i \in \mathcal{I}} |\mathbf{x}_i^T \mathbf{r}|$
    Move variable corresponding to $c$ from $\mathcal{I}$ to $\mathcal{A}$.
    Calculate the partial OLS solution $\mathbf{b}^{\mathcal{A}}_{\text{OLS}} = (\mathbf{X}_{\mathcal{A}}^T \mathbf{X}_{\mathcal{A}})^{-1} \mathbf{X}_{\mathcal{A}}^T \mathbf{y}$
    Calculate current direction $\mathbf{d} = \mathbf{X}_{\mathcal{A}} \mathbf{b}^{\mathcal{A}}_{\text{OLS}} - \boldsymbol{\mu}$
    Calculate the step length $\gamma = \min_{i \in \mathcal{I}} \left\{ \frac{\mathbf{x}_i^T \mathbf{r} - c}{\mathbf{x}_i^T \mathbf{d} - c}, \frac{\mathbf{x}_i^T \mathbf{r} + c}{\mathbf{x}_i^T \mathbf{d} + c} \right\}$, $0 < \gamma \leq 1$
    Update regression coefficients $\mathbf{b}_k = \gamma(\mathbf{b}^{\mathcal{A}}_{\text{OLS}} - \mathbf{b}_{k-1}) + \mathbf{b}_{k-1}$
    Update the fitted vector $\boldsymbol{\mu} = \boldsymbol{\mu} + \gamma \mathbf{d}$
end for
Let $\mathbf{b}_p$ be the full OLS solution $\mathbf{b}_p = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$
Output the series of coefficients $\mathbf{B} = \{\mathbf{b}_0 \dots \mathbf{b}_p\}$
```

To further illustrate the method, we apply it to the diabetes data set. As before, the observations are divided into a training set with 221 observations, a validation set with 111 observations and a test set consisting of the last 110 observations. The LAR algorithm is applied to the training data producing 11 $(p + 1)$ coefficient vectors $\mathbf{b}$, including the all-zeros vector. Knowing that the coefficient paths are linear between breakpoints, we plot the resulting regularization path in Figure 2.13(a). The coefficient values are plotted against the quantity $\sum_j |b_{ji}|$, the sum of absolute coefficients at each breakpoint. The reason for this choice of abscissa will become clear in Section 2.9. The regularization path ranges from the all-zeros vector on the left, to the full OLS solution on the right. The advantage of LAR over ridge regression is that LAR implements both coefficient shrinkage and variable selection. As regularization is increased, coefficients do not merely approach zero as is the case for ridge regression. Instead, they are shrunk to exactly zero, in effect excluding them from the regression model. Figure 2.13(b) shows the prediction error of the LAR model calculated using the validation data set at 200 locations along the regularization path. At



(a) Regularization path of the least angle regression algorithm. The vertical dashed line denotes the selected model.

(b) Prediction error along the LAR path.

Figure 2.13: The piecewise linear nature of the LAR algorithm, and the associated prediction error curve. Curves range from the all-zeros ($\mathbf{b} = \mathbf{0}$) solution on the far left, to the OLS solution on the far right.

the point of minimal error, the model consists of approximately eight predictors instead of the ten predictors of the full model. The price one pays for a parsimonious model is usually an increase in prediction error, although this increase may be small enough to motivate the use of a smaller model. In this example, we have the unusual case where the achieved prediction error was lower for LAR than for ridge regression. The test error is $3.26 \cdot 10^5$, compared to the validation error $3.23 \cdot 10^5$.

2.9 The LASSO

The least absolute shrinkage and selection operator (LASSO) method is in definition closely related to ridge regression. The ordinary least squares problem is augmented to include a constraint on the regression coefficients $\mathbf{b}$. While ridge regression formulates this constraint as an upper bound on the squared length of $\mathbf{b}$, the LASSO measures the size of $\mathbf{b}$ by the sum of absolute coefficients b_i. The corresponding cost function is

$$\mathbf{b}_{\mathrm{LASSO}} = \arg\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \lambda\|\mathbf{b}\|_1, \tag{2.59}$$

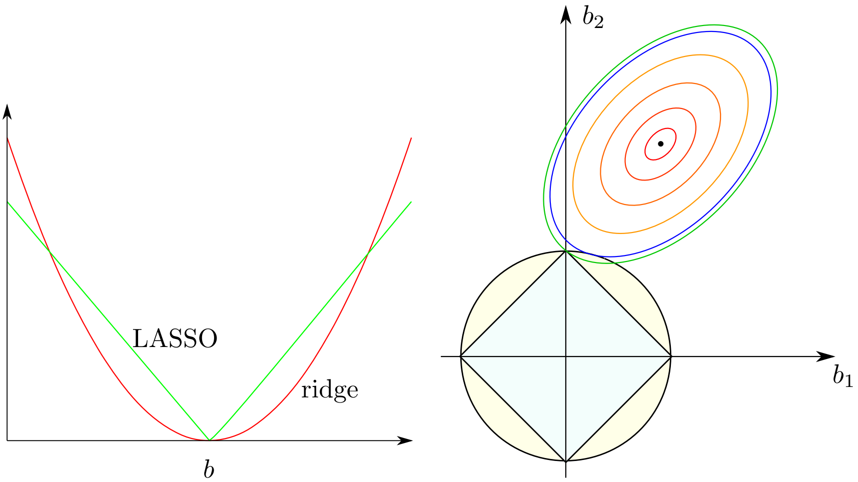
where $\|\mathbf{b}\|_1 = \sum_i |b_i|$ denotes the ℓ_1-norm. The difference between the LASSO and ridge regression is in other words due to the choice of penalty term; ridge regression uses the squared ℓ_2-norm whereas LASSO uses the ℓ_1-norm. This difference may seem insignificant as both implement forms of coefficient shrinkage leading to decreased variance of the estimates. The LASSO does, however, have another benefit. The form of the penalty function is such that coefficients are not merely shrunk *towards* zero, they are forced to *exactly* zero, one by one, as the amount of regularization is increased. Through this property, the LASSO implements both shrinkage and variable selection, leading to models which are easier to interpret.

We will provide an intuitive explanation of why the LASSO implements variable selection while ridge regression does not. Figure 2.14 shows the geometry of each regression problem. In Figure 2.14(a) the penalty function of the LASSO is shown in green and the ridge ditto is shown in red. For small values of b, the penalty gradually decreases for ridge regression and vanishes for b sufficiently small. The LASSO also penalizes variables with small magnitude less than larger variables, but uses a penalty that is directly proportional to the magnitude of b. Figure 2.14(b) provides another view of the LASSO and ridge penalty terms. Here, the OLS solution of a two-variable problem is marked with a black dot. Around this dot, concentric ellipses represent values of the loss function $\|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2$ for different choices of $\mathbf{b}$. The shape of these ellipses are determined by the variance-covariance structure of $\mathbf{X}$. Both ridge regression and the LASSO can be formulated in equivalent forms where the penalty function enters the equation as a separate constraint,

$$\|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 \quad \text{subject to} \quad \|\mathbf{b}\|^2 \leq t^2 \tag{2.60}$$

$$\|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 \quad \text{subject to} \quad \|\mathbf{b}\|_1 \leq t, \tag{2.61}$$

where each value of t corresponds to a unique value of λ in Equations 2.32 and 2.59. Using this formulation, we see that each penalty term defines a region in which we require the regression coefficients to reside. In Figure 2.14(b), these regions are shown for the ridge and LASSO penalties. The ridge penalty has the form of a circle, where t is the radius, whereas the LASSO penalty is diamond-shaped with ”radius” t at its corners. Solving the respective regression problem amounts to finding the values of $\mathbf{b}$ inside the region defined by the penalty function that lie closest to the OLS solution. The outer two ellipses of the loss function are tangent with the penalty regions of each method. The pointed geometry of the LASSO penalty makes it common for the ellipses to touch one of the corners rather than one of its sides. At the corners, one variable is zero. In higher dimensions, the ellipsoids of the loss function are more likely to hit either a corner, where a single variable is active, or a ridge connecting two corners, where one or several variables are inactive, than one of its faces. Ridge regression does not share this property, as the positions on the ridge penalty region corresponding to the corners of the LASSO region are no different from any other positions along the boundary. In Figure 2.14(b), the ridge solution is non-zero for both coordinates whereas the LASSO solution has $b_1 = 0$.



(a) The geometry of the LASSO penalty function (green) versus the ridge penalty (red) in one dimension. The LASSO constraint consistently penalizes values until they reach exactly zero, whereas the ridge regression penalty vanishes for small values of b.

(b) The geometry of the LASSO and ridge regression in two dimensions. The diamond represents the constraint region of the LASSO, whereas the light yellow region represents the ridge constraint. The OLS solution is marked with a black dot and values of the loss function are indicated using concentric ellipses.

Figure 2.14: Geometry of the LASSO versus ridge regression.

2.9.1 Computation

In Section 2.6, we derived the optimal ridge coefficients $\mathbf{b}_{\text{ridge}}$ by finding zero-crossings of the derivative of the cost function. However, the expression in Equation 2.59 is not differentiable since the derivative of the penalty function is undefined at $b = 0$. When the LASSO method [157] was presented, the estimates $\mathbf{b}_{\text{LASSO}}$ were calculated using numerical optimization procedures. Several algorithms exists for this purpose. One such approach is given here in brief.

The regression coefficients can be formulated $\mathbf{b} = \mathbf{b}^{+} - \mathbf{b}^{-}$, where $\mathbf{b}^{+}$ contains the positive elements of $\mathbf{b}$ (with zeros elsewhere) and $\mathbf{b}^{-}$ contains the positive representations of the negative elements. In this way, an $(n \times p)$ OLS problem can be restated as

$$\mathbf{y} = [\mathbf{X} \quad -\mathbf{X}] \begin{bmatrix} \mathbf{b}^{+} \\ \mathbf{b}^{-} \end{bmatrix} + \mathbf{r}, \tag{2.62}$$

a problem with n observations and $2p$ variables. The LASSO constraint can now be formulated using the $2p + 1$ inequalities $b_i^{+} \geq 0$, $b_i^{-} \geq 0$, $\forall i$ and $\sum_{i=1}^{p}(b_i^{+} + b_i^{-}) \leq t$. This is a quadratic optimization problem with linear constraints, which can be solved using iterative techniques for convex optimization [164].

The drawback of such procedures is that they provide a LASSO estimate for a single value of t, or correspondingly, λ. Typically, solutions for a range of values of t are calculated to provide a basis for model selection. Besides the large computational cost this implies, it may be difficult to select an appropriate number of values of t as well as its range.

The purpose of the development of least angle regression (cf. Section 2.8) was not only to devise a novel method for regression and variable selection, but also to shed light on the LASSO. Remarkably, it was shown that the LASSO and LAR, although conceptually different, produce very similar results, and that a LAR-type algorithm exists for computing the entire regularization path of the LASSO. Via a simple modification of the LAR algorithm, the LASSO solutions for all possible values of t can be obtained. The modification is due to the finding that the signs of the regression coefficients always agree with the signs of the current correlations $\mathbf{X}^T(\mathbf{y} - \boldsymbol{\mu})$ for the LASSO. This is not true for LAR, but the LAR algorithm can be modified to uphold this property.

The modification proceeds as follows. An important observation is that the current correlations do not change signs within a single LAR step. In fact, their signs are constant along the entire path as angles corresponding to active variables are equal (disregarding signs) and monotonically increase to 90 degrees at the end of the path where the OLS solution is reached. Studying the coefficient paths in Figure 2.13(a) we see that one coefficient changes sign within the final step towards the OLS solution. This represents a violation of the LASSO rule, as the coefficient and the corresponding correlation will take on different signs once b crosses zero. In the LASSO modified LAR algorithm, we check whether any coefficients cross zero within each step. If so, the step length is chosen such that the coefficient in question just reaches zero, and is subsequently excluded from the active set of variables. The step length at which each variable hits zero is easily found by setting the update expression for the coefficients $\mathbf{b}_{k+1} = \tilde{\gamma}(\mathbf{b}^{\mathcal{A}}_{\text{OLS}} - \mathbf{b}_k) + \mathbf{b}_k$ equal to zero and solving for $\tilde{\gamma}$,

$$\tilde{\gamma}_i = \frac{b_i}{b_i - b^{\mathcal{A}}_{i\ \text{OLS}}}, \forall i \in \mathcal{A}. \tag{2.63}$$

We denote these step lengths $\tilde{\gamma}$ to separate them from the LAR step lengths γ. If any $\tilde{\gamma}_i > 0$ is smaller than the next value of γ, a LASSO modification will occur in the next step. The step length is then adjusted and the relevant variable is excluded from $\mathcal{A}$. Algorithm 2.4 presents the resulting algorithm including the LASSO modification.

The proof of the agreement of signs between the coefficients and the current correlations, and that the modified LAR algorithm renders all possible LASSO solutions is given by Efron et al. [40]. The proof is rather extensive, with several lemmas leading up to the final theorem. Given the technical level of this thesis, we choose to avoid discussing the validity of the modification here, and simply present the resulting algorithm.

Figure 2.15(a) shows the resulting LASSO regularization path. As mentioned above, the LASSO condition occurs only once, during the last LAR step. This leads to the exclusion of a variable, which is included again at the end of the following step. In the iteration after, the OLS solution is reached. Figure 2.15(b) shows the C_p measures of prediction error along the path. Using this heuristic, a model with 7 variables is selected.

2.10 Elastic Net Regression

Ridge regression has the benefit of handling cases where there are more variables p than observations n, and is known to produce models with good prediction error. Procedures such as the LASSO implements variable selection and efficient algorithms exists for obtaining the entire set of possible solutions, but does not handle cases where $p > n$. The Elastic Net attempts to combine the strengths of these methods while eliminating their respective shortcomings. The setup is straightforward. The penalty terms of ridge regression and the LASSO are

Algorithm 2.4 The modified LAR algorithm for computation of the LASSO

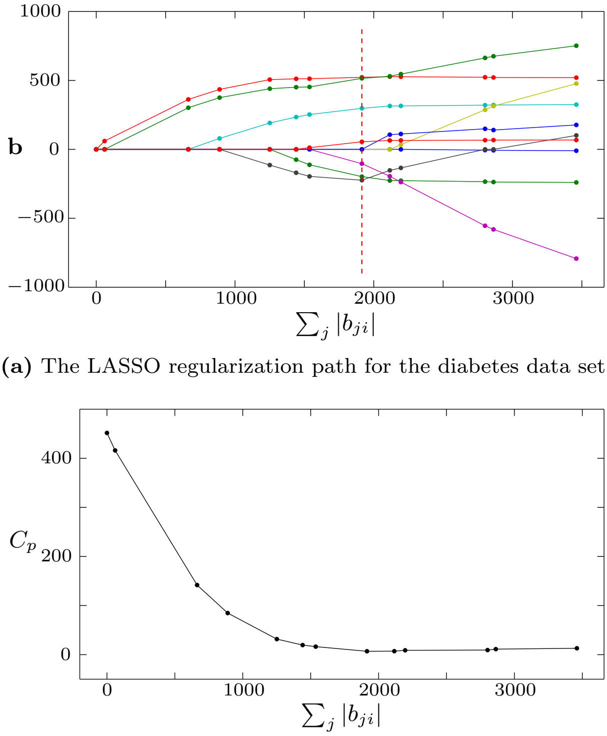
```
Initialize the coefficient vector $\mathbf{b}_0 = \mathbf{0}_p$ the fitted vector $\boldsymbol{\mu} = \mathbf{0}_n$, and an iteration counter $k = 0$.
Initialize the active set $\mathcal{A} = \emptyset$ and the inactive set $\mathcal{I} = \{1 \dots p\}$
while $|\mathcal{I}| > 0$ do
  Update residual $\mathbf{r} = \mathbf{y} - \boldsymbol{\mu}$
  Find maximal correlation $c = \max_{i \in \mathcal{I}} |\mathbf{x}_i^T \mathbf{r}|$
  if drop condition then
    Set drop condition to FALSE.
  else
    Move variable corresponding to $c$ from $\mathcal{I}$ to $\mathcal{A}$.
  end if
  Calculate the partial OLS solution $\mathbf{b}_{\text{OLS}}^{\mathcal{A}} = (\mathbf{X}_{\mathcal{A}}^T \mathbf{X}_{\mathcal{A}})^{-1} \mathbf{X}_{\mathcal{A}}^T \mathbf{y}$
  Calculate current direction $\mathbf{d} = \mathbf{X}_{\mathcal{A}} \mathbf{b}_{\text{OLS}}^{\mathcal{A}} - \boldsymbol{\mu}$
  Calculate drop condition step length $\tilde{\gamma} = \min_{i \in \mathcal{A}} b_{ik} / (b_{ik} - b_{i\,\text{OLS}}^{\mathcal{A}})$, $0 < \tilde{\gamma} < 1$
  if $|\mathcal{I}| = 0$ then
    Let the LAR step length $\gamma = 1$ (go all the way to the full OLS solution)
  else
    Calculate LAR step length $\gamma = \min_{i \in \mathcal{I}} \left\{ \frac{\mathbf{x}_i^T \mathbf{r} - c}{\mathbf{x}_i^T \mathbf{d} - c}, \quad \frac{\mathbf{x}_i^T \mathbf{r} + c}{\mathbf{x}_i^T \mathbf{d} + c} \right\}$, $0 < \gamma \leq 1$
  end if
  if $\tilde{\gamma} < \gamma$ then
    $\gamma = \tilde{\gamma}$
    Set drop condition to TRUE.
  end if
  Update regression coefficients $\mathbf{b}_{k+1} = \gamma(\mathbf{b}_{\text{OLS}}^{\mathcal{A}} - \mathbf{b}_k) + \mathbf{b}_k$
  Update the fitted vector $\boldsymbol{\mu} = \boldsymbol{\mu} + \gamma \mathbf{d}$
  if drop condition then
    Move variable corresponding to $\tilde{\gamma}$ from $\mathcal{A}$ to $\mathcal{I}$.
  end if
  $k = k + 1$
end while
Output the series of coefficients $\mathbf{B} = \{\mathbf{b}_0 \dots \mathbf{b}_k\}$
```

simply combined, yielding the following cost function,

$$\mathbf{b}_{\text{naive}} = \arg\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \lambda \|\mathbf{b}\|^2 + \delta \|\mathbf{b}\|_1, \tag{2.64}$$

where the optimal coefficients $\mathbf{b}_{\text{naive}}$ represent the solution for a particular choice of λ and δ. The name naive will be described below.

The geometric interpretation of the (combined) penalty term is shown in Figure 2.16. The constraint region defined by the Elastic Net consists of a combina-



(a) The LASSO regularization path for the diabetes data set.

(b) Prediction error measured by the C_p estimate along the path.

Figure 2.15: LASSO results for the diabetes data set, using all 442 observations as training data. The C_p measure estimates the prediction error as if tested on an independent validation set.

tion of the disc of ridge regression and the diamond of the LASSO. The relation between λ and δ determines the geometry of the region. Letting $\delta = 0$ results in the ridge solution, while $\lambda = 0$ leads to a pure LASSO procedure. The Elastic Net region region shown in red in Figure 2.16 has roughly $\delta = \lambda$. For $\delta > 0$, the region will be singular at each corner along the axes of $\mathbf{b}$, thus producing sparse solutions similarly to the LASSO.

Computation of the Elastic Net estimates is simple, given the path algorithm for the LASSO (cf. Algorithm 2.4). First, we review an alternative formulation of ridge regression. Instead of the loss+penalty formulation of Equation 2.32, ridge regression can be solved using OLS regression on an augmented data matrix and

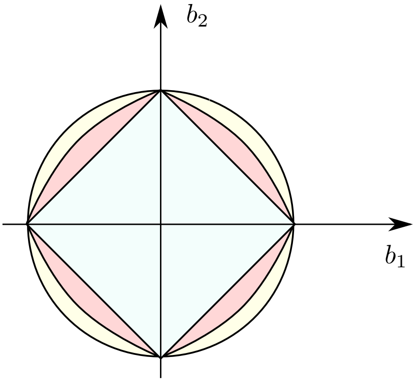


Figure 2.16: The regions defined by the constraints in ridge regression (yellow), the LASSO (blue), and the Elastic Net (red). The Elastic Net region is made up of some combination of the other two.

response vector,

$$\mathbf{b}_{\text{ridge}} = (\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^T\tilde{\mathbf{y}} \quad \text{where} \quad \tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \sqrt{\lambda}\mathbf{I}_p \end{bmatrix}, \quad \tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_p \end{bmatrix}. \tag{2.65}$$

This provides another explanation why ridge regression is able to provide an answer also in cases where $p > n$. The original $(n \times p)$ data matrix is expanded through the inclusion of p synthetic observations yielding the $((n+p) \times p)$ augmented matrix $\tilde{\mathbf{X}}$. Obviously, this matrix has more rows ("observations") than columns (variables), and the corresponding gram matrix can thus be inverted.

Using this technique, the Elastic Net can be formulated as a LASSO problem on augmented matrices,

$$\arg\min_{\mathbf{b}} \|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\mathbf{b}\|^2 + \delta\|\mathbf{b}\|_1, \tag{2.66}$$

For a fixed value of λ, the path corresponding to all relevant choices of δ can now be obtained using the LASSO algorithm. This does, however, mean that a suitable value of λ must be determined beforehand. Figure 2.17 shows results on the diabetes data set for three such values of λ, $\lambda = 0.001$, $\lambda = 0.1$ and $\lambda = 1000$. The first choice represents a lightly regularized version of the LASSO, handling cases where $p > n$. As $n > p$ in this case, the result is close to that of the pure LASSO. The second choice of λ shows a more strongly regularized model, where coefficients tend to follow simpler paths from their point of entry. This effect is taken to an extreme for $\lambda = 1000$. Each plot has an associated prediction error plot, estimated using 15-fold cross-validation. While the two first choices of λ produce seemingly useful models, the model corresponding to $\lambda = 1000$ perform only slightly better than the null model at its optimal value of δ (or

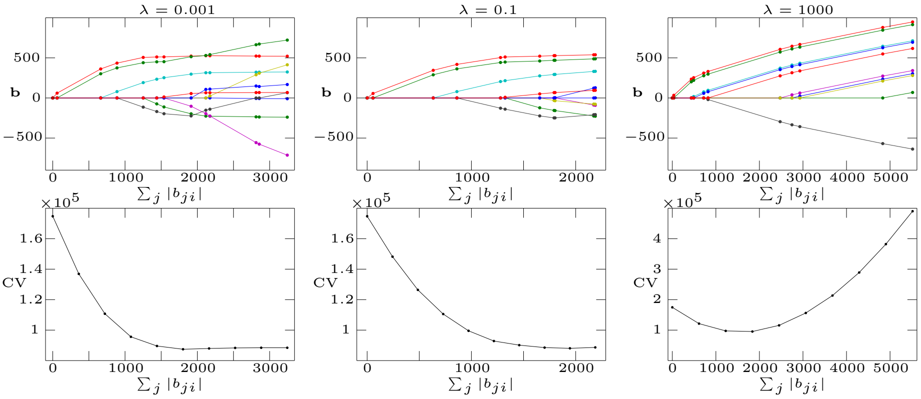


Figure 2.17: Results from using the Elastic Net on the diabetes data set for different values of the ridge-type regularization parameter λ. The top row shows the resulting coefficient paths, while the bottom row contains the corresponding 15-fold cross-validation error curves.

its corresponding value of $t = \sum_j |b_{ji}|$) — clearly this model is too rigid to be practical.

The inventors of the Elastic Net call the solution to Equation 2.64 the *naive* Elastic Net. In this form, the model has poor performance with regards to prediction, and is mainly useful for selecting a proper set of interesting variables. In [175], it is argued that the naive Elastic Net incurs a double amount of shrinkage. This statement is intuitively plausible; for a certain value of λ, we have the ridge regression solution at $\delta = 0$. For increasing values of δ, this already shrunk estimate is further shrunk, while uninteresting variables are excluded one by one. We wish to keep the results from variable selection, but adjust the coefficients for the shrinkage either incurred by the ridge penalty or the LASSO penalty. By multiplying the naive estimates by $1 + \lambda$, the ridge shrinkage is counteracted. More specifically,

$$\mathbf{b}_{\text{elastic}} = (1 + \lambda)\mathbf{b}_{\text{naive}}. \tag{2.67}$$

All coefficients in the paths of Figure 2.17 have been adjusted accordingly. In [175], more details are given regarding the appropriateness of this transformation, and several examples on different data sets show that the resulting prediction performance is comparable, if not better, than that of ridge regression and the LASSO.

2.11 The Non-negative Garrote

According to the English dictionary, a garrote is a method of capital punishment of Spanish origin in which an iron collar is tightened around a condemned person's neck until death occurs by strangulation or by injury to the spinal column at the base of the brain. This is (perhaps) reminiscent of the way regression coefficients are forced to exactly zero in this statistical namesake, the non-negative garrote. The idea of the Garrote is to take an initial estimate of the regression coefficients $\mathbf{b}^{\text{init}}$ and shrink these towards zero. The shrinkage concept is equivalent to previous methods in this chapter; the difference is that the estimation is guided by a fixed set of target coefficients. The most common choice of the initial (target) estimate $\mathbf{b}^{\text{init}}$ consists of the standard OLS coefficients. Let $\mathbf{Z} = [\mathbf{x}_1 b_1^{\text{init}} \dots \mathbf{x}_p b_p^{\text{init}}]$ be the matrix of component-wise contributions to the fitted vector $\hat{\mathbf{y}}$, such that $\hat{\mathbf{y}} = \mathbf{X}\mathbf{b}^{\text{init}} = \mathbf{Z}\mathbf{1}_p$. The non-negative garrote is then,

$$\arg\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{y} - \mathbf{Z}\mathbf{s}\|^2 + \lambda \sum_{i=1}^{p} s_i \qquad \text{subject to} \quad s_i \geq 0, \forall i. \tag{2.68}$$

The result is a set of shrinkage coefficients $\mathbf{s}$ that determine the extent to which each component of $\hat{\mathbf{y}}$, or equivalently, each regression coefficient b_i^{init}, is shrunk towards zero. Once the optimal shrinkage coefficients are obtained, the corresponding regression coefficients $\mathbf{b}_{\text{garrote}}$ are given by $b_i^{\text{garrote}} = s_i b_i^{\text{init}}, \forall i$.

For values of λ sufficiently large, the Garrote produces sparse solutions similar to LAR, the LASSO and the Elastic Net. Part of the explanation for this behavior is given by the solution to Equation 2.68 in the case of orthogonal (centered, unit length) predictors, in which case minimization can be carried out separately for each variable,

$$\begin{aligned} \frac{\partial}{\partial s_j}\left[\frac{1}{2}\|\mathbf{y} - \mathbf{x}_j b_j s_j\|^2 + \lambda s_j\right] = 0 \Leftrightarrow \\ -\mathbf{y}^T \mathbf{x}_j b_j + d_j b_j^2 \mathbf{x}_j^T \mathbf{x}_j + \lambda = 0 \Leftrightarrow \\ -b_j^2 + d_j b_j^2 + \lambda = 0 \Leftrightarrow \\ d_j = \max\left(0, 1 - \frac{\lambda}{b_j^2}\right). \end{aligned} \tag{2.69}$$

If the (initial) estimate b_j is large, the shrinkage coefficient will be close to 1, whereas it will be clamped to zero if b_j is small.

For $\lambda = 0$, the shrinkage constraint is not active. If the initial estimate consists of the OLS coefficients $\mathbf{b}_{\text{OLS}}$, the optimal $\mathbf{s}$ is the length-p vector of ones, in which case the OLS solution is returned. For other choices of $\mathbf{b}^{\text{init}}$, what is the setup in Equation 2.68 with $\lambda = 0$ solving? The shrinkage coefficients may take on any positive value in this case, yielding arbitrary values of the resulting estimators $\mathbf{b}_{\text{garrote}}$ but forcing them to be consistent in sign with $\mathbf{b}^{\text{init}}$. Further, $b_i^{\text{garrote}} = 0$ whenever $b_i^{\text{init}} = 0$. Both the sparsity and sign patterns of $\mathbf{b}^{\text{init}}$ are in other words preserved. However, it is important to note that $\mathbf{b}^{\text{init}}$ will not be returned for $\lambda = 0$ unless $\mathbf{b}^{\text{init}} = \mathbf{b}_{\text{OLS}}$. Instead, the solution is likely to be closer to $\mathbf{b}_{\text{OLS}}$ than to the initial estimates, but being restricted to follow the structure of $\mathbf{b}^{\text{init}}$. As λ grows, signs are preserved, while the solutions become increasingly sparse.

The Garrote has been shown to have desirable properties related to *consistency*. In general, a consistent estimator, for instance a vector of regression coefficients, is consistent if, with probability tending to 1, the estimates become equal to the true (unknown) parameters as the number of observations approaches infinity. In other words, a consistent estimator is sure to be close to the correct parameters if the underlying assumptions are met and we have enough observations. A regularized method is called *path consistent* if the regularization path contains the correct parameters as $n \to \infty$. For path algorithms that implement variable selection, such as the Garrote, we also require that the correct set of non-zero variables is identified at the position of the correct parameters. The Garrote is path consistent if the initial estimate is consistent. In contrast, LAR is not path consistent except under rather particular circumstances. Similar results exists for the LASSO and the Elastic Net. This does not, however, imply that these methods give poor estimates. While consistency apply in cases with $n \to \infty$, it is possible for e.g. the LASSO to achieve better results than the Garrote in cases with a limited number of observations.

At this point it should come as no surprise that a path algorithm exits for computing the entire solution set of the non-negative garrote. While other algorithms calculate the path of the regression coefficients directly, the Garrote establishes the path of the shrinkage coefficients $\mathbf{s}$. The procedure closely parallels the corresponding algorithm for the LASSO, where variables may both enter and leave the set of active variables. For the LASSO, a variable b_i leaves $\mathcal{A}$ if it becomes zero between two breakpoints. Similarly, the Garrote excludes a shrinkage coefficient s_i if it reaches 0 between breakpoints, as we require $s_i > 0, \forall i$ according to Equation 2.68. Using the OLS coefficients as the initial estimates, the algorithm terminates when the inactive set $\mathcal{I}$ becomes empty and all s_i reach 1. For other initial estimates, the algorithm is terminated when there is no feasible step length γ such that $0 < \gamma < 1$. In this case, γ is set to 1, a final step is taken before the procedure finishes. Algorithm 2.5 gives pseudo-code for the Garrote path algorithm.

Figure 2.18 shows results of using the Garrote on the diabetes data set using the OLS coefficients as initial estimates. Figure 2.18(a) shows the resulting

Algorithm 2.5 Path Algorithm for Breiman's Non-negative Garrote

```
Let $\mathbf{Z} = [b_1\mathbf{x}_1 \dots b_p\mathbf{z}_p]$ where $\mathbf{b}^{\mathrm{init}} = [b_1 \dots b_p]^T$.
Initialize the vector of shrinkage factors $\mathbf{s}_0 = \mathbf{0}_p$, the fitted vector $\boldsymbol{\mu} = \mathbf{0}_n$, and an iteration counter $k = 0$.
Initialize the active set $\mathcal{A} = \emptyset$ and the inactive set $\mathcal{I} = \{1 \dots p\}$
while $\gamma \neq 1$ do
  Update residual $\mathbf{r} = \mathbf{y} - \boldsymbol{\mu}$
  Find maximal covariance $c = \max_{i\in\mathcal{I}} \mathbf{z}_i^T\mathbf{r}$
  if drop condition then
    Set drop condition to FALSE.
  else
    Move variable corresponding to $c$ from $\mathcal{I}$ to $\mathcal{A}$.
  end if
  Calculate the partial OLS solution $\mathbf{s}^{\mathcal{A}}_{\mathrm{OLS}} = (\mathbf{Z}_{\mathcal{A}}^T\mathbf{Z}_{\mathcal{A}})^{-1}\mathbf{Z}_{\mathcal{A}}^T\mathbf{y}$
  Calculate current direction $\mathbf{d} = \mathbf{Z}_{\mathcal{A}}\mathbf{s}^{\mathcal{A}}_{\mathrm{OLS}} - \boldsymbol{\mu}$
  Calculate drop condition step length $\tilde{\gamma} = \min_{i\in\mathcal{A}} s_{ik}/(s_{ik} - s^{\mathcal{A}}_{i\,\mathrm{OLS}})$, $0 < \tilde{\gamma} < 1$
  Calculate step length $\gamma = \min_{i\in\mathcal{I}} \frac{\mathbf{x}_i^T\mathbf{r}-c}{\mathbf{x}_i^T\mathbf{d}-c}$, $0 < \gamma \leq 1$. If no such value of $\gamma$ exists, set $\gamma = 1$.
  if $\tilde{\gamma} < \gamma$ then
    $\gamma = \tilde{\gamma}$
    Set drop condition to TRUE.
  end if
  Update shrinkage factors $\mathbf{s}_{k+1} = \gamma(\mathbf{s}^{\mathcal{A}}_{\mathrm{OLS}} - \mathbf{s}_k) + \mathbf{s}_k$
  Update the fitted vector $\boldsymbol{\mu} = \boldsymbol{\mu} + \gamma\mathbf{d}$
  if drop condition then
    Move variable corresponding to $\tilde{\gamma}$ from $\mathcal{A}$ to $\mathcal{I}$.
  end if
  $k = k + 1$
end while
Output the series of shrinkage factors $\mathbf{S} = \{\mathbf{s}_0 \dots \mathbf{s}_k\}$
```

non-negative trace or the shrinkage coefficients. Individual s_i may be larger than 1 before reaching the OLS solution where $\mathbf{s} = \mathbf{1}_p$. The quantity $\sum_i s_i$ is monotonically increasing along the path. Once this path is obtained, the corresponding path for the Garrote regression coefficients is easily obtained by weighting the initial estimates using the shrinkage coefficients. Figure 2.18(b) shows the resulting trace. Figure 2.18(c) presents a plot of the C_p measure of prediction error. A model with eight non-zero variables is selected using this method. The calculation of C_p requires a measure of the number of parameters of a particular solution along the Garrote path. One such measure is given in

[170] and again in [169],

$$D_f = 2 \sum_{i=1}^{p} \left[I(s_i > 0) - s_i \right], \tag{2.70}$$

where $I(\cdot)$ is 1 if its argument is true and 0 otherwise.

2.12 References

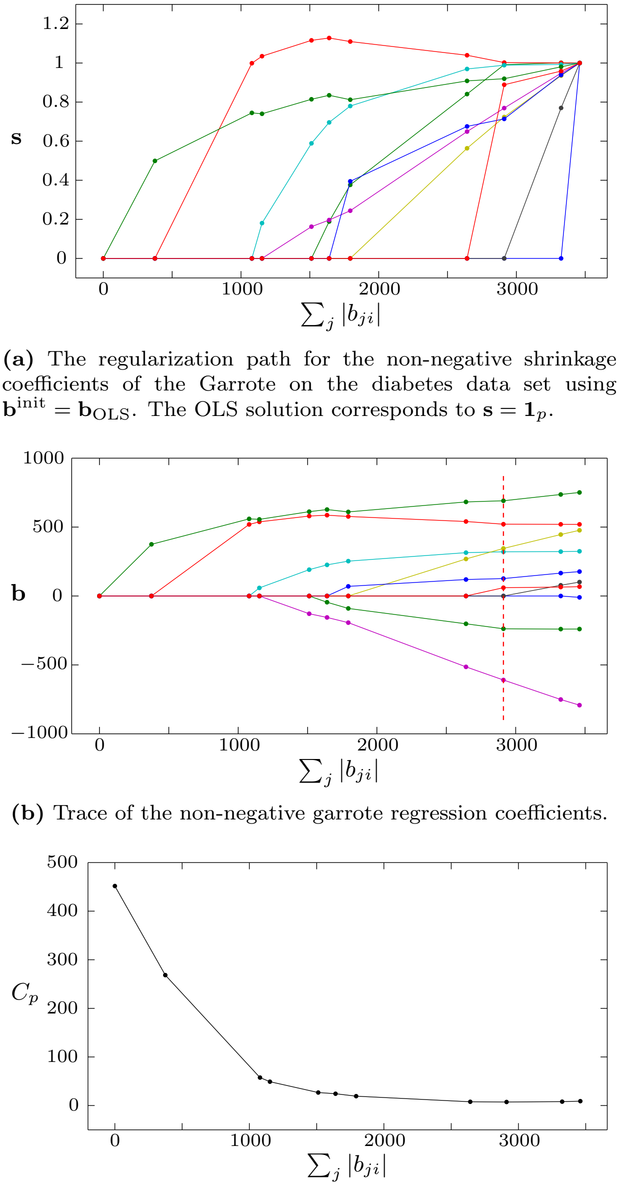
Discussion and explanation of the linear model and its variants can be found in most textbooks on multiple regression analysis. An old but useful reference is Cohen and Cohen [21]. Many such books also provide information on the bias-variance decomposition and the Gauss-Markov theorem. These properties are for instance carefully explained by Hastie et al. [59].

Pointwise regression is also a commonplace technique and a natural choice of analysis for high-dimensional problems where $p \gg n$. Such problems frequently arise in image analysis where there are typically more voxels/pixels (variables) than images (observations). Karl Friston and members of the Wellcome Department of Imaging Neuroscience in London have been instrumental in developing techniques for analysis and inference in such problems. Friston et al. [46] provides a good starting point.

Ridge regression, the earliest of the continuously regularized methods presented in this chapter, was developed and presented concurrently by Hoerl and Kennard [65] and Marquardt [97]. Its efficient computation using matrix decomposition has appeared in several papers. Hastie and Tibshirani [58] provide an overview of statistical techniques with quadratic regularization with such computational advantages, including ridge regression. The passage on the relation between strongly regularized ridge regression and pointwise regression is previously unpublished.

The presentation of stepwise procedures such as forward selection is included here since it provides a basis for Least Angle Regression and related methods. Several authors do, however, argue against its use. Babyak [4] provides a particularly pleasant discussion.

Least Angle Regression (LAR) was developed by Efron et al. [40]. In their monumental paper, they propose the LAR method, develop an efficient algorithm, investigate its relation to the LASSO and to stagewise regression (cf. [59], Algorithm 10.4), and provide modifications to the LAR algorithm such that the entire regularization path of the LASSO and stagewise regression can be obtained at low computational cost, all backed up by careful theoretical justification.



(a) The regularization path for the non-negative shrinkage coefficients of the Garrote on the diabetes data set using $\mathbf{b}^{\text{init}} = \mathbf{b}_{\text{OLS}}$. The OLS solution corresponds to $\mathbf{s} = \mathbf{1}_p$.

(b) Trace of the non-negative garrote regression coefficients.

(c) Prediction error measured by the C_p estimate along the path.

Figure 2.18: Results for the non-negative garrote on the (entire) diabetes data set. Using C_p, a model with 8 non-zero coefficients was selected.

The LASSO regression procedure was suggested by Tibshirani [157]. It became known for its consistency in selecting a reasonable set of important variables, but was hampered by its complex computation. The LAR paper [40] changed this, such that the entire LASSO path could be obtained at roughly the same computational cost required to solve a single OLS problem. Preceding work by Osborne et al. [109] presents a similar, but less direct, algorithm for the LASSO. The estimate of the number of degrees of freedom for the LASSO mentioned in Section 2.7 was proposed by Efron et al. [40], and further discussed and consolidated by Zou et al. [176]. Suggestions for extensions of the LASSO have been proposed by e.g. Tibshirani et al. [158] and Zou [174].

The Elastic Net regression method was proposed by Zou and Hastie [175].

The non-negative garrote was proposed by Breiman [16], and represents a predecessor of the LASSO. Yuan and Lin [170] develop an extension of this and similar algorithms for handling of categorial variables such as gender or car make. They also develop the estimate for the degrees of freedom of the Garrote, given here in Equation 2.70. In a follow-up paper [169], the basis for Algorithm 2.5 is described in detail, along with an investigation into the consistency of the Garrote and related methods.

The list of regularized statistical methods presented in this chapter is far from exhaustive. Other methods have been presented by George and McCulloch [49], Fu [47], Ojelund et al. [105], George and Foster [48], Fan and Li [42], and Shen and Ye [131]. Rosset and Zhu [122] discuss sufficient conditions such that a regularized method in the loss+penalty form has a piecewise linear solution path, and present examples.

Chapter 3

Classification

Classification is the act of separating observations into groups according to characteristics of the population. One example is the separation of people into males and females according to their length, weight and hand size. In linear regression, we estimate a regression equation, forming a hyperplane that models the response variable as a function of the predictors. In linear classification, we seek the hyperplane(s) that discriminates best between groups. A classifier is built from a training data set consisting of the measured data $\mathbf{X}$ of size $(n \times p)$ with n being the number of data points and p being the number of variables[1], and a corresponding vector of labels $\mathbf{y}$, defining class belonging for each observation. The labels must be distinct, but are otherwise arbitrary; although choosing -1 and +1 as labels in a two-class problem frequently simplifies the mathematical modus operandi. We will limit the exposition in this chapter to linear classification with two classes. Problems with two or more classes are usually handled by establishing a discriminant function for each class separately or between each pair of classes, and subsequently picking the class that most prominently separates from the others, given an unclassified input data point $\mathbf{x}$.

In the first section of this chapter, we will review classic ways of discriminating between groups, dating back to work by Fisher [44]. The discussion follows the one given by Hastie et al. [59] closely, both in derivation and notation.

[1] In classification, the variables are sometimes known as *traits*, or *characters*.

3.1 Linear and Quadratic Discriminant Analysis

One way of discriminating between classes is to estimate the quantity

$$P(G = k|X = \mathbf{x}). \tag{3.1}$$

Given an unclassified observation $\mathbf{x}$, this measures the probability that $\mathbf{x}$ belongs to class k. Here, G is a random variable denoting group belonging, and we assume that there are K classes in total. Point indices that are associated with a group i are collected in the set $\mathcal{G}_i$. Bayes theorem states that

$$P(G = k|X = \mathbf{x}) = \frac{P(X = \mathbf{x}|G = k)P(G = k)}{P(X = \mathbf{x})}. \tag{3.2}$$

To classify $\mathbf{x}$, the probability in relation to each class is calculated and the most probable class is picked. The denominator in Equation 3.2 will be the same in each of these calculations and can therefore be omitted. The probability of observing $\mathbf{x}$ given a class label k is denoted $P(X = \mathbf{x}|G = k)$ and is called the class-conditional probability function, while the a priori probability of a specific class is denoted $P(G = k) \equiv \pi_k$. In linear and quadratic discriminant analysis, the class-conditional probability functions are assumed to be continuous and Gaussian with a separate mean and variance-covariance matrix for each class. In this form, each such function is a *probability density function* which we denote $f_k(\mathbf{x})$. This yields $P(G = k|X = \mathbf{x}) \propto f_k(\mathbf{x})\pi_k$. The Gaussian density $f_k(\mathbf{x})$ has the form

$$f_k(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}_k|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)\mathbf{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)^T\right], \tag{3.3}$$

where $\mathbf{\Sigma}_k$ is the variance-covariance matrix of class k and $\boldsymbol{\mu}_k$ is the ditto mean, or *centroid.*

Since we are considering two classes $G = k$ and $G = l$ at a time, a suitable decision function can be defined by the ratio $P(G = k|X = \mathbf{x})/P(G = l|X = \mathbf{x})$. This function will be 1 for equal probabilities, larger than 1 if class k is more probable and less than 1 if class l is more probable. To arrive at a decision function that is zero for equal probabilities, we take the logarithm of this ratio. As seen in the following derivation, this also has the benefit of simplifying the calculations to a large extent.

$$\log\frac{f_k(\mathbf{x})\pi_k}{f_l(\mathbf{x})\pi_l} = \log\frac{|\mathbf{\Sigma}_l|^{1/2}\exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)\mathbf{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)^T\right]\pi_k}{|\mathbf{\Sigma}_k|^{1/2}\exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_l)\mathbf{\Sigma}_l^{-1}(\mathbf{x} - \boldsymbol{\mu}_l)^T\right]\pi_l} = \delta_k(\mathbf{x}) - \delta_l(\mathbf{x}) \tag{3.4}$$

where

$$\delta_i(\mathbf{x}) = -\frac{1}{2}\log|\mathbf{\Sigma}_i| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)\mathbf{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i)^T + \log\pi_i, \quad i \in \{k, l\} \tag{3.5}$$

The function $\delta_i(\mathbf{x})$ is called the discriminant function. There is one such function for each class and Equation 3.4 shows that point $\mathbf{x}$ is assigned to the class with the largest value of the corresponding discriminant function. Furthermore, the decision function $\delta_k(\mathbf{x}) - \delta_l(\mathbf{x}) = 0$ is quadratic. Hence, the classification procedure in this form is known as quadratic discriminant analysis (QDA). If we make the additional assumption that the group-specific variance-covariance matrices are equal, that is $\mathbf{\Sigma}_k = \mathbf{\Sigma}_l \equiv \mathbf{\Sigma}$, then the expression further simplifies into

$$\begin{aligned}\log\frac{f_k(\mathbf{x})\pi_k}{f_l(\mathbf{x})\pi_l} &= -\frac{1}{2}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_l)\mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_l)^T + \mathbf{x}\mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_l)^T + \log\frac{\pi_k}{\pi_l}\\ &= \delta_k(\mathbf{x}) - \delta_l(\mathbf{x}), \qquad (3.6)\end{aligned}$$

now with

$$\delta_i(\mathbf{x}) = -\frac{1}{2}\boldsymbol{\mu}_i\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_i^T + \mathbf{x}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_i^T + \log\pi_i, \quad i \in \{k, l\}. \qquad (3.7)$$

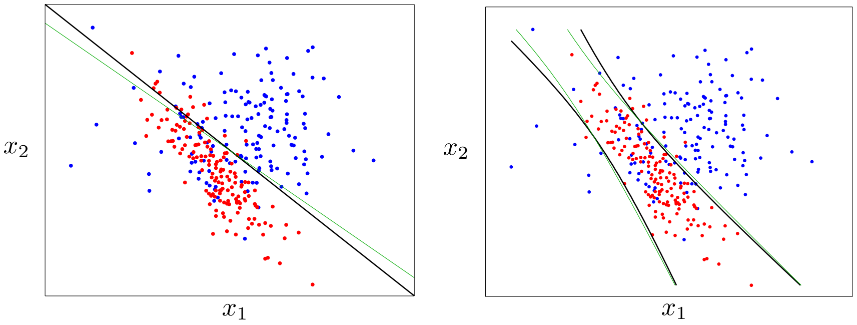
As seen, this additional assumption leads to linear discriminant functions and indeed, the procedure is known as linear discriminant analysis (LDA). If the relevant assumptions are met for each method, and the population means and dispersion matrices are known, these methods can be shown to be optimal, meaning that they will do better on average than any alternatives in such cases. Naturally, there are difficulties in making sure that the distributions are Gaussian and in the proper estimation of $\mathbf{\Sigma}_i$, $\boldsymbol{\mu}_i$ and π_i for all classes i. The standard approaches to estimating these are

$$\begin{aligned}\pi_i =& \frac{n_i}{n} & (3.8)\\ \boldsymbol{\mu}_i =& \frac{1}{n_i}\sum_{j\in\mathcal{G}_i}\mathbf{x}_j & (3.9)\\ \mathbf{\Sigma}_i =& \frac{1}{n_i - 1}\sum_{j\in\mathcal{G}_i}(\mathbf{x}_j - \boldsymbol{\mu}_i)^T(\mathbf{x}_j - \boldsymbol{\mu}_i) & (3.10)\\ \mathbf{\Sigma} =& \frac{1}{n-K}\sum_{i=1}^{K}\sum_{j\in\mathcal{G}_i}(\mathbf{x}_j - \boldsymbol{\mu}_i)^T(\mathbf{x}_j - \boldsymbol{\mu}_i). & (3.11)\end{aligned}$$

Figure 3.1 shows example results from using LDA and QDA on a data set with two classes containing 150 observations each. The elements of each group has been drawn from Gaussian distributions with parameters

$$\begin{cases}\boldsymbol{\mu}_1 &= [\ 0.5 \quad 0.5]\\ \boldsymbol{\mu}_2 &= [-0.5 \quad -0.5]\end{cases} \quad \mathbf{\Sigma}_1 = \begin{bmatrix} 3 & 0\\ 0 & 1\end{bmatrix} \quad \mathbf{\Sigma}_2 = \begin{bmatrix} 1 & -0.8\\ -0.8 & 1\end{bmatrix}. \qquad (3.12)$$

The figures show the estimated decision boundary in black. This boundary uses Equations 3.8–3.11 to estimate the parameters. Shown in green are the optimal decision boundaries where the population parameters in (3.12) have been used.



(a) Linear discriminant analysis.

(b) Quadratic discriminant analysis.

Figure 3.1: Classification using linear and quadratic discriminant analysis. Shown in blue and red are group 1 and group 2 respectively. Optimal boundaries are shown in green. The boundary estimated by QDA according to the equation $\delta_1(\mathbf{x}) - \delta_2(\mathbf{x}) = 0$ has split up, forming two ridges of zero-crossings isolating class 2 inside the more dispersed class 1.

3.2 Optimal Separating Hyperplanes

The type of discriminant analysis presented in the previous section is simple to implement and interpret, and often performs well. However, we note two properties that arguably limit the method.

- The decision boundary is estimated using information from the entire data set. This means that observations that are far from the boundary have as much influence on the results as observations that are close to the interface between the two classes. A method that focuses on this interface region, rather than the entire sample space is of interest.
- In relation to the former point, a weakness of LDA is that points in the training data set may be misclassified, also in cases where the data are separable using a single linear decision boundary. A sensible requirement of a classification method is that the estimated boundary separates the training data whenever this is possible.

The *Optimal separating hyperplane* (OSH) represents a classification method that attempts to resolve these limitations. As before, classes are separated using a hyperplane. However, while LDA uses the centroid and variance-covariance structure of each class to define this plane, the optimal separating hyperplane is positioned and oriented such that the separation between classes becomes maximal. This separation is defined as the distance from the hyperplane to the closest point in each class.

Before we give a formal explanation of OSH, a quick review of the distance from a point to a plane is reviewed. Given the plane $\mathcal{P} : \mathbf{x}\mathbf{b} + b_0 = 0$, the signed distance from an arbitrary point $\mathbf{x}_0$ to $\mathcal{P}$ is

$$D_{\mathcal{P}}(\mathbf{x}_0) = \frac{\mathbf{x}_0\mathbf{b} + b_0}{\|\mathbf{b}\|}. \tag{3.13}$$

To see this, note first that $\mathbf{b}$ is a normal vector to $\mathcal{P}$. Denote an arbitrary vector from $\mathcal{P}$ to $\mathbf{x}_0$ by $\mathbf{v} = \mathbf{x} - \mathbf{x}_0$. The projection of $\mathbf{v}$ onto $\mathbf{b}$ gives $\mathbf{v}\mathbf{b}/\|\mathbf{b}\| = (\mathbf{x}_0 - \mathbf{x})\mathbf{b}/\|\mathbf{b}\| = D_{\mathcal{P}}(\mathbf{x}_0)$, where the latter follows since $\mathbf{x}\mathbf{b} + b_0 = 0$.

It is convenient to define class labels by -1 and +1, such that the expression $y_i D_{\mathcal{P}}(\mathbf{x}_i)$ is positive for a correctly classified point x_i, regardless of which class it belongs to, and negative for misclassified points. That is, the class in the direction of the normal vector $\mathbf{b}$, for which $D_{\mathcal{P}}(\mathbf{x}_i)$ is positive, is assigned label +1 and conversely for the other class. In practice, it is irrelevant which class is assigned which label, since $\mathbf{b}$ is optimized upon in the OSH method; $\mathbf{b}$ will therefore take on a suitable direction to fit the chosen class labels at the optimum.

Using the above definitions, OSH can be formulated as a constrained maximization problem,

$$\arg\max_{\mathbf{b}, b_0} \quad C \qquad \text{subject to} \qquad y_i \frac{\mathbf{x}_i\mathbf{b} + b_0}{\|\mathbf{b}\|} \geq C, \quad \forall i. \tag{3.14}$$

In words, this optimization problem translates to "adjust plane parameters $\mathbf{b}$ and b_0 such that the smallest distance from the resulting plane to any point $\mathbf{x}_i$ is maximized". Classification is implied by this procedure, as any misclassified points result in negative distances and thus, a lower value of C.

Dropping the normalization term from Equation 3.13 we obtain a distance measure which is dependent on the length of $\mathbf{b}$. A short normal vector will result in relatively larger distances than if a longer normal vector is used as a metric for the distance calculation. This introduces another degree of freedom to the OSH optimization problem. However, we can instead choose to fix C (at $C = 1$ here) and write the setup as a minimization problem on the length of $\mathbf{b}$.

$$\arg\min_{\mathbf{b}, b_0} \quad \frac{1}{2}\|\mathbf{b}\|^2 \qquad \text{subject to} \qquad y_i(\mathbf{x}_i\mathbf{b} + b_0) \geq 1, \quad \forall i. \tag{3.15}$$

We choose to minimize the squared length of $\mathbf{b}$ divided by 2 rather than $\|\mathbf{b}\|$ since this simplifies the coming expressions. Trying to paraphrase this equivalent optimization problem would result in something along the lines of "find the smallest possible metric $\|\mathbf{b}\|$ such that the smallest distance from the plane to a point is equal to or greater than 1".

Equation 3.15 consists of a quadratic minimization criterion with affine constraints, yielding a convex optimization problem. This is a desirable property as this tells us that the solution is unique. However, a solution does not necessarily exist. If the data are not separable, there is no unique choice of $\mathbf{b}$ and b_0 such that the constraints in Equation 3.15 can be fulfilled.

Convex problems can often be reformulated in an equivalent (dual) manner which is easier to solve. Using Lagrange multipliers, we can incorporate the constraints into the criterion function, creating an unconstrained minimization problem. The resulting criterion function is

$$L_P = \frac{1}{2}\|\mathbf{b}\|^2 - \sum_{i=1}^{n} \alpha_i \left[y_i(\mathbf{x}_i\mathbf{b} + b_0) - 1 \right]. \tag{3.16}$$

We do, however, require that $\alpha_i \geq 0 \forall i$. As for any "unconstrained" function that is to be minimized, the derivatives are zero at the optimum. This yields,

$$\frac{\partial L_P}{\partial \mathbf{b}} = \mathbf{b} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^T = \mathbf{0} \qquad \Leftrightarrow \qquad \mathbf{b} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^T \tag{3.17}$$

$$\frac{\partial L_P}{\partial b_0} = \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3.18}$$

Using these results, Equation 3.16 can be written in its dual form,

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \mathbf{x}_i\mathbf{x}_j^T = \boldsymbol{\alpha}\mathbf{1}_n - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Y}\mathbf{X}\mathbf{X}^T\mathbf{Y}\boldsymbol{\alpha}, \tag{3.19}$$

where $\mathbf{Y} = \text{diag}(\mathbf{y})$. The resulting optimization problem is

$$\arg\max_{\boldsymbol{\alpha}} = \boldsymbol{\alpha}\mathbf{1}_n - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Y}\mathbf{X}\mathbf{X}^T\mathbf{Y}\boldsymbol{\alpha} \qquad \text{subject to} \qquad \alpha_i \geq 0, \quad \forall i. \tag{3.20}$$

Efficient solvers for quadratic programming problems such as this are available, or are part of larger computing environments. Once a solution $\boldsymbol{\alpha}$ has been obtained, $\mathbf{b}$ can be recovered using Equation 3.17. To recover b_0 we use that the constraint in Equation 3.15 is active for those points that are closest to $\mathcal{P}$. Denoting this set of points $\mathcal{M}$, the intercept is given by

$$b_0 = \frac{1 - y_i\mathbf{x}_i\mathbf{b}}{y_i} \quad \text{for some} \quad i \in \mathcal{M}, \tag{3.21}$$

which, since $y_i \in \{-1, +1\}$, can be expressed

$$b_0 = y_i - \mathbf{x}_i \mathbf{b} \qquad i \in \mathcal{M}, \tag{3.22}$$

The remaining question is how the set $\mathcal{M}$ is established. A useful identity is given by the complementary slackness condition

$$\alpha_i \left[y_i (\mathbf{x}_i \mathbf{b} + b_0) - 1 \right] = 0 \quad \forall i. \tag{3.23}$$

If $\alpha_i > 0$ then $y_i(\mathbf{x}_i \mathbf{b} + b_0) - 1$ must be equal to zero which means the constraint is active for this point — it belongs to $\mathcal{M}$. Conversely, if $y_i(\mathbf{x}_i \mathbf{b} + b_0) - 1 > 0$, α_i must be equal to zero. Such a point is not in $\mathcal{M}$.

Figure 3.2 shows results from applying OSH to a data set with two classes. The black line represents the hyperplane, while the thinner green lines show the resulting margin with width $2C = 2/\|\mathbf{b}\|$. Points in $\mathcal{M}$ are marked using black squares. The LDA decision function is also shown in red. LDA misclassifies one point in this data set.

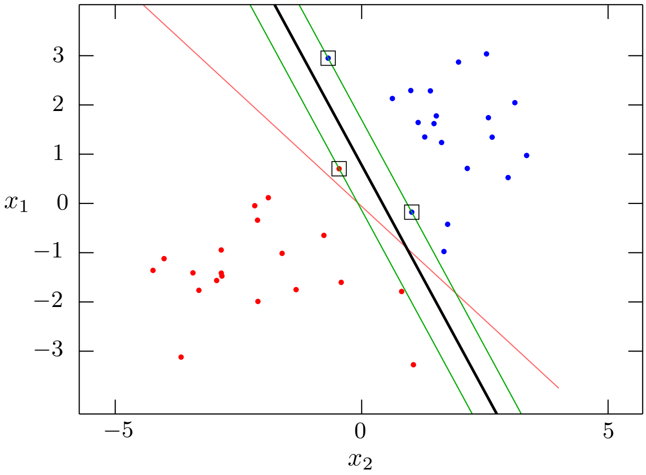


Figure 3.2: Optimal separating hyperplane for a small data set using a linear kernel. The hyperplane is shown in black, while the boundaries of the resulting margin are green. The LDA boundary is shown in red, misclassifying one training data point.

3.2.1 Non-linear Generalization

As for any statistical method, we can generalize the procedure to model non-linear decision boundaries using basis expansions. The idea is to solve the classification problem in an expanded space, and use the resulting boundary function to classify observations in the original space. If the dimensionality of the problem is increased in a sensible way, it becomes increasingly simple to separate between classes. In fact, in a sufficiently high-dimensional space, any data set without duplicate points is separable. Expanding each point using the transformation $h(\mathbf{x}_i)$, Equation 3.20 becomes

$$\arg\max_{\boldsymbol{\alpha}} = \boldsymbol{\alpha}\mathbf{1}_n - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Y}\mathbf{h}(\mathbf{X})\mathbf{h}(\mathbf{X})^T\mathbf{Y}\boldsymbol{\alpha} \qquad \text{subject to} \quad \alpha_i \geq 0, \quad \forall i. \tag{3.24}$$

The basis expansion $h(\mathbf{x}_i)$ increases the dimensionality of $\mathbf{x}_i$ from one to an arbitrary number of dimensions. Regardless of the dimensionality of $h(\mathbf{x})$, the resulting matrix of inner products $\mathbf{h}(\mathbf{X})\mathbf{h}(\mathbf{X})^T$ will have size $(n \times n)$. This means that the complexity of the resulting optimization problem is unchanged. It also means that we not necessarily have to specify the form of $h(\mathbf{x})$. Instead it is sufficient to specify the form of the inner product $h(\mathbf{x}_i)h(\mathbf{x}_j)^T$. This scalar value is commonly denoted $K(\mathbf{x}_i, \mathbf{x}_j)$ or simply $K_{i,j}$. The matrix of all combinations of i and j is denoted $\mathbf{K}$ and is called the *kernel matrix*. Common choices of kernels are

Linear kernel: $K_{i,j} = \mathbf{x}_i\mathbf{x}_j^T$

Polynomial kernel: $K_{i,j} = (1 + \mathbf{x}_i\mathbf{x}_j^T)^d$

Gaussian kernel: $K_{i,j} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma)$.

The linear kernel is equivalent to a first degree ($d = 1$) polynomial kernel and represents the original non-transformed formulation. Gaussian kernels are the most common choice for methods that can be fully specified using kernels. The width of the Gaussian kernel is tuned by the parameter σ. Large values of σ lead to smooth and coherent boundary functions while small values lead to wiggly and clustered results. For some kernels, there is a corresponding basis function $h(\mathbf{x})$ that can be specified. For other kernels such as the Gaussian kernel, this basis function is more difficult to specify, and may even be infinite-dimensional.

To classify an observation $\mathbf{x}_i$, we simply evaluate the resulting plane equation $\mathbf{x}_i\mathbf{b} + b_0$ and check the sign of the result. For visualization, it is of interest to obtain a functional expression for the resulting decision boundary in the original, non-expanded space. However, this is difficult for most kernels. Instead, the plane equation is evaluated over a fine grid of points covering the area of interest. We then find the approximate positions where this function changes sign and thus obtain a set of points on the boundary. The kernel formulation of the plane equation is

$$\mathcal{P} : \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b_0 = 0 \tag{3.25}$$

where

$$b_0 = y_i - \sum_{j=1}^{n} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \qquad i \in \mathcal{M}. \tag{3.26}$$

Note that we cannot recover an explicit representation of **b** unless the basis function corresponding to the employed kernel is known.

Figure 3.3 shows the classification problem from Figure 3.2 computed using OSH with a Gaussian kernel with $\sigma = 15$. For this rather large value of σ, the solution is not vastly different from the one obtained using a linear kernel. For smaller values of λ, the boundary becomes increasingly compact, surrounding one of the classes which is thereby separated from ”everything else”. Such a boundary will generalize poorly as new observations are classified.

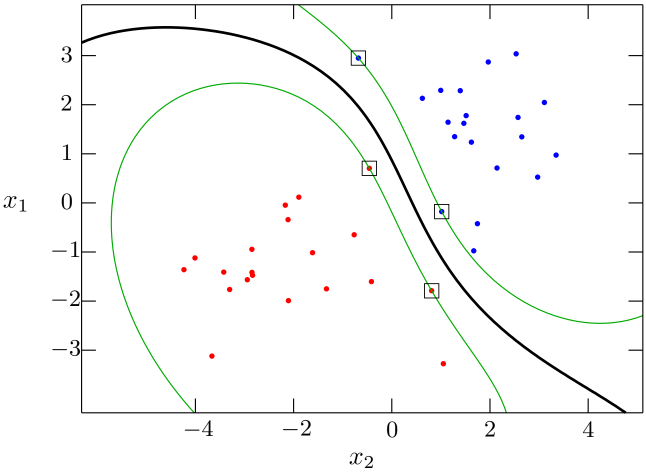
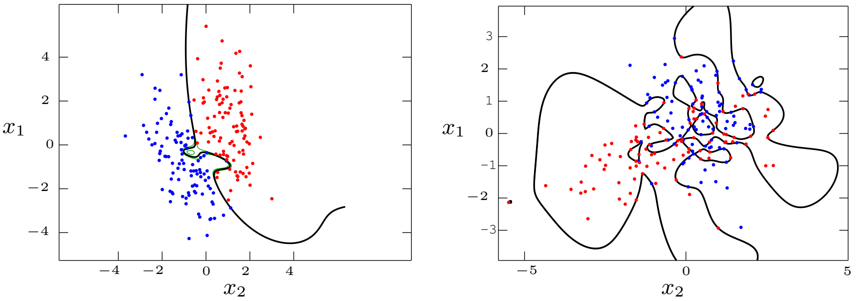


Figure 3.3: Optimal separating hyperplane for the same data set as in Figure 3.2 using a Gaussian kernel with $\sigma = 15$.

Returning to the reason we considered using basis expansions for OSH, we now consider overlapping data. In this case, we must use a sufficiently high-dimensional basis expansion. Here, we will exclusively use the Gaussian kernel, for which sufficiently high-dimensional translates to a sufficiently small value of σ. Figure 3.4 shows two cases. The data in Figure 3.4(a) have small overlap, and the resulting boundary for $\sigma = 500$ has a rather smooth shape, except in cases where a small detour must be made to avoid misclassifications. We have good reason to suspect that such small deviations from a more general shape are due to overfitting. In Figure 3.4(b) the overlap is more prominent, resulting in a severely overfit decision boundary, even with σ set as high as is computationally feasible at $\sigma = 1.5$. Clearly, this boundary will not be of much use as novel observations are to be classified, and is difficult to interpret.



(a) Small overlap

(b) More overlap

Figure 3.4: Using optimal separating hyperplanes to separate overlapping data. When the overlap is small, useful boundaries may be obtained, while severe overfitting is unavoidable if the overlap is large.

The following section presents a regularized form of optimal separating hyperplanes which is better equipped to deal with overlapping data.

3.3 Support Vector Machines

One way of dealing with overlapping data is to use the framework provided by optimal separating hyperplanes, but allowing a small set of points to be misclassified. If sufficiently many points are allowed on the wrong side of the decision boundary, the classification problem will have a solution, also in the original space using a linear kernel. One such approach is the support vector machine (SVM). Similarly to optimal separating hyperplanes, the SVM maximizes the margin between the classes, but allows observations to fall on the wrong side of the margin. If the distance between such an observation and the margin is greater than 1, the point is on the wrong side of the decision boundary and the point is misclassified. If a point $\mathbf{x}_i$ is on the wrong side of the margin, the corresponding distance is denoted ξ_i. Figure 3.5 introduces this notation. For points on the correct side of the margin, $\xi_i = 0$.

To minimize the misclassification rate, we would like the total distance $\sum_i \xi_i$ to be as low as possible. At the same time, the margin should be made as large as possible to ensure good separation between classes. These objectives work against each other, and a weighting term λ is introduced to balance this

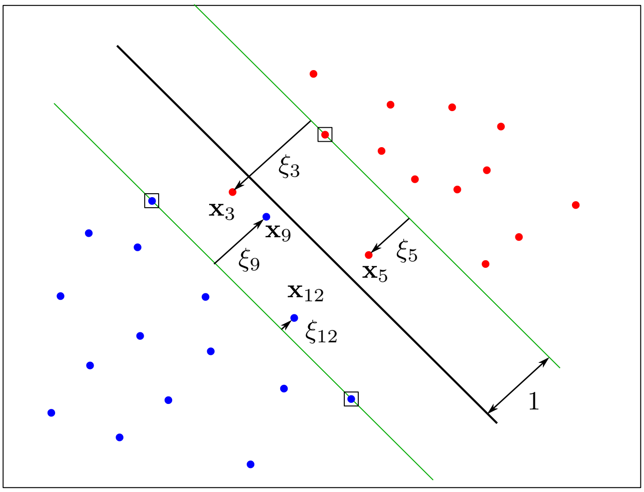


Figure 3.5: Notation and geometry of the support vector machine. Apart from the decision boundary, a *margin* with total width 2 is created where each point inside the margin has an associated distance measure ξ. Points with $\xi > 1$, e.g. $\mathbf{x}_3$, are misclassified. Distances are measured relative to the size of the normal $\mathbf{b}$ of the decision boundary.

trade-off. The resulting optimization problem is

$$\arg\min_{\mathbf{b},b_0} \quad \sum_{i=1}^{n} \xi_i + \frac{\lambda}{2}\|\mathbf{b}\|^2 \quad \text{subject to} \quad y_i(\mathbf{x}_i\mathbf{b} + b_0) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall i. \tag{3.27}$$

Most points $\mathbf{x}_i$ will reside on the correct side of the decision boundary at a distance greater that 1. For such points, ξ_i will be zero. For points closer to the boundary, $\xi_i > 0$. For large values of λ, focus is on constructing a large margin, and the ξ_i will become larger to make such a solution possible. For smaller values of λ, the focus is on minimizing the size of the ξ_i, resulting in a more narrow margin.

Standard computation of the SVM proceeds along the same lines as optimal separating hyperplanes. A primal optimization function is first constructed using Lagrange multipliers,

$$L_P = \sum_{i=1}^{n} \xi_i + \frac{\lambda}{2}\|\mathbf{b}\|^2 - \sum_{i=1}^{n} \alpha_i \left[y_i(\mathbf{x}_i\mathbf{b} + b_0) - 1 + \xi_i \right] - \sum_{i=1}^{n} \gamma_i \xi_i$$
$$\text{subject to} \quad \alpha_i \geq 0, \gamma_i \geq 0, \quad \forall i. \tag{3.28}$$

This function is then differentiated and set to zero, resulting in the following expressions,

$$\frac{\partial L_P}{\partial \mathbf{b}} = \lambda \mathbf{b} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^T = \mathbf{0} \qquad \Leftrightarrow \qquad \mathbf{b} = \frac{1}{\lambda} \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^T \tag{3.29}$$

$$\frac{\partial L_P}{\partial b_0} = \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3.30}$$

$$\frac{\partial L_P}{\partial \xi_i} = 1 - \alpha_i - \gamma_i = 0 \qquad \Leftrightarrow \qquad \alpha_i = 1 - \gamma_i \tag{3.31}$$

We also have the useful complementary slackness conditions,

$$\alpha_i \left[y_i(\mathbf{x}_i \mathbf{b} + b_0) - 1 + \xi_i \right] = 0 \tag{3.32}$$

$$\gamma_i \xi_i = 0 \tag{3.33}$$

Points on the correct side of the margin, with distances larger than 1, are thought of as being outside the margin, and are collected in a set $\mathcal{O}$. Points inside the margin are in the set $\mathcal{I}$. Some points will fall exactly on the edge of the margin, and are assigned to a set $\mathcal{M}$. From Equation 3.31, we see that $0 \leq \alpha_i \leq 1, \forall i$. Further, from Equation 3.32 we see that $\alpha_{i \in \mathcal{O}} = 0$. From Equations 3.30 and 3.33 we conclude that $\alpha_{i \in \mathcal{I}} = 1$. A multiplier α_i can be shown to be a continuous function of the regularization parameter λ, $\alpha_{i \in \mathcal{M}}$ will therefore travel from 1 to 0 as point $\mathbf{x}_i$ passes the edge of the margin from the inside and out, and vice versa.

Inserting Equations 3.29, 3.30 and 3.31 into Equation 3.28 and simplifying, we obtain the following dual function,

$$L_D = \boldsymbol{\alpha} \mathbf{1}_n - \frac{1}{2\lambda} \boldsymbol{\alpha}^T \mathbf{Y} \mathbf{X} \mathbf{X}^T \mathbf{Y} \boldsymbol{\alpha}, \tag{3.34}$$

where $\mathbf{Y} = \text{diag}(\mathbf{y})$. This function is remarkably similar to the corresponding function for optimal separating hyperplanes (cf. Equation 3.19). We see that this representation lends itself to the use of kernels, and kernel notation will be used from this point on. The resulting optimization problem is

$$\arg\max_{\boldsymbol{\alpha}} = \boldsymbol{\alpha} \mathbf{1}_n - \frac{1}{2\lambda} \boldsymbol{\alpha}^T \mathbf{Y} \mathbf{K} \mathbf{Y} \boldsymbol{\alpha} \qquad \text{subject to} \qquad \alpha_i \geq 0, \quad \forall i. \tag{3.35}$$

The equation for the hyperplanar decision boundary is denoted $f(\mathbf{x})$, and is equally similar to that of optimal separating hyperplanes,

$$\mathcal{P} : f(\mathbf{x}) = \frac{1}{\lambda} \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b_0 = 0 \tag{3.36}$$

where

$$b_0 = y_i - \frac{1}{\lambda} \sum_{j=1}^{n} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \qquad i \in \mathcal{M}. \tag{3.37}$$

Typically, this quadratic programming problem is solved for a particular value of λ using standard optimization software. Figure 3.6 shows results on the overlapping data set from Figure 3.4(b) using various values of λ. In the top row, a linear kernel has been used while the bottom row shows results from using a Gaussian kernel with $\sigma = 10$.

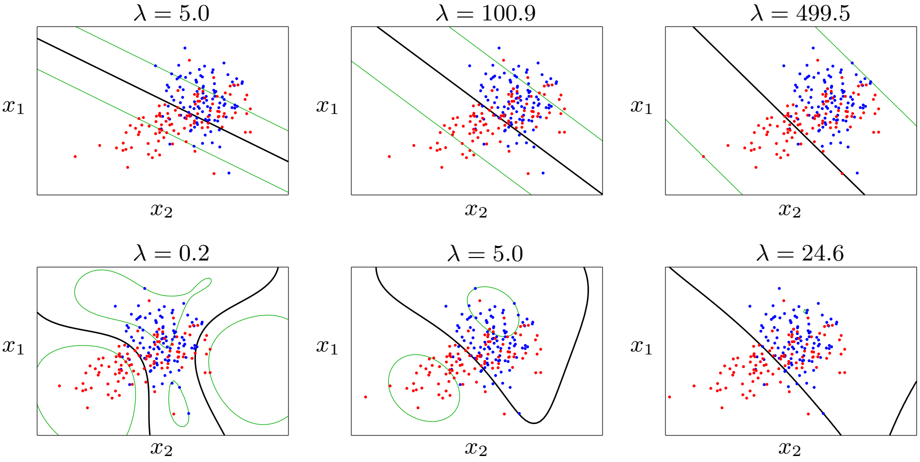


Figure 3.6: Classification of an overlapping data set using the support vector machine. In the top row, a linear kernel has been used while a Gaussian kernel with $\sigma = 10$ has been used in the bottom row. Black lines represent the decision function $f(\mathbf{x}) = 0$, while green lines represent the boundaries of the margin where $f(\mathbf{x}) = \pm 1$.

3.3.1 Computation

In line with the theme of this thesis, we will now review an algorithm that computes the entire regularization path of the SVM, using the fact that the Lagrange multipliers α_i are piece-wise linear functions of the regularization parameter λ. The proof and the algorithm takes an alternative but equivalent route compared to the original account by Hastie et al. [60]. The derivation presented here is arguably more direct and better suited for implementation.

We will begin by stating some useful definitions. We define the set $\mathcal{A}$ as the set including the indices of all training data points, i.e. $\mathcal{A} = \mathcal{I} \cup \mathcal{O} \cup \mathcal{M}$. Using indices and sets of indices, we define submatrices as $\mathbf{A}_{\{rows\},\{columns\}}$. For instance, the submatrix $\mathbf{K}_{i,\mathcal{A}}$ corresponds to the i^{th} row of the kernel matrix $\mathbf{K}$. The breakpoints at which the piecewise linear functions $\alpha_i(\lambda)$ change are called *events*. There are four types of events.

1. A point with index i joins the margin from the inside, $\mathcal{I} \rightarrow \mathcal{M}$.
2. A point with index i joins the margin from the outside, $\mathcal{O} \rightarrow \mathcal{M}$.
3. A point with index i leaves the margin for the inside, $\mathcal{M} \rightarrow \mathcal{I}$.
4. A point with index i leaves the margin for the outside, $\mathcal{M} \rightarrow \mathcal{O}$.

Between events, the sets do not change.

The specification of the SVM path algorithm consists of three parts. First, we derive the functional form of the multipliers $\alpha_i(\lambda)$ between events. We then discuss at what value of λ the next event occurs. This makes it possible to trace the multipliers along the path. In the third part, we take a look at how to find a suitable starting point on the path.

To derive an expression for the multipliers, we begin by specifying the distance function (plane equation) $f(\mathbf{x})$ evaluated for points in the training data set in matrix form,

$$\begin{aligned} f(\mathbf{x}_j) &= \frac{1}{\lambda}\mathbf{K}_{j,\mathcal{A}}\mathbf{Y}\boldsymbol{\alpha} + b_0 = && (3.38)\\ &= \frac{1}{\lambda}(\mathbf{K}_{j,\mathcal{A}} - \mathbf{K}_{i,\mathcal{A}})\mathbf{Y}\boldsymbol{\alpha} + y_i \qquad i \in \mathcal{M} && (3.39)\end{aligned}$$

Next, we note that $y_j f(\mathbf{x}_j) = 1 \quad \forall j \in \mathcal{M}$. This results in $|\mathcal{M}|$ equations that again can be summarized in matrix form,

$$\frac{1}{\lambda}\mathbf{Y}_{\mathcal{M},\mathcal{M}}\left[\mathbf{K}_{\mathcal{M},\mathcal{A}} - \mathbf{1}_{|\mathcal{M}|}\mathbf{K}_{i,\mathcal{A}}\right]\mathbf{Y}\boldsymbol{\alpha} + \mathbf{y}_{\mathcal{M}}y_i = \mathbf{1}_{|\mathcal{M}|} \qquad i \in \mathcal{M} \tag{3.40}$$

Using that $\alpha_j = 0 \quad \forall j \in \mathcal{O}$ and $\alpha_j = 1 \quad \forall j \in \mathcal{I}$, this expression can be expanded and rearranged into

$$\begin{aligned}\mathbf{Y}_{\mathcal{M},\mathcal{M}}\left[\mathbf{K}_{\mathcal{M},\mathcal{M}} - \mathbf{1}_{|\mathcal{M}|}\mathbf{K}_{i,\mathcal{M}}\right]\mathbf{Y}_{\mathcal{M},\mathcal{M}}\boldsymbol{\alpha}_{\mathcal{M}} &= \lambda(\mathbf{1}_{|\mathcal{M}|} - \mathbf{y}_{\mathcal{M}}y_i)\\ &- \mathbf{Y}_{\mathcal{M},\mathcal{M}}\left[\mathbf{K}_{\mathcal{M},\mathcal{I}} - \mathbf{1}_{|\mathcal{M}|}\mathbf{K}_{i,\mathcal{I}}\right]\mathbf{Y}_{\mathcal{I},\mathcal{I}}\mathbf{1}_{|\mathcal{I}|} \quad i \in \mathcal{M} \qquad (3.41)\end{aligned}$$

By defining a matrix $\mathbf{H} = \mathbf{Y}\left[\mathbf{K} - \mathbf{1}_n\mathbf{K}_{i,\mathcal{A}}\right]\mathbf{Y}$ we can reduce this rather cumbersome expression into

$$\mathbf{H}_{\mathcal{M},\mathcal{M}}\boldsymbol{\alpha}_{\mathcal{M}} = \lambda(\mathbf{1}_{|\mathcal{M}|} - \mathbf{y}_{\mathcal{M}}y_i) - \mathbf{H}_{\mathcal{M},\mathcal{I}}\mathbf{1}_{|\mathcal{I}|} \quad i \in \mathcal{M} \tag{3.42}$$

For $j = i$, the i^{th} row of $\mathbf{H}$ will be zero, making the system of equations rank deficient. However, we can replace this degenerate equation by the relation in Equation 3.30. Writing this equation

$$\mathbf{y}^T\boldsymbol{\alpha} = \mathbf{y}_{\mathcal{M}}^T\alpha_{\mathcal{M}} + \mathbf{y}_{\mathcal{I}}^T\mathbf{1}_{|\mathcal{I}|} \tag{3.43}$$

we see that it is sufficient to replace the i^{th} row of $\mathbf{H}$ by $\mathbf{y}^T$. We call this augmented matrix $\tilde{\mathbf{H}}$. This matrix and its relevant submatrices are now full rank and we can obtain the following expression for the Lagrange multipliers as functions of λ,

$$\alpha_{\mathcal{M}} = \lambda\tilde{\mathbf{H}}_{\mathcal{M},\mathcal{M}}^{-1}(1 - \mathbf{y}_{\mathcal{M}}y_i) - \tilde{\mathbf{H}}_{\mathcal{M},\mathcal{M}}^{-1}\tilde{\mathbf{H}}_{\mathcal{M},\mathcal{I}}\mathbf{1}_{|\mathcal{I}|} = \lambda\mathbf{p}_{\mathcal{M}} + \mathbf{q}_{\mathcal{M}} \tag{3.44}$$

This shows that the multipliers are linear functions of λ. The equation for the complete set of multipliers is $\boldsymbol{\alpha} = \lambda\mathbf{p} + \mathbf{q}$ where the elements of $\mathbf{p}$ and $\mathbf{q}$ for sets $\mathcal{I}$ and $\mathcal{O}$ are zero, except for $\mathbf{q}_{\mathcal{I}} = 1$, which ensures $\boldsymbol{\alpha}_{\mathcal{I}} = \mathbf{1}_{|\mathcal{I}|}$.

Now that we have derived the behavior of the multipliers between events, we seek a value λ_{l+1} of λ where the next event occurs. For reasons that will be explained below, we trace the path backwards, starting at a large value of λ and moving towards smaller values. The value of λ where the last event occurred is denoted λ_l. We are therefore seeking the value $\lambda_{l+1} < \lambda_l$ of the next event. We treat each of the four events defined above separately.

1. When the first event ($\mathcal{I} \to \mathcal{M}$) occurs for a point $\mathbf{x}_j$, the distance from $\mathbf{x}_j$ to the decision boundary will be exactly 1. For each point in $\mathcal{I}$ we can derive the value of λ at which this event occurs by solving the equation $y_j f(\mathbf{x}_j) = 1, \quad j \in \mathcal{I}$ for λ,

$$\begin{aligned} \frac{1}{\lambda}\mathbf{H}_{j,\mathcal{A}}(\lambda\mathbf{p} + \mathbf{q}) + y_j y_i = 1, \quad i \in \mathcal{M}, j \in \mathcal{I} \quad &\Leftrightarrow \\ \boldsymbol{\lambda} = \frac{\mathbf{H}_{\mathcal{I},\mathcal{A}}\mathbf{q}}{1 - \mathbf{y}_{\mathcal{I}}y_i - \mathbf{H}_{\mathcal{I},\mathcal{A}}\mathbf{p}}, \quad i \in \mathcal{M}.& \end{aligned} \tag{3.45}$$

   The result is $|\mathcal{I}|$ candidate values of λ.

2. Equivalently, the second event ($\mathcal{O} \to \mathcal{M}$) occurs at the following values of λ,

$$\boldsymbol{\lambda} = \frac{\mathbf{H}_{\mathcal{O},\mathcal{A}}\mathbf{q}}{1 - \mathbf{y}_{\mathcal{O}}y_i - \mathbf{H}_{\mathcal{O},\mathcal{A}}\mathbf{p}} \tag{3.46}$$

3. When the third event ($\mathcal{M} \to \mathcal{I}$) occurs for a point $\mathbf{x}_j$, its corresponding multiplier will obtain $\alpha_j = 1$. For all points in $\mathcal{M}$, we can find the values of

λ where this event occurs be setting Equation 3.44 equal to 1 and solving for λ,

$$\boldsymbol{\lambda} = \frac{1 - \mathbf{q}_{\mathcal{M}}}{\mathbf{p}_{\mathcal{M}}} \tag{3.47}$$

4. Similarly, the final event ($\mathcal{M} \to \mathcal{O}$) occurs as $\alpha_j = 0$. The corresponding values of λ are

$$\boldsymbol{\lambda} = -\frac{\mathbf{q}_{\mathcal{M}}}{\mathbf{p}_{\mathcal{M}}} \tag{3.48}$$

A candidate value of λ is calculated for each point and each relevant event, resulting in one candidate value for each point in $\mathcal{I}$ and $\mathcal{O}$, and two candidate values for each point in $\mathcal{M}$. Out of these candidates, the value of λ where the next event occurs must be the largest candidate value λ_{l+1} such that $\lambda_{l+1} < \lambda_l$.

Finally, we define a suitable starting point for the path. Using the standard quadratic programming technique to obtain values of the multipliers α_i for a particular value of λ allows us to start the algorithm at any point along the path. However, we would like to start the algorithm at one of the endpoints of the path and work our way towards the other. Computationally beneficial solutions occurs for very large values of λ, corresponding to a very large margin. There are two types of behavior of the SVM for large values of λ, depending on whether there are equally many observations in each class or not. As in [60], we let n_- and n_+ denote the number of observations in each class, and $\mathcal{I}_-$ and $\mathcal{I}_+$ denote the corresponding indices for points inside the margin.

If $n_- = n_+$, there is a large value of λ at which one observations from each class enters $\mathcal{M}$ from $\mathcal{I}$, while the rest of the observations remain in $\mathcal{I}$. The two must enter concurrently — otherwise we have a violation of Equation 3.30. If the plane equation (3.38) for the decision boundary was known, we could identify these observations by finding the most remote observation in each class. However, λ and b_0, which are the unknown elements of this expression, are equal for all observations. Further, we have $\boldsymbol{\alpha} = \mathbf{1}$. Hence, we have,

$$f(\mathbf{x}_j) \propto \mathbf{K}_{j,\mathcal{A}}\mathbf{Y}\boldsymbol{\alpha} = \mathbf{K}_{j,\mathcal{A}}\mathbf{y}. \tag{3.49}$$

Therefore, the first observation from each class to enter $\mathcal{M}$ is the one with the maximal distance according to this equation with $j \in \mathcal{I}_-$ and $j \in \mathcal{I}_+$ respectively. Now that the indices of the elements in $\mathcal{M}$ have been disclosed, we can find the corresponding values of λ and b_0, again using Equation 3.38, yielding two equations with two unknowns. The system of equations is

$$\mathbf{y}_{\mathcal{M}}[\mathbf{K}_{\mathcal{M},\mathcal{A}}\mathbf{y} \quad \mathbf{1}] \begin{bmatrix} 1/\lambda \\ b_0 \end{bmatrix} = \mathbf{1}, \tag{3.50}$$

which can be solved to obtain λ and b_0.

The unbalanced case where e.g. $n_+ > n_-$ is both computationally and theoretically more difficult. For large values of λ and $n_+ > n_-$, the plus-side margin will stay positioned among the most remote observations in $\mathcal{I}_+$ while the decision boundary and the opposite margin moves closer as λ shrinks. The α_i remain constant until the minus-side margin reaches one of the observations in $\mathcal{I}_+$. We wish to find the value of λ at which this event occurs. Up to this point, $\alpha_i = 1, i \in \mathcal{I}_-$, meaning that $\mathbf{y}_{\mathcal{I}_-}^T \boldsymbol{\alpha}_{\mathcal{I}_-} = -n_-$. As always, we require Equation 3.30 to hold. Therefore, $\sum_{i=1}^n \alpha_i = 2n_-$. This sum will remain constant until the first event occurs. Maximizing the dual in Equation 3.28 is therefore equivalent to the following minimization problem in this case,

$$\begin{gathered}\arg\min_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^T \mathbf{YKY}\boldsymbol{\alpha} \\ \text{subject to} \quad \alpha_i = 1 \quad \forall i \in \mathcal{I}_-, \quad 0 \le \alpha_i \le 1 \quad \forall i \in \mathcal{I}_+, \quad \sum_{i=1}^n \alpha_i = 2n_- \end{gathered} \tag{3.51}$$

Using that $\alpha_i = 1 \quad \forall i \in \mathcal{I}_-$, the setup can be further simplified. To see this, we expand the criterion function into parts belonging to each class.

$$\begin{aligned} &\boldsymbol{\alpha}^T \mathbf{YKY}\boldsymbol{\alpha} \\ &\quad = \begin{bmatrix}\boldsymbol{\alpha}_+^T & \boldsymbol{\alpha}_-^T\end{bmatrix} \begin{bmatrix}\mathbf{Y}_+ & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_-\end{bmatrix} \begin{bmatrix}\mathbf{K}_+ & \mathbf{K}_{+-} \\ \mathbf{K}_{-+} & \mathbf{K}_-\end{bmatrix} \begin{bmatrix}\mathbf{Y}_+ & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_-\end{bmatrix} \begin{bmatrix}\boldsymbol{\alpha}_+ \\ \boldsymbol{\alpha}_-\end{bmatrix} \\ &\quad = \boldsymbol{\alpha}_+^T \mathbf{Y}_+ \mathbf{K}_+ \mathbf{Y}_+ \boldsymbol{\alpha}_+ + 2\mathbf{y}_-^T \mathbf{K}_{-+} \mathbf{Y}_+ \boldsymbol{\alpha}_+ + \mathbf{y}_-^T \mathbf{K}_- \mathbf{y}_- \end{aligned} \tag{3.52}$$

This results in the reduced minimization problem

$$\begin{gathered}\arg\min_{\boldsymbol{\alpha}_+} \mathbf{y}_-^T \mathbf{K}_{-+} \mathbf{Y}_+ \boldsymbol{\alpha}_+ + \frac{1}{2} \boldsymbol{\alpha}_+^T \mathbf{Y}_+ \mathbf{K}_+ \mathbf{Y}_+ \boldsymbol{\alpha}_+ \\ \text{subject to} \quad 0 \le \alpha_i \le 1 \quad \forall i \in \mathcal{I}_+, \quad \sum_{i \in \mathcal{I}_+} \alpha_i = n_- \end{gathered} \tag{3.53}$$

The corresponding setup for the case where $n_- > n_+$ is equivalent, but with a change of signs.

Studying the values of α that we obtain from this procedure, we can identify points in $\mathcal{M}$ if $0 < \alpha < 1$. Points in $\mathcal{I}$ are harder to identify, as we cannot distinguish these from points that have just entered $\mathcal{M}$ from $\mathcal{I}$. The same goes for points in $\mathcal{O}$ which cannot be separated from points on the interface between $\mathcal{M}$ and $\mathcal{O}$. However, there are other ways to classify the remaining points. There are two possible cases. The first is when $\mathcal{M}$ is empty and the surplus of multipliers from $\mathcal{I}_+$ are in $\mathcal{O}_+$. In this case, we end up with the balanced situation where $|\mathcal{I}|_- = |\mathcal{I}|_+$. At this instant, one element from each class has just entered $\mathcal{M}$ from $\mathcal{I}$, otherwise Equation 3.30 would be violated. We identify these variables in the same way as for the balanced initialization procedure described above. Note that Equation 3.49 is used with the current vector of multipliers instead of $\boldsymbol{\alpha} = \mathbf{1}$. The other possibility is that $\mathcal{M}$ is nonempty and contains positive elements with $0 < \alpha_i < 1$. In this case, a single element from $\mathcal{I}_-$ has just entered $\mathcal{M}$ and has $\alpha_i = 1$. To identify which element this is, Equation 3.49 is used to calculate the distances for all elements in $\mathcal{I}_-$, and the most remote observation is singled out. The calculation of λ and b_0 then proceeds as above, using Equation 3.50.

At any point along the path, the margin set $\mathcal{M}$ may become empty. Two new points will then join $\mathcal{M}$ in the next event, one from each class. This is the same situation as for the balanced initialization procedure described above, but where the current states of $\mathcal{I}$ and $\boldsymbol{\alpha}$ are used.

After this ordeal, we arrive at a complete algorithm for computing the SVM path. Algorithm 3.1 states the procedure.

Figure 3.7 depicts the paths corresponding to the two rows of images in Figure 3.6. In the top path, a linear kernel as been used, while a Gaussian kernel with $\sigma = 10$ was employed in the bottom path. Typically, less constrained solutions, such as those obtained using a Gaussian kernel, lead to more complicated paths.

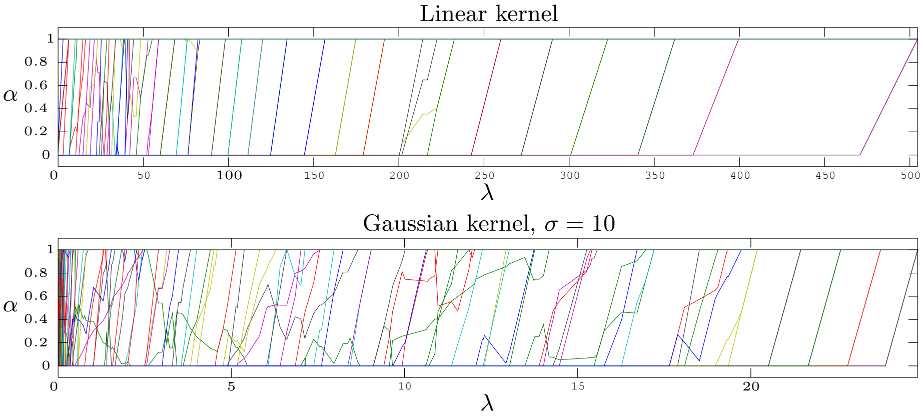


Figure 3.7: Example paths of the support vector machine, corresponding to the rows of images in Figure 3.6. The top path corresponds to a linear kernel, while a Gaussian kernel with $\sigma = 10$ has been used in the bottom path.

Algorithm 3.1 Path Algorithm for the Support Vector Machine

```
Init $\mathcal{I} = \mathcal{A}$, $\mathcal{B} = \emptyset$, $\mathcal{O} = \emptyset$, $\boldsymbol{\alpha} = \mathbf{1}$.
if classes are balanced such that $n_+ = n_-$ then
    For all elements in $\mathcal{I}$, calculate distances from margin according to Equation 3.49.
    Find most remote element in $\mathcal{I}_+$ and $\mathcal{I}_-$.
    Move indices corresponding to these elements from $\mathcal{I}$ to $\mathcal{M}$.
    Calculate $\lambda$ and $b_0$ corresponding to this event using Equation 3.50.
else
    if $n_- > n_+$ then
        Switch the sets $\mathcal{I}_+$ and $\mathcal{I}_-$ by setting $\mathbf{y} = -\mathbf{y}$ and switching $n_+$ and $n_-$.
    end if
    Calculate $\boldsymbol{\alpha}$ using the quadratic programming setup of Equation 3.3.1
    Update sets according to the values of $\boldsymbol{\alpha}$.
    if $\mathcal{M} = \emptyset$ then
        Add the most remote point in $\mathcal{I}_+$ to $\mathcal{M}$ as done in the balanced case above.
    end if
    Add the most remote point in $\mathcal{I}_-$ to $\mathcal{M}$ as done in the balanced case above.
    Calculate $\lambda$ and $b_0$ according to Equation 3.50, negating the answer if the sets $\mathcal{I}_+$ and $\mathcal{I}_-$ were switched above.
end if
while $\lambda > \lambda_{\min}$ do
    if $\mathcal{M} = \emptyset$ then
        Move elements to $\mathcal{M}$ and find new values of $\lambda$ and $b_0$ according to the balanced case above.
    else
        Compute $\mathbf{p}$ and $\mathbf{q}$ according to Equation 3.44.
        Calculate $\lambda$ candidates according to event 1 using (3.45).
        Calculate $\lambda$ candidates according to event 2 using (3.46).
        Calculate $\lambda$ candidates according to event 3 using (3.47).
        Calculate $\lambda$ candidates according to event 4 using (3.48).
        Choose candidate $\lambda_{l+1}$ with the largest value smaller than $\lambda_l$.
        Calculate new coefficients, $\boldsymbol{\alpha} = \lambda_{l+1}\mathbf{p} + \mathbf{q}$ and $b_0 = y_i - \mathbf{K}_{i,\mathcal{A}}\mathbf{Y}\boldsymbol{\alpha}$ for some $i \in \mathcal{M}$.
        Update sets accordingly.
    end if
end while
```

3.4 Support Vector Domain Description

The support vector domain description (SVDD) is a technique that is strongly related to the support vector machine, but is solving a different type of classification problem. Previous classification methods in this chapter discriminate between known classes, using a training data set to build a classifier which is then used to classify new observations. *Clustering* is another class of statistical methods that also divides the input data into regions according to characteristics of the data set, but whereas classification methods depend on the training set of paired observations and labels, clustering methods estimate regions directly from the properties of the data in $\mathbf{X}$; there is no vector $\mathbf{y}$ of labels to guide the process. The SVDD is a method that falls somewhere in-between classification and clustering. The goal of the method is to separate trustworthy data from outliers. This process is known as *one-class classification* as we try to isolate one class from "everything else". Another name is *data description* or *domain description*. The resulting decision boundary will encapsulate interesting observations and leave out uninteresting ones, leading to a geometry that is characteristic for the support of the data set. Similar to clustering, there are no labels to guide the process; instead it relies on differences found in the variables themselves.

The SVDD models the decision boundary using a hypersphere. Observations enclosed by this function are considered trustworthy data while points on the outside are treated as outliers. The hypersphere is specified by its p-dimensional center $\mathbf{a}$ and its scalar radius R. Figure 3.8 outlines the geometry of one solution for the SVDD in $p = 2$ dimensions. The variable ω_i represents the perpendicular distance from the boundary to an exterior point $\mathbf{x}_i$. For interior points, and points positioned exactly on the boundary, $\omega_i = 0$. The distance ω_i corresponding to an exterior point i can be written $\omega_i = \|\mathbf{x}_i - \mathbf{a}\| - R$, however, in the following we will use the closely related measure $\xi_i = \|\mathbf{x}_i - \mathbf{a}\|^2 - R^2$. Similarly to support vector machines, the SVDD is defined through an optimization problem that estimates the parameters $\mathbf{a}$, R and $\boldsymbol{\xi}$ such that the volume of the hypersphere is as small as possible, but also such that the sum of distances $\boldsymbol{\xi}$ to points outside the boundary is kept low,

$$\arg\min_{R,\mathbf{a},\xi_i} \sum_{i=1}^{n} \xi_i + \lambda R^2, \quad \text{subject to} \quad \|\mathbf{x}_i - \mathbf{a}\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0 \quad \forall i, \tag{3.54}$$

In Sections 3.2 and 3.3, we turned this type of problem into a simpler, dual, problem using Lagrange multipliers. The same procedure applies here,

$$L_p : \arg\min_{R,\mathbf{a},\xi_i} \sum_{i=1}^{n} \xi_i + \lambda R^2 + \sum_{i=1}^{n} \alpha_i(\|\mathbf{x}_i - \mathbf{a}\|^2 - R^2 - \xi_i) - \sum_{i=1}^{n} \gamma_i \xi_i. \tag{3.55}$$

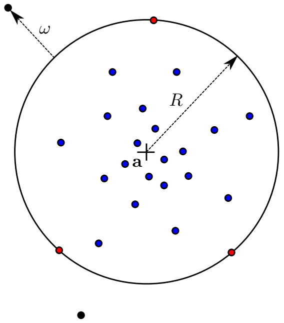


Figure 3.8: The geometry of the SVDD in two dimensions. Red, blue and black dots represent boundary points (3), data (20) and outliers (2) respectively. The hypersphere radius and center is denoted R and $\mathbf{a}$ respectively while ω is the distance from the boundary to an exterior point.

We take derivatives of this function to obtain the following useful relations,

$$\frac{\partial L_p}{\partial R} = 0 \qquad \Leftrightarrow \qquad \lambda = \sum_i \alpha_i, \tag{3.56}$$

$$\frac{\partial L_p}{\partial \mathbf{a}} = 0 \qquad \Leftrightarrow \qquad \mathbf{a} = \frac{\sum_i \alpha_i \mathbf{x}_i}{\sum_i \alpha_i} = \frac{\sum_i \alpha_i \mathbf{x}_i}{\lambda}, \tag{3.57}$$

$$\frac{\partial L_p}{\partial \xi_i} = 0 \qquad \Leftrightarrow \qquad \alpha_i = 1 - \gamma_i. \tag{3.58}$$

The complimentary slackness conditions are,

$$\alpha_i(\|\mathbf{x}_i - \mathbf{a}\|^2 - R^2 - \xi_i) = 0, \tag{3.59}$$

$$\gamma_i \xi_i = 0. \tag{3.60}$$

Inserting Equations (3.56-3.58) into (3.55) results in the dual formulation which is to be maximized with respect to (3.56-3.58),

$$L_d : \arg\max_{\alpha_i} \sum_{i=1}^{n} \alpha_i \mathbf{x}_i \mathbf{x}_i^T - \frac{1}{\lambda} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \mathbf{x}_i \mathbf{x}_j^T \quad : \quad 0 \le \alpha \le 1, \quad \sum_{i=1}^{n} \alpha_i = \lambda. \tag{3.61}$$

Again, we arrive at a compact quadratic optimization problem with linear constraints which can be solved using standard software. In Chapter 9 we present a detailed description of a path algorithm for the SVDD. Examples and results are also deferred to this section.

3.5 References

In this chapter, we have merely scratched the surface on the body of available methods for classification. Other methods include logistic regression and other generalized linear models, naive Bayes classification, k-nearest neighbor classification, boosting, classification trees, neural networks and Gaussian mixture models, most of which are discussed by Hastie et al. [59]. Another good reference for a variety of classification methods is Ripley [119].

An example of a regularized classification method is given by Friedman [45], who propose a trade-off between linear and quadratic discriminant analysis to deal with problems with many variables and few observations.

Linear and quadratic discriminant analysis were pioneered by R.A. Fisher [44]. LDA is also known as Fisher's linear discriminant.

The optimal separating hyperplane is an improvement over Rosenblatt's *perceptron* algorithm [121] from 1958, and is discussed by Vapnik [165]. Vapnik also developed the support vector machine which is presented in the same reference.

The support vector domain description was developed by Tax and Duin [151] and again with a more thorough treatment in [152].

Chapter 4

Principal Component Analysis

In this chapter, we will review the third class of statistical methods addressed in this thesis, techniques for data decomposition and dimensionality reduction. Several classes of such methods exits, and we mention a few of them among the references in Section 4.4. Focus is, however, on a family of methods related to principal component analysis (PCA), the most widely applicable and popular of such methods.

In order to motivate the use of PCA to accomplish a reduction of the number of dimensions of a data set, we take a look at a phenomenon known as the *curse of dimensionality*. Imagine a set of data points in a single dimension. Partitioning this dimension within a finite range into, say, 100 bins, our data may occupy most of these if we have a few hundred observations. If we consider higher-dimensional data, and additional dimensions are partitioned in the same fashion, the total number of bins is 100^p. For $p = 10$ dimensions, we have $100^{10} = 100\,000\,000\,000\,000\,000\,000$ bins. In such a case, a data set consisting of hundreds of observations will occupy an infinitesimal part of the total input space, and it seems unlikely that such a sparse representation of the world it inhabits can provide meaningful information. Put in a different way, if n observations occupy 75 % of the bins in a single dimension, we would need n^{10} observations to get the same approximate coverage. If our observations for instance consist of hospital patients followed over several years in a study of aging (cf. Chapter 7), where tens or hundreds of variables are collected, we cannot get anywhere near the number of observations necessary to densely populate the predictor space. Instead, a few hundred such observations must be considered an unusually large data set. The question is if all hope of a reliable high-dimensional analysis is lost? Luckily, it turns out that a data set may occupy a much larger space in *structured* ways. If such a structure is known or can be modelled sufficiently accurately, the analysis can be restricted to this subspace. For instance, it can be shown that a regression analysis can be performed with high-dimensional data if the assumptions stated in Section 2.4 are met and the noise variance is sufficiently low; the expected prediction error increases linearly as a function of p with slope σ_ε^2/n is such cases [59], compared to an exponential increase in the non-structured case. PCA can be used to find an explicit representation of such subspace structures.

Dimensionality reduction and PCA can also be described as the search for patterns of behavior in the data. A familiar example of a high-dimensional system is the weather, which is influenced by parameters such as location, temperature, humidity, wind speed, seasons, moon phases, etc. When we discuss or try to predict the weather, it is natural to simplify the analysis by using knowledge of relations between variables. Summers are generally hot, cold weather is accompanied by low humidity while thunderstorms occur when hot and cold air streams meet. By considering such connections, the apparent dimensionality of the problem is reduced.

PCA in its standard form is applicable to Gaussian data, but is fairly robust to deviations from normality as long as the data is reasonably symmetric about its centroid. The shape of a set of observations in $\mathbb{R}^p$ is given by the variance-covariance matrix. If $\text{cov}(\mathbf{X}) = \mathbf{\Sigma}$ is some multiple of the identity matrix, this corresponds to a spheroidal distribution, while a general diagonal matrix corresponds to a point cloud that is elongated along the direction of each coordinate axis. For an arbitrary variance-covariance matrix, the non-zero covariances rotate (and scale) the point cloud. From a geometrical viewpoint, the goal of PCA is to establish this rotation and scaling, such that we can find the most important directions through the data set — the axes of the ellipsoid it describes. To pin down the terminology, these axes are called the *principal axes.* The length-n vector of observations obtained by projecting the data onto the i^{th} principal axis is the i^{th} *principal component* (PC). The variance of the i^{th} PC is proportional to the length of the i^{th} major axis of the ellipsoid. Figure 4.1 shows this geometry, using a Gaussian data set in $p = 2$ dimensions of $n = 1000$ observations. The ellipse shows the shape of the distribution as specified by the variance-covariance matrix and encloses 2.5 standard deviations of the data in each direction, corresponding to 98.8 % of the total variance. The red arrows represent the original coordinate axes while the green arrows denote the principal axes of the data, or equivalently, the major axes of the ellipse. These arrows extend three standard deviations (99.7 % of the total variance).

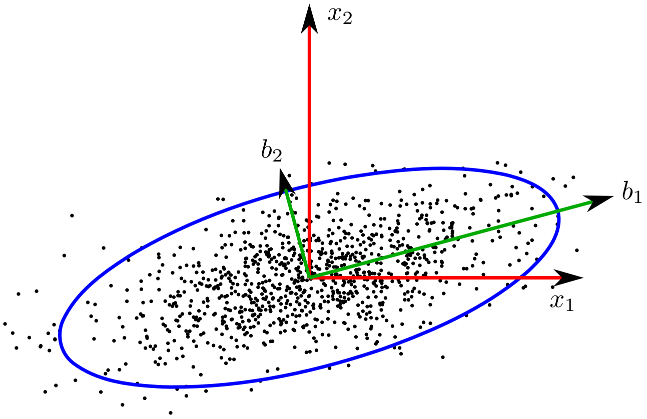


Figure 4.1: The geometry of PCA for a Gaussian data set in two dimensions. The ellipsoid extends 2.5 standard deviations along each derived coordinate axis an captures 98.8 % of the total variance.

The principal components can be viewed as a new set of observed variables which describe the data in an informative manner. First of all, they are uncorrelated, roughly meaning that we can talk about one component without having to refer to others. Second, the new variables consist of (linear) combinations of the original variables, which is why we may interpret each PC as a typical behavior of the system the data portray. Another important property of the principal components is that they exhibit a natural ordering according to the variance they describe. The first component describes the largest portion of the total variance. The second component contains as much information as possible without referring to the first, and so on. The variance described by each component quickly drops as we regard more directions, often making the contribution of later components insignificant. This leads to the *dimensionality reduction* property of PCA. We can simply disregard components with variances below some threshold. The number of dimensions of the new problem is denoted k, where $k \leq p$.

After this introduction to PCA, we now review approaches to computing the principal components and axes. In line with previous chapters, let $\mathbf{X}$ be the $(n \times p)$ data matrix with n observations and p variables. Similar to Chapter 2, we assume the variables have been mean centered, but not necessarily normalized.

We formulate PCA as the search for a rotation matrix $\mathbf{B}$ with $\mathbf{B}^T\mathbf{B} = \mathbf{I}$ which rotates the cloud of observations such that the principal axes of the data becomes aligned with coordinate axes. If the data is rotated in such a manner, the variances are easily obtained along each axis, and the covariances will be zero. We denote the matrix of rotated variables $\mathbf{Z}$ with $\mathbf{Z} = \mathbf{XB}$. The diagonal variance-covariance matrix of $\mathbf{Z}$ is $(n-1)^{-1}\mathbf{Z}^T\mathbf{Z} = (n-1)^{-1}\mathbf{B}^T\mathbf{X}^T\mathbf{X}\mathbf{B} = \mathbf{B}^T\mathbf{\Sigma}\mathbf{B}$. Regarding the first component alone, we wish to maximize the variance along this direction. This can be formulated

$$\arg\max_{\mathbf{b}} \mathbf{b}^T\mathbf{\Sigma}\mathbf{b} \qquad \text{subject to} \qquad \mathbf{b}^T\mathbf{b} = 1. \tag{4.1}$$

This problem is easily solved by incorporating the constraint using a single Lagrange multiplier $\alpha \neq 0$,

$$\arg\max_{\mathbf{b}} \mathbf{b}^T\mathbf{\Sigma}\mathbf{b} - \alpha(\mathbf{b}^T\mathbf{b} - 1) \tag{4.2}$$

Differentiating this expression, setting to zero and rearranging gives,

$$\Sigma\mathbf{b} = \alpha\mathbf{b} \tag{4.3}$$

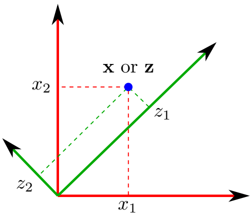
This is recognized as an eigenvalue problem. The criterion function in Equation 4.1 is maximized for $\mathbf{b}$ equal to the eigenvector of $\mathbf{\Sigma}$ corresponding to the largest eigenvalue α. The variance along this direction is $\mathbf{b}^T\Sigma\mathbf{b} = \mathbf{b}^T\alpha\mathbf{b} = \alpha$. Regarding the second component, we maximize the variance along the second direction subject to being orthogonal to the first. In general for the i^{th} component we have,

$$\arg\max_{\mathbf{b}_i} \mathbf{b}_i^T\mathbf{\Sigma}\mathbf{b}_i \qquad \text{subject to} \qquad \mathbf{b}_i^T\mathbf{b}_i = 1, \quad \mathbf{b}_i^T\mathbf{b}_j = 0, \quad j = 1\ldots i-1 \tag{4.4}$$

The solution to this problem can be obtained by factoring out the variance explained by earlier components, and then solving Equation 4.1 using the resulting data matrix. The solution is exactly given by the i^{th} eigenvector of $\mathbf{\Sigma}$, with the variance explained by the i^{th} eigenvalue. In summary, PCA can be performed through an eigenanalysis of the variance-covariance matrix where the eigenvectors are the principal axes and the eigenvalues are the variances explained along each direction. The $(p \times p)$ matrix $\mathbf{B}$ of eigenvectors is the rotation matrix discussed above. By defining a threshold on the variances α_i, we can reduce this matrix to size $(p \times k)$ giving the following system of equations,

$$\underset{n\times k}{\mathbf{Z}} = \underset{n\times p}{\mathbf{X}}\,\underset{p\times k}{\mathbf{B}}\,. \tag{4.5}$$

The dimensionality reduction is evident here, $\mathbf{Z}$ will be much smaller than $\mathbf{X}$ if $k \ll p$. The matrix $\mathbf{Z}$ is called the *scores matrix.* Equation 4.5 can be viewed as a system of linear equations. Each PC (column of $\mathbf{Z}$) is a linear combination of the variables in $\mathbf{X}$. The coefficients of the linear combination pertaining to the i^{th} PC are given by the i^{th} column of $\mathbf{B}$. For this reason, the matrix $\mathbf{B}$ describing the principal axes is called the *loading matrix* with individual coefficients known as *loadings* while the elements of $\mathbf{Z}$ are called *scores*, measuring the position of the observations on the derived axes in $\mathbf{B}$. Figure 4.2 motivates these definitions.



$$\mathbf{z} = \mathbf{x}\mathbf{B} \Leftrightarrow [3.5 \quad 0.71] = [2 \quad 3] \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Figure 4.2: Graphical explanation of the PCA notation. The single observation regarded here is called $\mathbf{x} = [2 \quad 3]$ in the original coordinate system and $\mathbf{z} = [3.5 \quad 0.71]$ when projected onto the principal axes. The axes are assumed to be tilted at 45 degrees, resulting in the loading (rotation) matrix shown.

PCA can be calculated using a singular value decomposition (SVD) of the data matrix $\mathbf{X}$. The (economy size) SVD of $\mathbf{X}$ yields matrices $\mathbf{U}$ ($(n \times k)$, orthogonal columns), $\mathbf{D}$ ($(k \times k)$, diagonal), and $\mathbf{V}$ ($(p \times k)$ orthogonal columns) with $k = \text{rank}(\mathbf{X})$ such that

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T. \tag{4.6}$$

Using $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, we can write the variance-covariance matrix $\boldsymbol{\Sigma} = (n-1)^{-1}\mathbf{V}\mathbf{D}^2\mathbf{V}^T$. Multiplying each side of this expression by $\mathbf{V}$ from the right, we get,

$$\boldsymbol{\Sigma}\mathbf{V} = \mathbf{V}\frac{\mathbf{D}^2}{n-1}, \tag{4.7}$$

representing the complete set of eigenvalue-eigenvector pairs of $\boldsymbol{\Sigma}$. Evidently, through an SVD of $\mathbf{X}$ we get $\mathbf{B} = \mathbf{V}$ and $\boldsymbol{\alpha} = (n-1)^{-1}\text{diag}(\mathbf{D}^2)$.

To warm up for coming sections, we give an alternative formulation of PCA in terms of fitting a linear manifold to the data as presented by Hastie et al. [59]. Consider a hyperplane spanned by $k \leq p$ orthogonal vectors in $\mathbb{R}^p$,

$$f(\boldsymbol{\mu}, \mathbf{w}) = \boldsymbol{\mu} + \mathbf{A}\mathbf{w}. \tag{4.8}$$

We choose the hyperplane parameters $\boldsymbol{\mu}$ (intercept) and $\mathbf{A}$ (orientation) such that, using an appropriate vector $\mathbf{w}_i$ of coefficients for each point, the sum of squared distances from the observations in $\mathbf{X}$ to $f(\boldsymbol{\mu}, \mathbf{w})$ is minimized,

$$\arg\min_{\boldsymbol{\mu},\mathbf{A},\mathbf{W}} \sum_{i=1}^{n} \|\mathbf{x}_i^T - (\boldsymbol{\mu} + \mathbf{A}\mathbf{w}_i)\|^2 \qquad \text{subject to} \qquad \mathbf{A}^T\mathbf{A} = \mathbf{I} \tag{4.9}$$

with $\mathbf{x}_i$ being the i^{th} observation (row) of $\mathbf{X}$. Differentiating with respect to $\mathbf{w}_i$ and $\boldsymbol{\mu}$, setting to zero and rearranging gives

$$\boldsymbol{\mu} = \bar{\mathbf{x}}^T \tag{4.10}$$

$$\mathbf{w}_i = \mathbf{A}^T(\mathbf{x}_i - \bar{\mathbf{x}})^T, \tag{4.11}$$

where $\bar{\mathbf{x}}$ is the average of all observations in $\mathbf{X}$. Since $\mathbf{X}$ is mean centered, $\bar{\mathbf{x}} = \mathbf{0}$. The problem reduces to one of finding the optimal basis $\mathbf{A}$,

$$\arg\min_{\mathbf{A}} \sum_{i=1}^{n} \|\mathbf{x}_i^T - \mathbf{A}\mathbf{A}^T\mathbf{x}_i^T)\|^2 \qquad \text{subject to} \qquad \mathbf{A}^T\mathbf{A} = \mathbf{I} \tag{4.12}$$

A standard result of the SVD is that the expression $\mathbf{V}\mathbf{V}^T\mathbf{x}_i^T$ with $\mathbf{V}$ from Equation 4.6 provides the best rank(k) approximation to $\mathbf{x}_i$, which is exactly what is sought in Equation 4.12. From this, we conclude that $\mathbf{A} = \mathbf{V} = \mathbf{B}$, the principal axes.

In summary, PCA finds an orthogonal rotation (loading) matrix $\mathbf{B}$ which is used to rotate the coordinates of the data in $\mathbf{X}$ such that the resulting data set (scores matrix) $\mathbf{Z}$ has orthogonal variables (columns). In the following, we will study augmented, or regularized, variants of PCA. Standard PCA is the only transformation of $\mathbf{X}$ where the new coordinate system is orthogonal and where the data projected onto these variables has uncorrelated variables [72]. Any modification of PCA must therefore surrender at least one of these properties.

4.1 Sparse Principal Component Analysis

A drawback of PCA is that the all loadings of the matrix $\mathbf{B}$ are typically non-zero, meaning that each new variable (column of $\mathbf{Z}$) is a linear combination of *all* original variables in $\mathbf{X}$. This makes interpretation of the principal components difficult. If we wish to try and understand the behavioral patterns of the present data set as discussed above, we require each PC to be dependent on a limited set of variables. This is the goal of *sparse* PCA (SPCA), to approximate the properties of PCA with a constraint on the *cardinality*[1] of each loading vector (column of $\mathbf{B}$). Estimating the leading[2] such loading vector amounts to the maximization problem

$$\arg\max_{\mathbf{b}} \mathbf{b}^T\mathbf{\Sigma}\mathbf{b} \qquad \text{subject to} \qquad \mathbf{b}^T\mathbf{b} = 1, \quad \text{card}(\mathbf{b}) \leq m, \tag{4.13}$$

where card($\mathbf{b}$) is the cardinality of $\mathbf{b}$ and m is an upper bound on the number of non-zero elements. Solving this non-convex optimization problem is provably difficult (NP-hard) and must therefore be regularized in some manner [31, 99]. Below, we will review methods that give approximate solutions to this problem.

[1] The cardinality of a vector is here taken to represent the number of non-zero elements.

[2] The leading loading vector is the eigenvector corresponding to the largest eigenvalue.

4.1.1 Estimation using Truncation

The simplest way to obtain sparse loading vectors is to interpret the results of standard PCA as *approximately* sparse and enforce sparsity by setting sufficiently small loadings to zero. To achieve a certain upper cardinality bound m, it may be necessary to set loadings with significant magnitude to zero. Remaining loadings are unaltered. This augmentation results both in non-orthogonal loading vectors and correlated principal components. Also, after truncation, there is almost certainly some adjustment of the remaining non-zero loadings that will lead to a better solution. Such an adjustment is, however, usually not considered for this method. Cadima and Jolliffe [17] show that truncation leads to solutions far from the optimal value, discouraging use of this method for SPCA.

4.1.2 Direct Estimation using the Elastic Net

To improve on the truncation method, we seek an approach that is able to adjust the non-zero loadings as other loadings are forced to zero. In Chapter 2, we reviewed several such methods for regression. It turns out that we can use these for the estimation of sparse principal components. Consider the following approximation of a PC using ridge regression,

$$\mathbf{b}_{\text{ridge}} = \arg\min_{\mathbf{b}} \|\mathbf{z}_i - \mathbf{X}\mathbf{b}\|^2 + \lambda\|\mathbf{b}\|^2, \tag{4.14}$$

where we have replaced the response variable with the i^{th} PC. We seek the loading vector b that best approximates this variable under the coefficient shrinkage of the penalty term. An approximation $\hat{\mathbf{z}}_i$ of the i^{th} PC can be obtained using the SVD in the manner of Equation 2.40 with $\mathbf{y} = \mathbf{z}_i = \mathbf{X}\mathbf{v}_i = \mathbf{u}_i d_{ii}$,

$$\mathbf{b}_{\text{ridge}} = \mathbf{V}(\mathbf{D}^2 + \lambda\mathbf{I})^{-1}\mathbf{D}\mathbf{U}^T\mathbf{u}_i d_{ii} = \mathbf{v}_i \frac{d_{ii}^2}{d_{ii}^2 + \lambda.} \tag{4.15}$$

This shows that the optimal coefficient vector $\mathbf{b}_{\text{ridge}}$ is a scaled version of the standard PCA loading vector $\mathbf{v}_i$. If we normalize this result to unit length, we get the exact PCA solution. For $\lambda = 0$, we get the same solution, but this requires $n > p$. The purpose of the ridge penalty is to fix the solution also in cases where $p > n$. Choosing any value of $\lambda > 0$ always ensures a solution.

Having formulated PCA in this manner, it is straight-forward to impose sparsity via the ℓ_1 (LASSO) penalty, yielding an Elastic Net regression problem (cf. Section 2.10),

$$\mathbf{b}_{\text{SPCA}} = \arg\min_{\mathbf{b}} \|\mathbf{z}_i - \mathbf{X}\mathbf{b}\|^2 + \lambda\|\mathbf{b}\|^2 + \delta\|\mathbf{b}\|_1. \tag{4.16}$$

The additional constraint drives some loadings to exactly zero, while others are adjusted to reconstruct the standard PC as well as possible. With normalization of the results, we get the i^{th} original PC for $\delta = 0$ and increasingly sparse representations with growing values of δ. This problem is no longer independent of the value of λ, but solutions have been shown to be insensitive to this parameter [134, 177].

4.1.3 Estimation using the SPCA Criterion

A drawback of the method presented in the previous section is that the results are heavily guided by regular PCA. Instead of approximating the actual loadings, we would prefer an estimation criterion for SPCA that approximates the *properties* of PCA while imposing sparsity. These properties are successive maximization of variance, orthogonality of principal axes and uncorrelatedness of the principal components. Building on the Elastic Net procedure from the previous section, we wish to find a self-contained expression that does not rely on PCA.

The derivation starts with the alternative formulation of PCA in Equation 4.12. It is possible to prove that this expression can be relaxed by changing the term $\mathbf{A}\mathbf{A}^T\mathbf{x}_i^T$ to $\mathbf{A}\mathbf{B}^T\mathbf{x}_i^T$ where $\mathbf{B}$ is an arbitrary $(p \times k)$ matrix, and still get the original principal axes at the optimum *if* the problem is regularized using a ridge regression-type term. The new formulation of PCA becomes,

$$\arg\min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^{n} \|\mathbf{x}_i^T - \mathbf{A}\mathbf{B}^T\mathbf{x}_i^T)\|^2 + \lambda \sum_{j=1}^{k} \|\mathbf{b}_j\|^2 \qquad \text{subject to} \qquad \mathbf{A}^T\mathbf{A} = \mathbf{I}. \tag{4.17}$$

In other words, at the optimum we have $\mathbf{A} = \mathbf{B}$, the loading matrix of PCA.

As in the previous section, a second constraint on the ℓ_1-norm of the loading vectors is now added to obtain sparse solutions,

$$\arg\min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^{n} \|\mathbf{x}_i^T - \mathbf{A}\mathbf{B}^T\mathbf{x}_i^T)\|^2 + \lambda \sum_{j=1}^{k} \|\mathbf{b}_j\|^2 + \delta \sum_{j=1}^{k} \|\mathbf{b}_j\|_1$$
$$\text{subject to} \qquad \mathbf{A}^T\mathbf{A} = \mathbf{I}. \tag{4.18}$$

This formulation of SPCA is called the *SPCA criterion.* It is seen that while orthogonality is not imposed on $\mathbf{B}$, its columns will generally exhibit limited deviance from orthogonality since $\mathbf{B}$ can be seen as a regularized non-square inverse of $\mathbf{A}$, where $\mathbf{A}$ is orthogonal due to the extra constraint.

4.1.3.1 Computation

The remaining question is how to solve the SPCA criterion to obtain an estimate of the sparse loading matrix $\mathbf{B}$. To obtain $\mathbf{B}$, we must also determine $\mathbf{A}$ yielding a difficult high-dimensional optimization problem. To simplify the process, an alternating optimization scheme is employed. Assume $\mathbf{A}$ is known and fixed, then (after a fair bit of algebra) it is possible to show that the estimation of $\mathbf{B}$ amounts to solving k independent Elastic Net problems,

$$\arg\min_{\mathbf{b}_i} \|\mathbf{X}\mathbf{a}_i - \mathbf{X}\mathbf{b}_i)\|^2 + \lambda\|\mathbf{b}_i\|^2 + \delta\|\mathbf{b}_i\|_1, \qquad i = 1 \dots k. \tag{4.19}$$

The level of sparsity is chosen at this stage. Two approaches apply here, one where a value of δ is chosen for either each component separately or one value for all vectors. The alternative approach is to specify the cardinality m of the solution. This is easily implemented since it amounts to stopping the Elastic Net path algorithm as soon as the active set contains m variables (cf. Section 2.10).

If $\mathbf{B}$ is considered fixed and we wish to estimate $\mathbf{A}$, the penalties in Equation 4.18 amount to a simple translation and can therefore be omitted. The remaining optimization problem is,

$$\arg\min_{\mathbf{A}} \sum_{i=1}^{n} \|\mathbf{x}_i^T - \mathbf{A}\mathbf{B}^T\mathbf{x}_i^T)\|^2 \qquad \text{subject to} \qquad \mathbf{A}^T\mathbf{A} = \mathbf{I}. \tag{4.20}$$

The solution to this problem can be obtained by computing the SVD of the matrix $\mathbf{X}^T\mathbf{X}\mathbf{B}$ such that,

$$\mathbf{X}^T\mathbf{X}\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^T \quad \Rightarrow \quad \mathbf{A} = \mathbf{U}\mathbf{V}^T. \tag{4.21}$$

Again, we omit the proof which is detailed in [177]. The iterative scheme proceeds by alternately estimating $\mathbf{A}$ and $\mathbf{B}$ until convergence. The entire SPCA algorithm is given in Algorithm 4.1.

Algorithm 4.1 Sparse Principal Component Analysis

```
Initialize $\mathbf{A}$ to the $k$ leading principal axes of standard PCA.
while not converged do
    Given $\mathbf{A}$, estimate a suitably sparse loading matrix $\mathbf{B}$ by solving $k$ Elastic
    Net regressions according to Equation 4.19.
    Normalize the loading vectors (columns) in $\mathbf{B}$ to unit length.
    Using the obtained matrix $\mathbf{B}$, calculate $\mathbf{A}$ using Equation 4.21.
end while
```

4.1.4 Bounds and Optimality

There are several interesting results with respect to SPCA which help to improve and understand the quality of obtained solutions. We will mention a few of these here, discussed in detail by Moghaddam et al. [99]. In this section, we assume that the eigenvalues are sorted in ascending order, i.e. $\alpha_{\min} = \alpha_1$ and $\alpha_{\max} = \alpha_p$.

We focus on two questions:

1. For a certain cardinality m and distribution of the non-zero coefficients in a length-p loading vector, what is the maximal variance that can be explained and what are the corresponding loadings?
2. Likewise, what is the smallest variance that can be obtained?

Central to investigating these issues is the Rayleigh-Ritz theorem which we review below for real matrices,

THEOREM 4.1 *Let* $\mathbf{M} \in \mathbb{R}^{p\times p}$ *be a symmetric matrix. Then the* Rayleigh quotient

$$R(\mathbf{x}) = \frac{\mathbf{x}^T\mathbf{M}\mathbf{x}}{\mathbf{x}^T\mathbf{x}} \tag{4.22}$$

has critical points[3] *equal to the eigenvectors of* $\mathbf{M}$ *and* critical values *equal to the corresponding eigenvalues.*

A consequence of this theorem is that the maximal and minimal eigenvalues of $\mathbf{M}$ correspond to the global maximum and minimum of the Rayleigh quotient respectively.

Returning to the formulation of PCA from Equation 4.1 we see that the criterion function indeed is a Rayleigh quotient given the constraint.

To answer the first question, assume we knew the correct set of non-zero variables of the sparse leading loading vector $\mathbf{x}$, and assign indices corresponding to these variables to the set $\mathcal{A}$. Then the Rayleigh quotient from the formulation of PCA can be reduced using the following subvectors and submatrices,

$$R(\mathbf{b}) = \mathbf{b}_{\mathcal{A}}^T\mathbf{\Sigma}_{\mathcal{A},\mathcal{A}}\mathbf{b}_{\mathcal{A}} \qquad \text{subject to} \qquad \mathbf{b}_{\mathcal{A}}^T\mathbf{b}_{\mathcal{A}} = 1 \tag{4.23}$$

[3] A function $f(\mathbf{x})$ has critical points at all points $\mathbf{x}_0$ where $\nabla f(\mathbf{x}_0) = 0$ or $f(\mathbf{x}_0)$ is not differentiable.

The Rayleigh-Ritz theorem tells us that the maximum of this "quotient" is given by the maximal eigenvalue at the critical point given by the corresponding eigenvector. Quite simply, this reveals the optimal solution given the true set of non-zero indices $\mathcal{A}$. It also suggests a simple all subsets-type algorithm (cf. Section 2.7) for finding the best possible sparse leading loading vector of cardinality m, when the number of dimensions is limited. By exhaustively trying all $p!/(m!(p-m)!)$ combinations of m non-zero variables, the optimal solution is given for the combination with the largest eigenvalue. When the leading loading vector is found, the variance explained along this direction can be factored out from the data set, and the process is repeated to find the second loading vector. Factoring out the contribution along a pricipal axis can either be done through the orthogonalization process described in the Gram-Schmidt Algorithm 2.1 or equivalently, by downdating $\mathbf{\Sigma}$ according to,

$$\mathbf{\Sigma} = \mathbf{\Sigma} - \alpha \mathbf{b}\mathbf{b}^T, \tag{4.24}$$

where α and $\mathbf{b}$ are the optimal eigenvalue and eigenvector respectively. Cases where the number of variables make an all subsets approach infeasible also benefit from these results. Given a pattern $\mathcal{A}$ of non-zero indices, be it through simple truncation or the SPCA criterion, we can always adjust the loadings of each principal axis to achieve an increase in explained variance by finding the leading eigenvector of the submatrix $\mathbf{\Sigma}_{\mathcal{A},\mathcal{A}}$.

The Rayleigh-Ritz theorem shows that no vector $\mathbf{x}$, sparse of full, can lead to a larger value of $R(\mathbf{x})$ than the largest eigenvalue of $\mathbf{M}$. This provides a reference upper bound suitable for comparing SPCA solutions. Further, we have

$$\alpha_i(\mathbf{\Sigma}) \quad \leq \quad \alpha_i(\mathbf{\Sigma}_{\mathcal{A},\mathcal{A}}) \quad \leq \quad \alpha_{i+p-m}(\mathbf{\Sigma}), \tag{4.25}$$

that is, the eigenvalues of any submatrix of $\mathbf{\Sigma}$ are bounded by the eigenvalues of $\mathbf{\Sigma}$. In particular, this means that $\alpha_m(\mathbf{\Sigma})$ is a lower bound for the leading eigenvalue of the submatrix $\mathbf{\Sigma}_{\mathcal{A},\mathcal{A}}$, providing an answer to question two above. This bound is an important aid in selecting a suitable cardinality for the loading vectors. By studying the eigenvalue spectrum of the full variance-covariance matrix, we can select the cardinality that guarantees, at minimum, a certain amount of variance, e.g. 70%. The recipe for this process is as follows. Plot the eigenvalues of $\mathbf{\Sigma}$ in ascending order normalized by the largest eigenvalue. Find the index of the smallest eigenvalue larger than the desired fraction of variance. The index is the cardinality that guarantees this fraction.

4.2 Application

In this section we present results on the small diabetes data set from Chapter 2. Performing PCA or variants thereof on the data matrix of this data set may reveal relations between variables that give new clinical insight, or may be used to simplify a subsequent regression analysis by deriving a small set of descriptive variables. For interpretation, there is a clear benefit of using sparse PCA in the analysis, as the interpretation of combinations of e.g. four variables is far easier than regarding the full set of ten variables. At $p = 10$, this data set is sufficiently small to lend itself to the all subsets approach to find the optimal leading loading vector for a given cardinality. Interestingly, we will see that when more than one loading vector is estimated, this method is no longer optimal in terms of unique, or *adjusted*, variance, a term which will be explained below.

Figure 4.3 shows the amount of variance explained by the leading loading vector for each method. Also shown is the eigenspectrum of $\mathbf{\Sigma}$ (regular PCA), forming a lower bound for each choice of cardinality. For this data set, this bound is rather loose and does not help much in choosing a suitable level of sparsity.

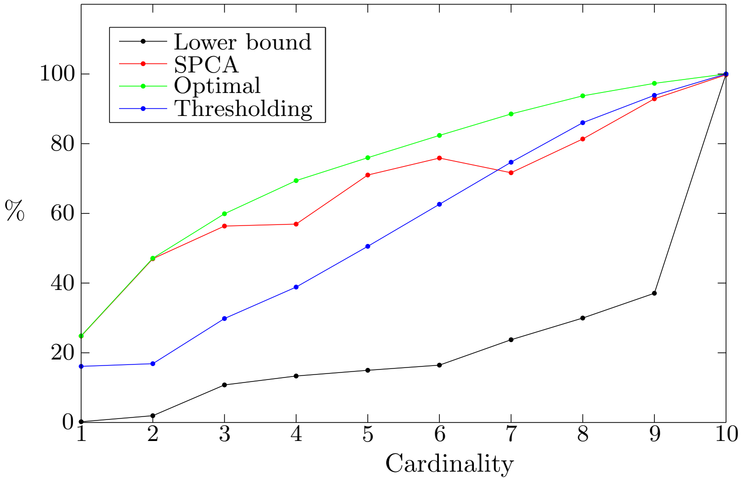


Figure 4.3: Variance of the leading loading method for the optimal all subsets method, the SPCA criterion and simple thresholding. The ordinate measures fractions (%) of the upper bound α_{10} of PCA. The black line shows the lower bound for each choice of cardinality; regardless of the method being used, variances cannot fall short of these values.

Table 4.1 gives the variance (α) explained by each principal component and the loading matrix $\mathbf{B}$. All coefficients are non-zero and interpretation of the linear combinations that govern each principal component is difficult. In Table 4.2, we show the sparse loading matrix and corresponding variances of the all subsets procedure with cardinality $m = 4$. The loading vectors created using this approach are not orthogonal. While all eigenvectors of a single submatrix of $\mathbf{\Sigma}$ are orthogonal, the eigenvectors of different submatrices are not. The correlation between loading vectors can be quite significant. As a result, the variance explained along one direction is partly present along other directions. To obtain a fair estimate of the unique amount of variance explained along each direction we employ a Gram-Schmidt-type procedure. The first direction is taken to be the one that explains the most (unadjusted) variance. The presence of this direction is then factored out from other directions, and the process is repeated until an ordering and a set of adjusted variances are obtained. See Chapter 6 for more details on this procedure. The variances shown in Table 4.2 and Table 4.3 are adjusted in this fashion. After adjustment, the all subsets procedure is no longer certain to be optimal, except for the leading eigenvector which remains unadjusted. This leads to the suspicion that methods that explicitly seek near-orthogonal directions, such as with the SPCA criterion, may achieve better performance. The variances in Table 4.3 suggest that this may be true. Figure 4.4 investigates this question more carefully. Here, the total amount of variance explained by all ten loading vectors is plotted as a function of cardinality for each method. It is seen that for very sparse solutions, the SPCA criterion outperforms the all subsets procedure for this data set. It is also seen that simple thresholding does quite well for high cardinalities.

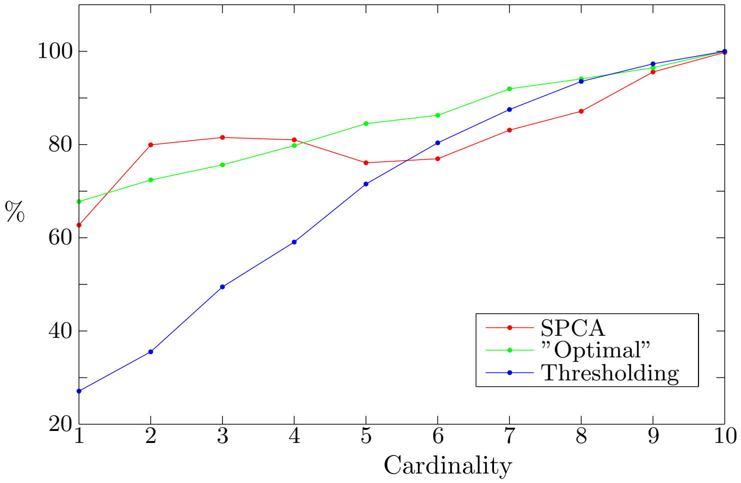


Figure 4.4: Total variance explained by all ten loading vectors for each method and cardinality.

4.3 Varimax Rotated Principal Components

Previous methods for SPCA in this chapter produce strictly sparse solutions. If approximately sparse loading matrices are considered, it is possible to find orthogonal loading (rotation) matrices that achieve this. Rearranging Equa-

α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}
0.085	0.78	4.3	5.3	6.0	6.6	9.5	12	14	40

$\mathbf{b}_1$	$\mathbf{b}_2$	$\mathbf{b}_3$	$\mathbf{b}_4$	$\mathbf{b}_5$	$\mathbf{b}_6$	$\mathbf{b}_7$	$\mathbf{b}_8$	$\mathbf{b}_9$	$\mathbf{b}_{10}$
-0.21	0.044	0.49	-0.41	-0.68	0.22	-0.10	0.014	-0.0081	-0.0033
-0.18	-0.38	-0.10	-0.67	0.37	-0.041	-0.067	0.44	0.0021	-0.0037
-0.30	-0.15	0.16	0.49	0.12	0.40	-0.51	0.39	-0.042	-0.0082
-0.27	-0.13	0.51	-0.019	0.48	0.27	0.32	-0.47	-0.027	0.0032
-0.34	0.57	-0.068	-0.068	0.12	-0.0054	0.073	0.12	0.042	-0.70
-0.35	0.45	-0.26	-0.16	0.11	0.13	-0.23	-0.19	0.35	0.56
0.28	0.50	0.38	-0.076	0.24	-0.10	-0.0075	0.32	-0.48	0.31
-0.42	-0.068	-0.38	0.0079	-0.14	0.033	0.071	-0.18	-0.77	0.090
-0.37	-0.026	0.063	0.26	-0.15	-0.17	0.64	0.44	0.18	0.26
-0.32	-0.084	0.27	0.087	0.031	-0.80	-0.35	-0.16	0.015	-0.0026

Table 4.1: Variance (% of total variation) along each direction (top) and the corresponding loading vectors (bottom) for standard PCA.

α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}
0.051	0.22	3.0	4.7	4.9	6	7	11	13	27

$\mathbf{b}_1$	$\mathbf{b}_2$	$\mathbf{b}_3$	$\mathbf{b}_4$	$\mathbf{b}_5$	$\mathbf{b}_6$	$\mathbf{b}_7$	$\mathbf{b}_8$	$\mathbf{b}_9$	$\mathbf{b}_{10}$
0	0	-0.51	0	-0.76	0	0	0.40	0	0.26
0	0.81	0	0	0	-0.34	0	0.34	0.43	0
0	0	0	0.50	0	0	0.67	0.59	0	0
0	0.32	-0.59	0	0	0	0.50	0	-0.29	-0.28
0.53	0	0	0	-0.30	-0.39	0.38	0	0	0
0.52	0	0	0	-0.38	0	0.37	0	0.65	0
0	-0.39	0	-0.56	0	-0.47	0	0	0	-0.71
0.50	0.28	0	0.44	0	0	0	0	0	0
0.42	0	-0.42	0	0.41	0	0	0.60	0	0.58
0	0	-0.43	0.47	0	-0.71	0	0	0.53	0

Table 4.2: Adjusted variances (% of total variation) (top) and loading matrix (bottom) for the all subsets method.

α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}
0.0002	0.10	3.7	4.1	6.1	8.0	8.7	11	16	22

$\mathbf{b}_1$	$\mathbf{b}_2$	$\mathbf{b}_3$	$\mathbf{b}_4$	$\mathbf{b}_5$	$\mathbf{b}_6$	$\mathbf{b}_7$	$\mathbf{b}_8$	$\mathbf{b}_9$	$\mathbf{b}_{10}$
0	0	0	0	-0.98	-0.16	0	0	0	0
0	0	0	-0.96	0	0	0	0	0	0
0	0.44	0.22	0	0	0	0	0.97	-0.085	-0.075
0	0	0.93	-0.045	-0.17	0	0	0	0	0
0.65	0	0	0	-0.022	-0.081	0.38	0	0	0
0.69	0	0	0	0	0	0.0044	0	0	0
0	-0.73	0	0.22	0	0	0	0	0.08	0.79
0.28	0.51	0	-0.13	0	0	0.26	0.027	-0.45	-0.56
0	0.092	0.18	0	-0.013	-0.27	0.88	0.16	0	-0.20
0.0070	0	0.19	0	0	-0.94	0	0.12	-0.88	0

Table 4.3: Adjusted variances (% of total variation) (top) and loading matrix (bottom) resulting from estimation via the SPCA criterion.

tion 4.5, (with $k = p$), we have the following expression for the data matrix,

$$\mathbf{X} = \mathbf{Z}\mathbf{B}^T. \tag{4.26}$$

Just as each PC is a linear combination of the original variables, each original variable is a linear combination of the principal components. However, the expression

$$\mathbf{X} = \mathbf{Z}\mathbf{R}^T\mathbf{R}\mathbf{B}^T = \tilde{\mathbf{Z}}\tilde{\mathbf{B}}^T, \tag{4.27}$$

where $\mathbf{R}$ is any $(p \times p)$ orthogonal is an equivalent expression for $\mathbf{X}$. Going back to the PCA formulation of Equation 4.5, we have,

$$\tilde{\mathbf{Z}} = \mathbf{X}\tilde{\mathbf{B}}^T = \mathbf{Z}\mathbf{R}^T = \mathbf{X}\mathbf{B}\mathbf{R}^T \tag{4.28}$$

Multiplication by the matrix $\mathbf{R}$ will rotate the coordinate system in $\mathbf{B}$ but the columns remain orthogonal. The columns of the scores matrix do, however, become correlated from this rotation, as a result of $\mathbf{Z}$ having columns in $\mathbb{R}^n$ while the rotation is in $\mathbb{R}^p$. Note again that PCA is the only transformation with both orthogonal loading vectors and uncorrelated scores.

The matrix $\mathbf{R}$ can be chosen such that the variance of the columns of $\tilde{\mathbf{B}}$ is maximized. To do this, it is beneficial to have some large loadings and some close to zero, rather than a more even distribution of loadings. Therefore, this criterion, called the *Varimax* criterion, leads to an approximately sparse loading matrix. Algorithms for estimating $\mathbf{R}$ for Varimax and other types of orthogonal rotations are given in Chapter 5, where we also give examples of their application to medical image analysis.

4.4 References

A comprehensive reference for principal component analysis and related methods is the book of Jolliffe [73]. A more detailed discussion on the curse of dimensionality is provided by Bellman [7] and by Hastie et al. [59].

The formulation of sparse PCA in Equation 4.13 is adopted from the work of d'Aspremont et al. [31] who proposed a convex relaxation of the cardinality constraint leading to an algorithm that can be solved using semidefinite programming. Jolliffe et al. [74] replace the cardinality constraint with a LASSO penalty on the loading vectors, driving some loadings to exactly zero. The resulting optimization problem is, however, computationally difficult to handle.

The direct approach using the Elastic Net and the SPCA criterion are both due to Zou et al. [177] who give detailed explanations of the algebra leading up to these formulations. It is also shown that if the regularization parameter λ is set to infinity, the Elastic Net procedure for estimating $\mathbf{B}$ reduces to a soft thresholding rule. We investigate the resulting, efficient, algorithm in Chapter 8. Zou et al. [177] also proposed the adjustment procedure of the variances obtained from non-orthogonal (oblique) rotation matrices. An equivalent method was independently proposed by Gervini and Rousson [50]. In Chapter 6 we propose a forward selection-type technique for a more reproducible variant of this approach.

The discussion on eigenvalue bounds and optimality is adopted from the work of Moghaddam et al. [99], who provide several other bounds, algorithms and theorems.

The component rotation technique briefly introduced in Section 4.3 and investigated more thoroughly in Chapter 5 is among the earliest methods for obtaining a loading matrix that is approximately sparse, and thus, simpler to interpret. The Varimax rotation is due to Kaiser [76] and belongs to the class of orthomax rotations, described by Harman [57].

Other approaches to sparse PCA have been put forth by e.g. Chennubhotla and Jepson [18], Vines [166], Hausman [61], and Rousson and Gasser [123].

Other methods for data decomposition and dimensionality reduction include independent component analysis [68], maximum autocorrelation factors [82, 150], and non-negative matrix factorization [89].

Part II

Contributions

Chapter 5

Sparse Modeling of Landmark and Texture Variability using the Orthomax Criterion

Mikkel B. Stegmann, Karl Sjöstrand and Rasmus Larsen

Abstract

In the past decade, statistical shape modeling has been widely popularized in the medical image analysis community. Predominantly, principal component analysis (PCA) has been employed to model biological shape variability. Here, a reparameterization with orthogonal basis vectors is obtained such that the variance of the input data is maximized. This property drives models toward *global* shape deformations and has been highly successful in fitting shape models to new images. However, recent literature has indicated that this uncorrelated basis may be suboptimal for exploratory analyses and disease characterization. This paper explores the orthomax class of statistical methods for transforming variable loadings into a *simple structure* which is more easily interpreted by favoring sparsity. Further, we introduce these transformations into a particular framework traditionally based on PCA; the Active Appearance Models (AAMs). We note that the orthomax transformations are independent of domain dimensionality (2D/3D etc.) and spatial structure. Decompositions of both shape and texture models are carried out. Further, the issue of component ordering is treated by establishing a set of relevant criteria. Experimental results are given on chest radiographs, magnetic resonance images of the brain, and face images. Since pathologies are typically spatially localized, either with respect to shape or texture, we anticipate many medical applications where sparse parameterizations are preferable to the conventional global PCA approach.

5.1 Introduction

Due to the frequent noisy and highly complex nature of many medical imaging modalities, constrained solutions are often required. One popular class of constrained image analysis is the various forms of shape models. Here, a top-down approach is taken to the localization of a structure in a medical image using an explicit model of the geometrical layout of the structure supplemented by a set of associated variation patterns. Combined, these two entities should optimally be able to represent the given variability of the structure and nothing apart from that. Hence, only valid solutions can be produced, provided that the model can be fitted with a sufficiently high likelihood. Further, in many applications it may be desirable to be able to extend the use of such models from the classic segmentation or registration scenario, to a level where the model parameterization possesses inherent *interpretive powers* where latent variables are expressed directly. An example of such is disease characterization by surrogate markers, see e.g. Mitchell et al. [98]

Decomposition of shape and texture variability is predominately carried out by principal component analysis (PCA), which produces a reparameterization with orthogonal basis vectors such that the variance of the input data is maximized. Although this basis is in many senses optimal, recent literature indicate that it might not posses a sufficiently expressive basis for some medical interpretation scenarios [148, 149, 161, 162]. Since PCA maximizes variance, new variables (i.e. the principal components) will typically affect the shape or texture globally. In turn, this may lead to confounding of effects due to the chance correlation inherent to limited medical data sets. Interestingly, it has been observed that independent component analysis (ICA) of shape produces new variables showing more localized effects, and thus being able to describe specific pathologies [148, 149, 161, 162]. We note that *localization* is often a desirable property for a basis aimed at explaining complex latent relations between pathology and geometry/texture. Consequently, it seems natural to promote transformations favoring locality directly, rather than indirectly. In this paper we explore the orthomax criterion for optimizing sparsity corresponding to new variables being associated to localized modes of variation.

5.2 Related Work

Although orthomax rotations are well-known within the statistical discipline *factor analysis*, this work is arguably among the first within medical image analysis to explore a method that directly optimizes sparsity. This can be seen as a natural continuation of the previously mentioned work on ICA shape modeling by Üzümcü et al. [161], Üzümcü et al. [162] and Suinesiaputra et al. [148, 149] in addition to the general literature on alternative parameterizations of shape.

Even for complex biological phenomena, principal component analysis typically yields a very good decomposition of shape variability in a cohort. However, significant non-linearities exist in some cases, which render the implicit assumption of a multivariate Gaussian distribution invalid. Thus, PCA models will yield a poor specificity, leading to potential synthesis of implausible shape configurations. Some of these problematic cases are designed synthetically to emphasize the limitations of a PDM, while others are demonstrating actual, real-world examples of shape variability with dominating non-linearities. Attempts to deal with such non-linearity include the polynomial regression PDM, PRPDM, by Sozou et al. [135]. Later, Sozou et al. [135] outperformed this using a back propagation neural network employing a multi-layer perceptron, which resulted in another xPDM acronym; the MLPPDM. A different approach is to employ a kernel-based density estimation of the shape distribution. This was proposed by Cootes and Taylor [22, 23] along with a computationally more attractive variant using a Gaussian mixture model to approximate the density function. Building on similar ideas, Heap and Hogg [62] proposed a hierarchical PDM, the HPDM, also based on multiple Gaussian models. Non-linear shape models are also treated in depth by Bowden [14].

Advances within machine learning that allow working implicitly in infinite dimensional spaces have also been utilized in shape modeling using kernel methods. By employing a variant of non-linear PCA called Kernel PCA (KPCA), complex non-linear shape distributions can be modeled. This was demonstrated on shapes from projections of varying-angle faces by Romdhani et al. [120] Further developments of this work were presented by Twining and Taylor [160] on synthetic shapes, and shapes from images of nematode worms.

While PCA decomposes variation by maximization of variance, other measures may also be of interest when a shape basis is to be chosen. For example, Larsen [81, 82] and Larsen et al. [84] chose to maximize autocorrelation along 2D shape contours using the maximum autocorrelation factors (MAF) due to Switzer [150]. Hilger et al. [63] later employed MAF as texture basis in Active Appearance Models (AAMs). The MAF approach was further extended to three-dimensional PDMs by Hilger et al. [64], Larsen and Hilger [83], and Larsen et al. [85]. Interestingly, it turns out that Molgedey-Schusters algorithm for performing ICA [101] is equivalent to MAF analysis, see Larsen et al. [84].

Turning to the specific use of the orthomax criterion, Ramsay and Silverman [116] give an instructive case study on varimax rotation of principal components based on one-dimensional temperature curves. Related to this is also the work by Peterson et al. [114] where two-dimensional contours of the brain structure *corpus callosum* were decomposed using PCA and subsequently rotated using the varimax criterion. While a similar corpus callosum case is presented here, the two papers are contrasted by the depth in which the rotation method is treated, and the depth in which the case study is analyzed, e.g. w.r.t. functional correlates such as IQ, handedness, et cetera.

Chennubhotla and Jepson [18] developed a *sparse PCA* method[1] bearing resemblance to the original varimax algorithm [76] by employing a sequence of bi-variate rotations. However, rather than optimizing variance, a function composed of the projected data and the basis vectors were investigated. This was carried out with a weight term controlling the transition from a PCA solution to a sparse solution. Examples were given on images, vector fields, and one-dimensional curves.

Regular PCA extracts new variables, the principal components, as linear combinations of the original variables. For interpretation purposes, the problem is that each new variable is a linear combination of *all* original variables. Sparse PCA aims at approximating the properties of regular PCA, while keeping the number of dependent variables, or equivalently, the number of non-zero loadings, small. Recently, Zou et al. [177] presented an algorithm for computing sparse loading matrices. It is heavily based on variable selection methods from regression analysis, primarily the elastic net [175]. Similarly to the SCoTLASS [74] method, a constraint is imposed on each loading vector, limiting the sum of absolute loadings. This drives some loadings to exactly zero, producing a sparse loading matrix in the strict sense. Results are given for the classic "pit-props" data set, some simulated data, and the Ramaswamy microarray data set. Results on medical shape data can be found in Sjöstrand et al. [134].

[1] This method is different from the sparse PCA method by Zou, Hastie and Tibshirani described below.

5.3 Methods

5.3.1 Principal Component Analysis

Consider a set of n vectors $\{\mathbf{x}_i\}_{i=1}^n \in \mathbb{R}^p$ having sample dispersion matrix $\mathbf{\Sigma_x}$. These could denote shape given by landmarks, or texture given by image intensities. Principal component analysis (PCA) transforms these vectors into a decorrelated basis $\mathbf{b}$ with dispersion matrix $\mathbf{\Sigma_b} = \mathrm{diag}(\boldsymbol{\lambda})$ by $\mathbf{b} = \mathbf{\Phi}^T(\mathbf{x} - \bar{\mathbf{x}})$, where $\mathbf{\Phi}$ denotes the eigenvector solution to $\mathbf{\Sigma_x \Phi} = \mathbf{\Phi \Sigma_b}$ and $\bar{\mathbf{x}}$ denotes the sample mean. Each eigenvector holds a variation pattern referred to as a deformation mode, where each of the original p variables is *loaded* by a given amount. Consequently, the terms eigenvectors, deformation modes, and variable loadings will be used interchangeably in the following. Let eigenvalues, λ_i, and corresponding eigenvectors be ordered so that $\lambda_1 \geq \cdots \geq \lambda_n = 0$ (when $n < p$). The deformation modes given by the higher order part of $\mathbf{b}$ are typically discarded by a variance-based criterion retaining e.g. 95% of trace($\mathbf{\Sigma_b}$) in k modes. A new example in $\mathbb{R}^p$ given by $\mathbf{b}$ can now be synthesized by the projection $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{\Phi b}$. Examples of using this generative property of PCA for image interpretation include inter-point distance models [25] and the later point distribution models (PDMs) [26].

5.3.2 Sparse Modeling Using the Orthomax Criterion

Orthomax rotations of a principal component basis reintroduce component correlation to obtain a *simple structure* of the final basis. Let $\mathbf{\Phi}$ be a $p \times k$ orthonormal matrix (of column eigenvectors) and $\mathbf{R}$ be an orthonormal rotation matrix in $\mathbb{R}^k$, i.e. $\mathbf{R}^T\mathbf{R} = \mathbf{I}_k$, where $\mathbf{I}_k$ denotes the $k \times k$ identity matrix. Further, let $\mathbf{R}_{ij}$ denote the scalar element in the i^{th} row and j^{th} column in matrix $\mathbf{R}$. The class of orthomax rotations can now be defined as

$$\mathbf{R}_{orthomax} = \arg\max_{\mathbf{R}} \left(\sum_{j=1}^{k} \sum_{i=1}^{p} (\mathbf{\Phi R})_{ij}^4 - \frac{\gamma}{p} \sum_{j=1}^{k} \left(\sum_{i=1}^{p} (\mathbf{\Phi R})_{ij}^2 \right)^2 \right), \tag{5.1}$$

where $\mathbf{R}_{orthomax}$ denotes the resulting rotation and γ denotes the type. This paper investigates $\gamma = 1$ (varimax[76]) and $\gamma = 0$ (quartimax, e.g. [57]). Further rotations include: $\gamma = k/2$ (equamax), and $\gamma = p(k-1)/(p+k-2)$ (parsimax). Orthomax rotations are traditionally computed using a sequence of bi-variate rotations [57, 76]. However, since varimax and quartimax are the only cases treated here, this work employ an iterative method based on singular value decomposition (SVD) for solving Equation 5.1, which is given in Algorithm 5.1. Notice that this returns the rotated basis, rather than $\mathbf{R}_{orthomax}$. The algorithm is also employed in the statistical language R and the computational system Matlab. It was first described by Horst [66] and independently shortly after in a different – albeit equivalent [155] – formulation by Sherin [132]. The relation between Equation 5.1 and Algorithm 5.1 is detailed in Section 5.4.

Algorithm 5.1 Estimation of Orthomax Rotation for $\gamma \in [0; 1]$

```
Require: $\mathbf{\Phi} \in \mathbb{R}^{p \times k}$, $\gamma$, $q$, $tol$, Diag(·) (sets off-diagonal elements to zero), $\circ$ Hadamard (element-wise) product
$\mathbf{R} = \mathbf{I}_k$
$d = 0$
for $i = 1$ to $q$ do
    $d_{old} = d$
    $\mathbf{\Lambda} = \mathbf{\Phi R}$
    $[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \text{svd}(\ \mathbf{\Phi}^T(\mathbf{\Lambda} \circ \mathbf{\Lambda} \circ \mathbf{\Lambda} - \frac{\gamma}{p}\mathbf{\Lambda} \cdot \text{Diag}(\mathbf{\Lambda}^T\mathbf{\Lambda}))\ )$
    $\mathbf{R} = \mathbf{U}\mathbf{V}^T$
    $d = \text{trace}(\mathbf{S})$
    if $d/d_{old} < tol$ then
        break
    end if
end for
$\mathbf{\Lambda} = \mathbf{\Phi R}$
return $\mathbf{\Lambda}$
```

Let us investigate the varimax variation a bit more closely. Let $\mathbf{\Lambda}$ denote the orthomax-rotated basis, $\mathbf{\Phi R}$, and let $\bar{\mathbf{\Lambda}}_j^2$ denote the mean of the j^{th} column of $\mathbf{\Lambda}$ having its elements squared. From Equation 5.1 we see that choosing $\gamma = 1$ will yield the maximal variance of the squared rotated variable loadings summed over all modes;

$$p \sum_{j=1}^{k} \left(\frac{1}{p} \sum_{i=1}^{p} (\mathbf{\Lambda}_{ij}^2)^2 - \frac{1}{p^2} \left(\sum_{i=1}^{p} \mathbf{\Lambda}_{ij}^2 \right)^2 \right) = p \sum_{j=1}^{k} \left(\frac{1}{p} \sum_{i=1}^{p} \left(\mathbf{\Lambda}_{ij}^2 - \bar{\mathbf{\Lambda}}_j^2 \right)^2 \right). \quad (5.2)$$

Since $\mathbf{R}$ is an orthonormal matrix, and thus cannot change the squared sum of the new basis vectors in $\mathbf{\Lambda}$, the variance of each column in $\mathbf{\Lambda}$ can only be increased by bringing some variable loadings close to zero, and let others grow large. Hence, a more simple structure of $\mathbf{\Lambda}$ is obtained. This tends to make the components, or the basis vectors, easier to interpret. The cost is that component correlation will be introduced for any rotation of the PCA basis, except for

180 degrees, in which case the variance would remain unchanged. Relaxing $\mathbf{\Phi}$ to be orthogonal, rather than orthonormal, will lead to both non-orthogonal variable loadings (i.e. $\mathbf{\Lambda}^T\mathbf{\Lambda}$ not diagonal), as well as to correlated variables [72]. It should be added that subgroups of $\mathbf{\Phi}$ can be rotated, while other modes are left unchanged. Thus, dispersions with block diagonals will be obtained. Such subgroups could be determined by identifying clusters in the eigenvalue spectrum of an initial PCA transformation [71]. However, $\mathbf{\Phi}$ will remain orthonormal and all components will be rotated in this paper.

Setting $\gamma = 0$ yields the special case denoted *quartimax*; a method introduced almost simultaneously by several researchers [57], and which preceded the varimax approach by a few years. In the quartimax case, Equation 5.1 becomes:

$$\mathbf{R}_{orthomax} = \arg\max_{\mathbf{R}} \sum_{j=1}^{k}\sum_{i=1}^{p}(\mathbf{\Phi}\mathbf{R})_{ij}^4. \tag{5.3}$$

It turns out that this expression minimizes the parsimony criterion put forward by Ferguson (see Harman [57]),

$$\sum_{i=1}^{p}\sum_{j=1}^{k}\sum_{q=1}^{j-1}(\mathbf{\Lambda}_{ij}\mathbf{\Lambda}_{iq})^2, \tag{5.4}$$

since $\mathbf{R}$ remains orthonormal and therefore does not change the squared sum of loadings. If this sum is squared, then for a single variable, i, we have

$$\left(\sum_{j=1}^{k}\mathbf{\Lambda}_{ij}^2\right)^2 = \sum_{j=1}^{k}\mathbf{\Lambda}_{ij}^4 + 2\sum_{j=1}^{k}\sum_{q=1}^{j-1}\mathbf{\Lambda}_{ij}^2\mathbf{\Lambda}_{iq}^2. \tag{5.5}$$

Consequently, as Equation 5.5 remains constant when summed over all variables, Equation 5.4 is minimized when Equation 5.3 is maximized. In other words, by emphasizing simplicity within rows of $\mathbf{\Lambda}$, quartimax is contrasted to varimax that emphasizes simplicity within columns of $\mathbf{\Lambda}$. Refer to Harman [57] for further details on the various, but similar, quartimax formulations.

When focusing on shape variability, one important – albeit rare – situation deserves mentioning. Imagine that k is close to p. Then $\mathbf{\Lambda}$ will approach the identity matrix, $\mathbf{I}$. This will happen even when the starting point is a very uneven eigenvalue spectrum. Such behavior is of course entirely correct; we should obtain a maximally sparse solution for a set of eigenvectors that span $\mathbb{R}^p$. But the implication for a shape model based on shapes in $\mathbb{R}^d$ ($d = 2$ or $d = 3$ typically) is that the solution depends solely on the original orientation of the d-dimensional coordinate system. The solution is in other words not rotation invariant and this fact becomes very apparent when k approaches p. In summary, choice of k will greatly influence the level of obtained sparsity, when all modes are rotated. This issue was also commented by Suinesiaputra et al. [148, 149]

Obviously, texture models are not affected by the above issue, since $d = 1$. Although the computations in Algorithm 5.1 becomes substantial when p is very large (say $p = 30000$ for a texture model) the growth is fortunately linear in p. Notice that the costly singular value decomposition is carried out on a $k \times k$ matrix, which does not pose a problem, as $k \ll p$ for such models.

Another issue is the ordering of the new variables stemming from an orthomax rotation. To this end, we discuss a set of criteria below that all order components by decreasing value of the criterion.

Component variance. This is the normal ordering of the principal components. Using this criterion, very sparse modes will tend to reside among the last components due to the orthonormality of $\mathbf{\Lambda}$. That is, sparse modes will be scaled more than dense, and consequently lead to smaller component scores.

Variance of squared loadings. As this is the criterion being optimized by the varimax rotation, this ordering may be a natural choice for having sparsity concentrated among the first modes.

Locality. Favorable if prior knowledge is available regarding interesting subparts of the original p-dimensional space. Used by Üzümcü et al. [161], Üzümcü et al. [162] and Suinesiaputra et al. [148, 149] when ordering sparse, ICA-based shape modes according to their effects along near-circular endo- and epicardial borders in cardiac magnetic resonance images.

Correlation. This ordering is suitable if k has been chosen to produce an appropriate amount of sparsity in the resulting modes and the objective is to find sparse, yet weakly correlated, modes. Those will thus be present in the latter part of the ordered modes.

Autocorrelation. Although, sparsity typically is obtained by fairly well-defined coherent parts of the original p dimensional domain, this behavior is not required by design. Ordering by autocorrelation will discriminate between abrupt changes and more smooth coherent modes. However, care has to be taken when estimating the autocorrelation for multi-part or open-contour shapes and for textures.

Clustering. If numerically large variable loadings are localized in several clusters e.g. along a contour in a shape model, then ordering according to the numbers of clusters and cluster size may be interesting.

Section 5.5 will demonstrate the use of three of the above criteria.

5.4 Details on Algorithm 1

This section serves to demonstrate the validity of Algorithm 5.1 in relation to Equation 5.1. Let $\circ$ denote the Hadamard (element-wise) product, let $\mathbf{A}_j$ denote the j^{th} column of $\mathbf{A}$, and let $\mathbf{\Gamma} = \mathbf{\Lambda} \circ \mathbf{\Lambda}$ (remember $\mathbf{\Lambda} = \mathbf{\Phi}\mathbf{R}$). Further, the following two Hadamard relations [103] will be used:

Let $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}^T$ denote $m \times n$ matrices and let $\mathbf{1}_q$ be a column vector of q ones. Then

$$\text{trace}((\mathbf{A} \circ \mathbf{B})(\mathbf{C}^T \circ \mathbf{D})) = \text{trace}((\mathbf{A} \circ \mathbf{B} \circ \mathbf{C})\mathbf{D}) \tag{5.6}$$

and

$$\mathbf{1}_m^T(\mathbf{A} \circ \mathbf{B})(\mathbf{C}^T \circ \mathbf{D})\mathbf{1}_m = \text{trace}(\mathbf{C}\ \text{Diag}(\mathbf{A}^T\mathbf{B})\mathbf{D}). \tag{5.7}$$

Equation 5.1 can now be written in matrix form,

$$\begin{aligned}
\mathbf{R}_{orthomax} &= \arg\max_{\mathbf{R}} \left(\sum_{j=1}^{k}\sum_{i=1}^{p} \mathbf{\Gamma}_{ij}^2 - \frac{\gamma}{p}\sum_{j=1}^{k}\left(\sum_{i=1}^{p}\mathbf{\Gamma}_{ij}\right)^2 \right) \\
&= \arg\max_{\mathbf{R}} \left(\text{trace}(\mathbf{\Gamma}^T\mathbf{\Gamma}) - \frac{\gamma}{p}\sum_{j=1}^{k}\left(\mathbf{1}_p^T\mathbf{\Gamma}_j\right)^2 \right) \\
&= \arg\max_{\mathbf{R}} \left(\text{trace}(\mathbf{\Gamma}^T\mathbf{\Gamma}) - \frac{\gamma}{p}\sum_{j=1}^{k}\mathbf{1}_p^T\mathbf{\Gamma}_j\mathbf{\Gamma}_j^T\mathbf{1}_p \right) \\
&= \arg\max_{\mathbf{R}} \left(\text{trace}(\mathbf{\Gamma}^T\mathbf{\Gamma}) - \frac{\gamma}{p}\mathbf{1}_p^T\mathbf{\Gamma}\mathbf{\Gamma}^T\mathbf{1}_p \right)
\end{aligned}$$

$$
\begin{aligned}
&= \arg\max_{\mathbf{R}} \left(\operatorname{trace}((\boldsymbol{\Lambda} \circ \boldsymbol{\Lambda})^T (\boldsymbol{\Lambda} \circ \boldsymbol{\Lambda})) - \frac{\gamma}{p} \mathbf{1}_p^T (\boldsymbol{\Lambda} \circ \boldsymbol{\Lambda})(\boldsymbol{\Lambda} \circ \boldsymbol{\Lambda})^T \mathbf{1}_p \right) \\
&= \arg\max_{\mathbf{R}} \Big(\operatorname{trace}(\left(\boldsymbol{\Lambda}^T \circ \boldsymbol{\Lambda}^T \circ \boldsymbol{\Lambda}^T \right) \boldsymbol{\Lambda}) - \\
&\qquad\qquad \frac{\gamma}{p} \operatorname{trace}(\boldsymbol{\Lambda} \cdot \operatorname{Diag}(\boldsymbol{\Lambda}^T \boldsymbol{\Lambda}) \boldsymbol{\Lambda}^T) \Big) \\
&= \arg\max_{\mathbf{R}} \Big(\operatorname{trace}(\mathbf{R}^T \boldsymbol{\Phi}^T (\boldsymbol{\Lambda} \circ \boldsymbol{\Lambda} \circ \boldsymbol{\Lambda})) - \\
&\qquad\qquad \frac{\gamma}{p} \operatorname{trace}(\boldsymbol{\Lambda}^T \boldsymbol{\Lambda} \cdot \operatorname{Diag}(\boldsymbol{\Lambda}^T \boldsymbol{\Lambda})) \Big) \\
&= \arg\max_{\mathbf{R}} \left(\operatorname{trace}(\mathbf{R}^T \boldsymbol{\Phi}^T (\boldsymbol{\Lambda} \circ \boldsymbol{\Lambda} \circ \boldsymbol{\Lambda}) - \frac{\gamma}{p} \mathbf{R}^T \boldsymbol{\Phi}^T \boldsymbol{\Lambda} \cdot \operatorname{Diag}(\boldsymbol{\Lambda}^T \boldsymbol{\Lambda})) \right) \\
&= \arg\max_{\mathbf{R}} \left(\operatorname{trace}(\mathbf{R}^T \mathbf{Q}) \right) \\
&\quad \text{where} \qquad \mathbf{Q} = \boldsymbol{\Phi}^T (\boldsymbol{\Lambda} \circ \boldsymbol{\Lambda} \circ \boldsymbol{\Lambda} - \frac{\gamma}{p} \boldsymbol{\Lambda} \cdot \operatorname{Diag}(\boldsymbol{\Lambda}^T \boldsymbol{\Lambda})). \qquad (5.8)
\end{aligned}
$$

In Algorithm 5.1, an iterative approach is taken to solving Equation 5.1. Here, the part where **R** enters non-linearly, i.e. **Q**, is kept fixed using the current estimate of **R**. Then, the singular value decomposition, in line 6 of Algorithm 5.1, produces the optimal **R** for the linear part as shown in Equation 5.8 which subsequently replaces the current estimate. The initial estimate of **R** is the identity matrix.

By assuming that **Q** does not depend on **R**, then Equation 5.8 would be maximized if and only if $\mathbf{R}^T\mathbf{Q}$ is symmetric and positive semi-definite[2]. This can be accomplished by choosing $\mathbf{R} = \mathbf{U}\mathbf{V}^T$, where **U** and **V** are taken from the singular value decomposition $\mathbf{Q} = \mathbf{U}\mathbf{S}\mathbf{V}^T$. That $\mathbf{R}^T\mathbf{Q}$ is symmetric and positive semi-definite can been seen by the following substitution[3]:

$$
\mathbf{R}^T\mathbf{Q} = \mathbf{R}^T\mathbf{U}\mathbf{S}\mathbf{V}^T = (\mathbf{U}\mathbf{V}^T)^T\mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{V}\mathbf{U}^T\mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{V}\mathbf{S}\mathbf{V}^T. \qquad (5.9)
$$

This concludes our presentation of the background of Algorithm 5.1 based on Neudecker [103] and ten Berge [154]. Further details can be found in [66, 103, 132, 155, 156].

[2] See the compact proof of Theorem 2 in [154] on Procrustes analysis.

[3] Remember that **S** is a diagonal matrix of singular values, and **U** and **V** hold orthogonal singular vectors.

5.5 Experimental Results

To illustrate the effects of orthomax rotation, three different cases have been selected and decomposed by principal component analysis and subsequently rotated using the varimax criterion. Two shape studies were carried out on contours stemming from chest radiographs and magnetic resonance images (MRI) of the human brain. Normal perspective images of frontal faces formed basis for a case study of sparse texture variability. Varimax-rotated components are compared to principal components in all three cases.

Figure 5.1 shows a decomposition of 247 chest radiograph annotations of the lungs, heart and clavicles based on 166 landmarks. This data is described in detail in [138] and [163]. In our case, the 16 largest principal components were retained. All varimax modes in Figure 5.1(b) were sorted by the absolute sum of the correlation coefficients in order to probe for localized yet weakly correlated modes, see Figure 5.2(c). We see that varimax mode 4 is related to the position of the aortic arch. Modes 3 and 5 relate to heart size, while modes 8 and 10 relate to clavicle orientation. These localized modes are contrasted by the conventional PCA modes shown in Figure 5.1(a). Another way of visualizing the variable loadings in each case is to relate gray-scales to the magnitudes of the elements in $\mathbf{\Phi}$ and $\mathbf{\Lambda}$. This is carried out in Figure 5.2(a), which clearly demonstrates the sparsity of the varimax solution. The 'flattening' of the eigenvalue spectrum carried out by the varimax rotation is illustrated in Figure 5.2(b) where the respective variances are plotted.

Figure 5.3 shows a decomposition of 62 annotations of the corpus callosum in mid-sagittal brain MRI using 78 landmarks. This data is described in more detail in [138, 142, 145]. Varimax ordering is similar to the previous case study. In Figure 5.3(b) we observe that varimax mode 1 relates to the isthmus area, mode 2 to bending of the splenium, mode 3 to the truncus area, while mode 4 is clearly related to the area of the rostrum and genu. In contrast, PCA mode 1 describes a simultaneous bending of the entire corpus callosum with an area change of the rostrum, genu and splenium. Again, the sparsity of the varimax-rotated components can also be appreciated in Figure 5.4(a).

Figures 5.5 and 5.6 show the results of a decomposition of 37 gray-scale face images. Further analyses of this data set can be found in [141] and [137]. Prior to our analyses, all images were compensated for any variation in shape by a piece-wise affine image warp similar to the one usually carried out in Active Appearance Models [24, 28]. While the PCA modes in Figures 5.5(a–j) demonstrate several effects within each mode, the varimax modes in Figures 5.5(k–t) show nicely isolated effects. The first principal component for example, shows absence/presence of beard as well as nostrils. Notice that Figures 5.5(a–t) show the magnitude of the variable loadings of each pixel position of the model, rather than the actual values of the basis vectors. Black and white represents high and low magnitude, respectively. Varimax modes are in this case ordered according to the sparsity criterion, namely the variance of the squared loadings of a mode. Interpretations of the first five varimax modes are as follows, absence/presence of i) nostrils, ii) lip spacing, iii) eyebrow thickness/shadow, iv) shadow below lower lip, and v) mustache. Each of these modes are shown in Figure 5.6 as modifications of the mean texture. Here it becomes more apparent that the modes, albeit being sparse, also carries additional information outside the areas mentioned above. This further indicates that even subtle changes to the texture can carry substantial changes to the perceived identity.

Orthomax rotations have also been implemented in a complete Active Appearance Models framework [141] with the aim of assessing their potential in future registration studies. A preliminary cross-validation study in the face data set showed a slight, though presumably insignificant, increase in accuracy[4] when employing varimax rotation to the texture model, compared to standard PCA-based texture model. To this end it is important to stress that uncoupled shape and texture models must be employed. If not, the third PCA traditionally used in AAMs will diagonalize the covariance matrices and yield a combined basis identical to that of the standard AAM.

Quartimax rotations were also carried out in the two former case studies. Although the deformation modes by design should show more exclusive changes[5] this behavior was not very clear. Due to the lack of differentiation from the varimax case, we have chosen not to show the quartimax modes.

5.6 Discussion

The medical image analysis literature is surprisingly devoid of references to sparse modeling using the orthomax criterion. The main contribution of this paper is therefore three-fold, i) broadening the knowledge of this simple, yet powerful, modification of principal components, ii) discussing its merits, and iii) providing a diverse range of examples on its use in medical applications.

We have found the method to be conceptually simple to understand as well as to implement. This is partly due to being a well-understood and well-described method within factor analysis. We further note that orthomax transformations are independent of domain dimensionality (2D/3D etc.) and spatial structure.

[4] Measured using the point to point distance between the ground truth shape and the converged model shape.

[5] In the sense that if one subpart of the shape is affected in one mode, it should not be much affected in the remaining modes.

An additional benefit is that many common computational frameworks already provide an implementation, e.g. R, S-plus, Matlab, et cetera. Considering the selection of k (the number of retained components) to lie with PCA, orthomax rotations are parameter-free. This is obviously a two-edged sword; while it leaves no frustrating choices up to the operator, it lacks the fine-grained flexibility, found in e.g. the sparse PCA method by Zou et al. [177]. Compared to sparse PCA, orthomax rotations have the benefit of being computationally feasible even for very high-dimensional spaces, found in e.g. texture modeling. Unfortunately, and unlike sparse PCA, orthomax rotations will rarely provide entirely sparse components. This is also illustrated by the examples in this paper. However, the relative differences in magnitude within orthomax modes may in practice be considered sufficiently sparse in many cases. As hinted earlier, the resulting amount of sparsity is directly related to the rank of the variation and the number of principal components subjected to orthomax rotation.

A long term goal for sparse modeling in relation to image interpretation and registration is to be able to separate inherent variation sources from chance correlation, thus providing greater – and justifiable – model flexibility, and in addition provide parameterizations that capture latent structures more accurately. The latter aspect could be of crucial importance in highly flexible, non-linear regression methods sensitive to initialization.

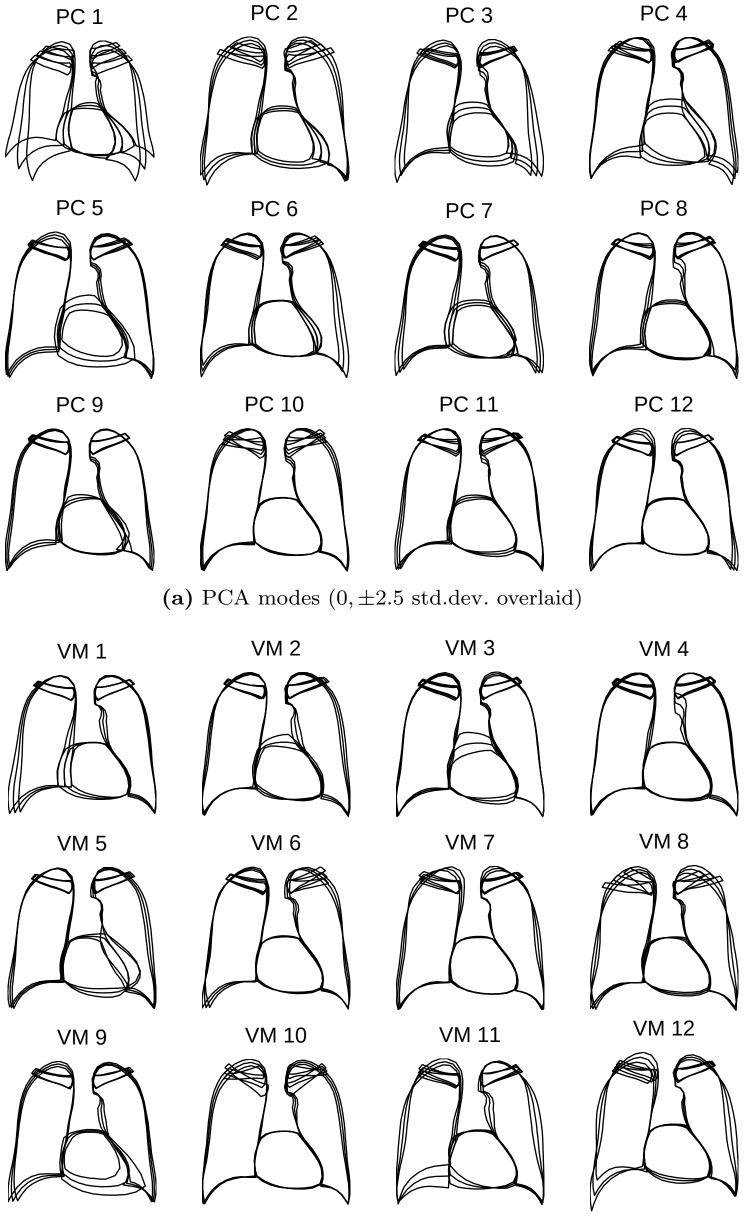
Application-wise, we note that pathologies are typically spatially localized, either with respect to shape or texture. Thus, we anticipate many medical application areas where sparse parameterizations, similar to the presented approach, are preferable to the conventional global PCA approach.

5.7 Conclusion

We have explored a computationally simple approach for rotation of principal components using the orthomax criterion, which directly optimizes sparsity leading to localized modes of variation suitable for medical image interpretation and exploratory analyses. We have found that both high-dimensional sparse modeling of shape variability ($p \approx 300$), as well as extremely high-dimensional sparse modeling of texture variability ($p \approx 30000$) are feasible. Case studies on radiographs, brain MRI, and face images showed local modes of natural variation contrary to global PCA modes. Applications include computer-aided diagnosis in terms of exploratory analyses, disease characterization, et cetera.

Acknowledgements

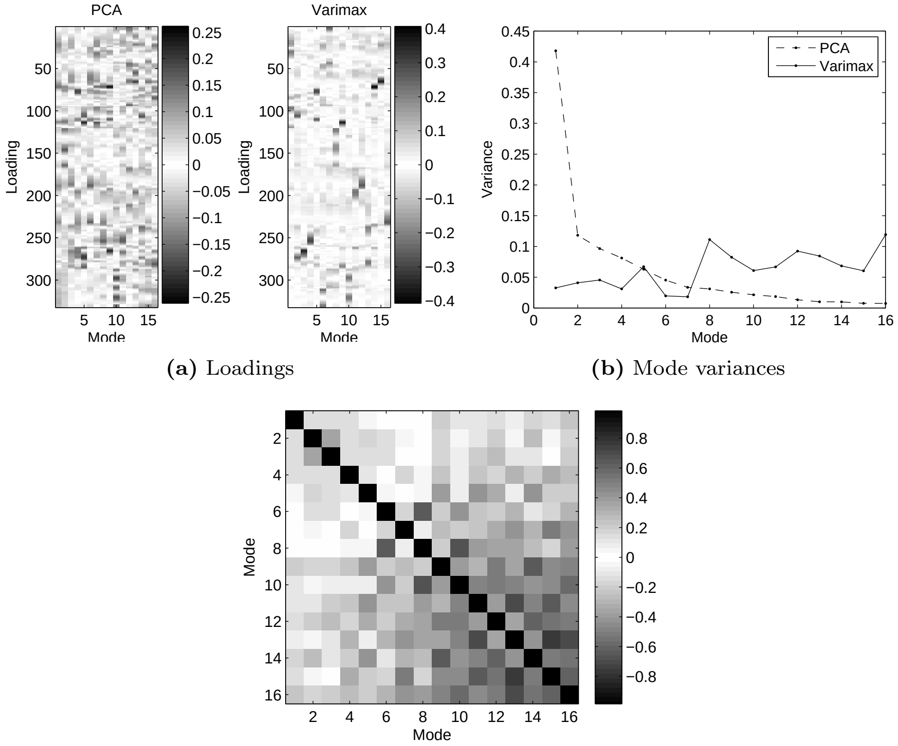
Dr. Bram van Ginneken, Image Sciences Institute, University Medical Center Utrecht, kindly provided the lung annotations used in this study. Dr. Ginneken also assisted with the anatomical interpretation. Charlotte Ryberg and Egill Rostrup, The Danish Research Centre for Magnetic Resonance, Copenhagen University Hospital Hvidovre, kindly provided the MRIs used to produce the corpus callosum annotations. M. B. Stegmann was supported by The Danish Research Agency, grant no. 2059-03-0032. K. Sjöstrand was supported by The Technical University of Denmark, DTU.



(a) PCA modes $(0, \pm 2.5$ std.dev. overlaid)

(b) Varimax modes $(0, \pm 2.5$ std.dev. overlaid)

Figure 5.1: Shape modes calculated from 247 chest radiograph annotations of the lungs, heart and clavicles.

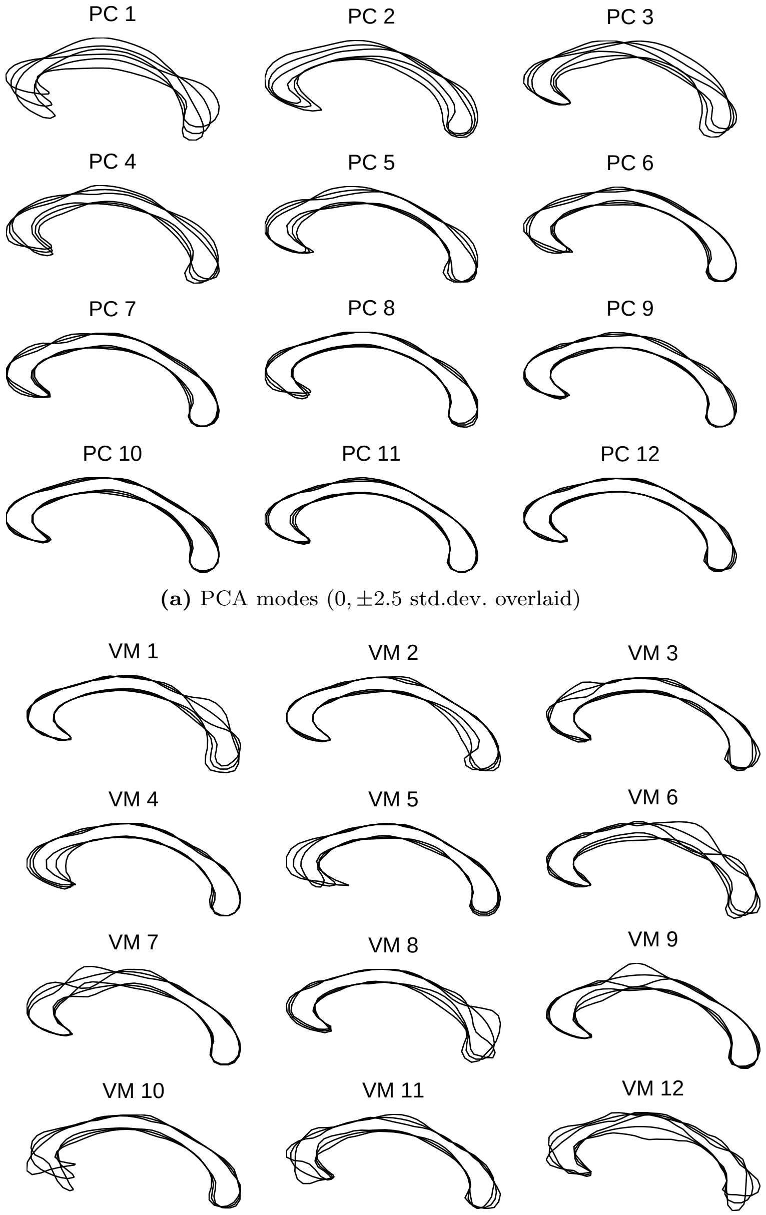


(a) Loadings

(b) Mode variances

(c) Correlation coefficients

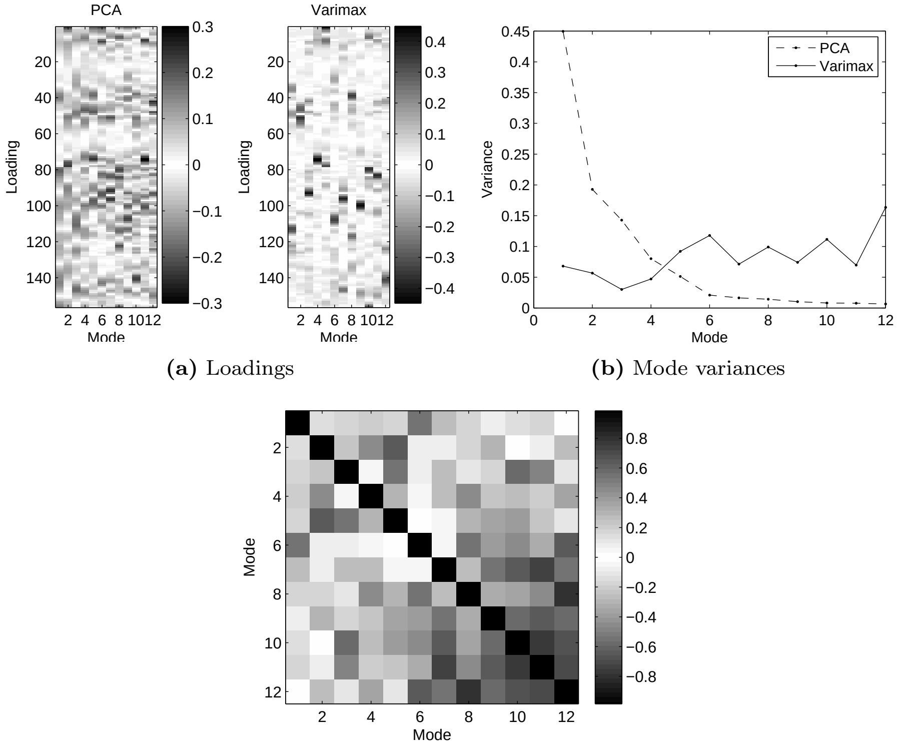
Figure 5.2: Loadings, variances and correlation coefficients for PCA and varimax, calculated on the lung data set.



(a) PCA modes $(0, \pm 2.5$ std.dev. overlaid)

(b) Varimax modes $(0, \pm 2.5$ std.dev. overlaid)

Figure 5.3: Shape modes calculated from 62 corpus callosum annotations in mid-sagittal brain magnetic resonance images.

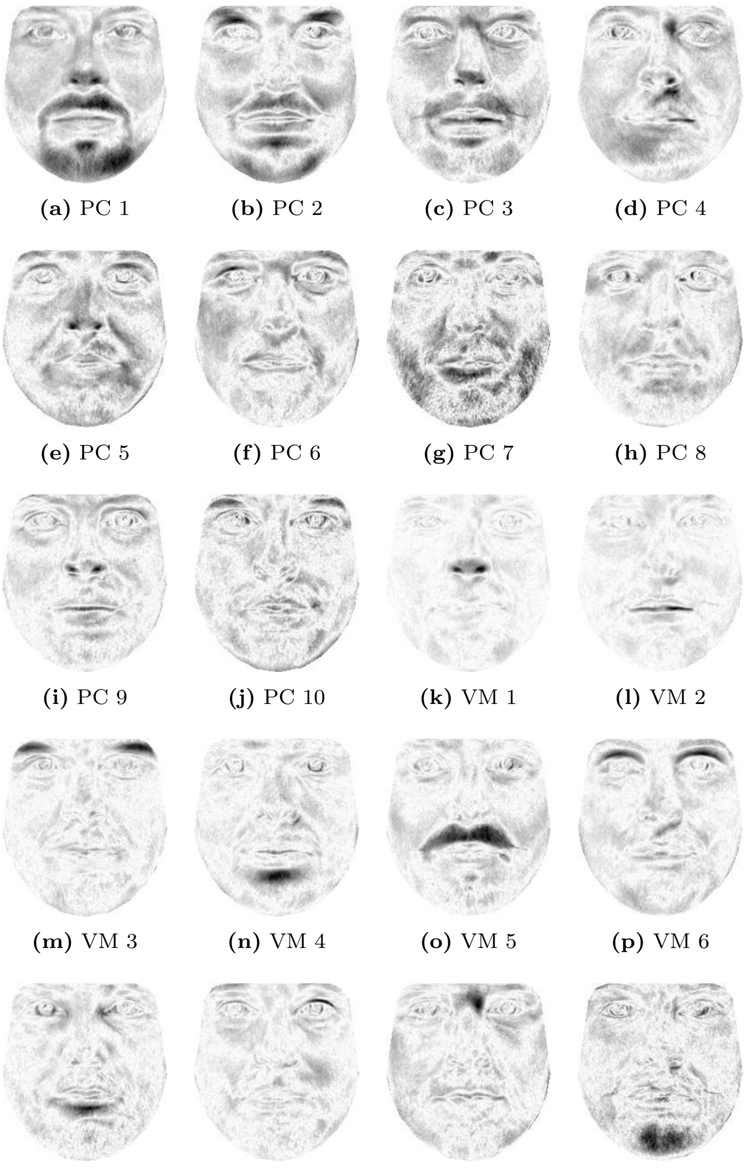


(a) Loadings

(b) Mode variances

(c) Correlation coefficients

Figure 5.4: Loadings, variances and correlation coefficients for PCA and varimax, calculated on the corpus callosum data set.



(a) PC 1 **(b)** PC 2 **(c)** PC 3 **(d)** PC 4

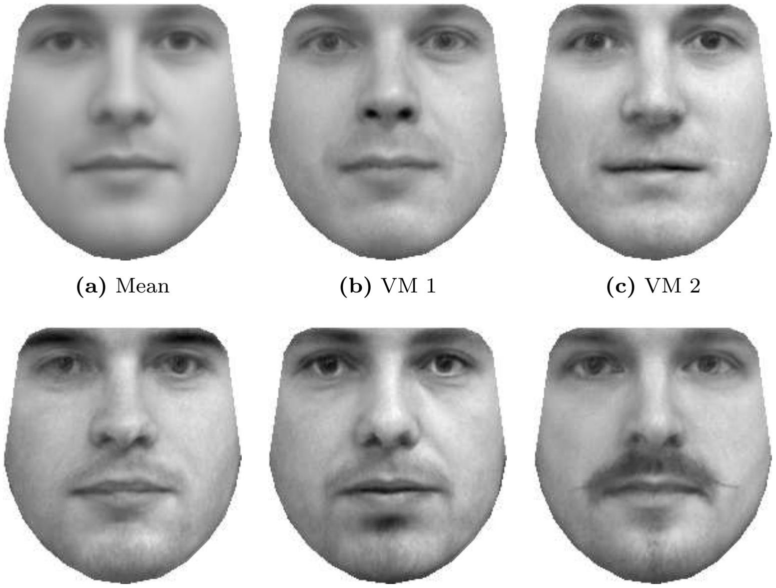
(e) PC 5 **(f)** PC 6 **(g)** PC 7 **(h)** PC 8

(i) PC 9 **(j)** PC 10 **(k)** VM 1 **(l)** VM 2

(m) VM 3 **(n)** VM 4 **(o)** VM 5 **(p)** VM 6

(q) VM 7 **(r)** VM 8 **(s)** VM 9 **(t)** VM 10

Figure 5.5: The magnitude of eigenvectors calculated from 37 face images arranged as eigenimages. PCA modes are ordered according to the variance of corresponding principal score. Varimax modes are ordered according to sparsity given by the variance of the squared loadings.



(a) Mean **(b)** VM 1 **(c)** VM 2

(d) VM 3 **(e)** VM 4 **(f)** VM 5

Figure 5.6: Varimax texture modes calculated from 37 face images. (a) mean texture. (b–f) mean texture modified by 2.5 standard deviations of the corresponding mode scores.

Chapter 6

Sparse Principal Component Analysis in Medical Shape Modeling

Karl Sjöstrand, Mikkel B. Stegmann, and Rasmus Larsen

Abstract

Principal component analysis (PCA) is a widely used tool in medical image analysis for data reduction, model building, and data understanding and exploration. While PCA is a holistic approach where each new variable is a linear combination of all original variables, *sparse PCA* (SPCA) aims at producing easily interpreted models through sparse loadings, i.e. each new variable is a linear combination of a subset of the original variables. One of the aims of using SPCA is the possible separation of the results into isolated and easily identifiable effects. This article introduces SPCA for shape analysis in medicine. Results for three different data sets are given in relation to standard PCA and sparse PCA by simple thresholding of small loadings. Focus is on a recent algorithm for computing sparse principal components, but a review of other approaches is supplied as well. The SPCA algorithm has been implemented using Matlab and is available for download. The general behavior of the algorithm is investigated, and strengths and weaknesses are discussed. The original report on the SPCA algorithm argues that the ordering of modes is not an issue. We disagree on this point and propose several approaches to establish sensible orderings. A method that orders modes by decreasing variance and maximizes the sum of variances for all modes is presented and investigated in detail.

6.1 Introduction

Few computational methods for data understanding, exploration and reduction has found more use than principal component analysis (PCA). PCA takes an $(n \times p)$ data matrix $\mathbf{X}$, n being the number of observations and p being the number of variables, and transforms it by $\mathbf{Z} = \mathbf{XB}$ such that the derived variables (the columns of $\mathbf{Z}$) are uncorrelated and correspond to directions of maximal variance in the data. The derived coordinate axes are the columns of $\mathbf{B}$, called *loading vectors* with individual elements known as *loadings.* These are at right angles with each other; PCA is simply a rotation of the original coordinate system, and the $(p \times p)$ loading matrix $\mathbf{B}$ is the rotation matrix. The new variables (the columns of $\mathbf{Z}$) are known as *principal components* (PCs). Usually only the first k components, $k < p$, are retained since these explain the majority of the sample set variance. This makes $\mathbf{Z}$ $(n \times k)$ and $\mathbf{B}$ $(p \times k)$. The loading matrix can be calculated using a singular value decomposition of the data matrix $\mathbf{X}$ or through an eigenanalysis of the corresponding covariance or correlation matrix.

Another way of viewing PCA is by treating each new variable as a linear combination of the original variables. The loadings then translate to coefficients and may be investigated in detail to determine the important factors behind each PC. The problem is that each new variable is a linear combination of *all* variables, and the loadings are typically non-zero. This makes interpretation difficult. *Sparse principal component analysis* aims at approximating the properties of regular PCA while keeping the number of non-zero loadings small.

The most straight-forward way of obtaining sparse loadings is by simple thresholding, where sufficiently small loadings are truncated to zero. The threshold can be chosen using e.g. Jeffers' criterion [69] of excluding, disregarding signs, loadings below 70% of the largest loading for each PC. Thresholding can be misleading in several respects, as discussed by Jolliffe [17]. The influence of a variable on a specific PC is not dependent on the magnitude of the corresponding loading only, but is governed by a series of relationships, such as variable size, or analogously, variance.

Among the earliest methods for obtaining a *simple structure* of the loadings of the original variables is the class of orthomax rotations [57], where an initial basis is rotated due to some objective criterion. The basis can for example be provided by a PCA. Let $\mathbf{B}$ be a $p \times k$ orthonormal matrix (of column eigenvectors) and $\mathbf{\Omega}$ be an orthonormal rotation matrix in $\mathbb{R}^k$, i.e. $\text{trace}(\mathbf{\Omega})\mathbf{\Omega} = \mathbf{I}$. Then, the class of orthomax rotations can be defined as

$$\mathbf{\Omega}_o = \arg\max_{\mathbf{\Omega}} \left(\sum_{j=1}^{k} \sum_{i=1}^{p} (\mathbf{B\Omega})_{ij}^4 - \frac{\gamma}{p} \sum_{j=1}^{k} \left(\sum_{i=1}^{p} (\mathbf{B\Omega})_{ij}^2 \right)^2 \right), \qquad (6.1)$$

where $\mathbf{\Omega}_o$ denotes the resulting rotation and γ denotes the type. In the orthomax class we find the Varimax [76] case where $\gamma = 1$. Here, Equation 6.1 simplifies to a sum of variances. The variances are calculated for each loading vector where the individual loadings are squared. This emphasizes sparsity within each loading vector by clustering loadings into an approximate bimodal distribution of large and very small loadings. Although the resulting components may not be strictly sparse, one benefit of the Varimax method is that it is computationally feasible in high-dimensional cases, see e.g. [139]

Chennubhotla and Jepson [18] present another criterion for finding a suitable rotation matrix based on the entropy of the loading matrix. A cost function,

$$C = C_1 + \lambda C_2,$$

is minimized where $C_1 = \sum_{j=1}^{k} -d_j \log d_j$ and d_j is the relative variance of the jth principal component. Next, $C_2 = \sum_{i=1}^{p} \sum_{j=1}^{k} -b_{i,j}^2 \log b_{i,j}^2$, were $b_{i,j}$ denotes the elements of the $(p \times k)$ loading matrix. Optimizing C_1 alone gives the standard PCA solution, while C_2 is minimal for the identity matrix, thus promoting sparsity. Similarly to the Varimax criterion, suppressed loadings will be small but non-zero. To achieve strict sparsity, thresholding of small loadings is performed as discussed above. The resulting loading vectors will, contrary to those constructed using the Varimax criterion, explain a decreasing amount of variance of the original data set; a feature it has in common with regular PCA. Additionally, the number of non-zero loadings also decrease, making a multi-scale interpretation possible.

Simple principal components [166] is a technique for producing particularly simple, and possibly sparse, loading vectors. It uses a series of in-plane rotations affecting two loading vectors at a time such that the resulting directions explain maximal variance subject to being represented by integers. The end result is a set of orthogonal loading vectors represented by (primarily small) integers. Empirical evidence shows that the correlations between the resulting PCs are low. Small loadings will typically be translated to zeros, resulting in a sparse loading matrix structure. Similar ideas have been put forth by Hausman [61] and Rousson and Gasser [123].

d'Aspremont et al. [31] take a variational approach to sparse PCA. The PCs are estimated separately by approximating a positive semidefinite symmetric matrix (the covariance or correlation matrix) by a rank-one matrix, $\mathbf{bb}^T$. To impose sparsity, a constraint is added on the maximum number of non-zero elements of $\mathbf{b}$, known as the *cardinality* of $\mathbf{b}$. This direct formulation results in a non-convex optimization problem that is difficult to solve. The problem is therefore relaxed by replacing the cardinality constraint with a convex one, making the computation feasible. The resulting PCs are reported to explain a larger proportion of variance than competing algorithms, but the complexity of the formulation grows quickly with the number of variables.

This article focuses on a method for computing sparse loading vectors using concepts from variable selection in regression. A method coined SCoTLASS [74] (Simplified Component Technique-LASSO) predates this method and is based on similar ideas. Maximizing the expression

$$\mathbf{b}_i^T \mathbf{R} \mathbf{b}_i,$$

where $\mathbf{R}$ denotes the covariance matrix of $\mathbf{X}$, subject to

$$\mathbf{b}_i^T \mathbf{b}_i = 1 \quad \text{and} \quad \mathbf{b}_j^T \mathbf{b}_i = 0, \;\; j \neq i,$$

renders the solution of a regular PCA. The authors propose to add the constraint

$$\|\mathbf{b}_i\|_1 = \sum_{j=1}^{p} |b_{ij}| \leq t, \quad t \in \mathbb{R}^+, \quad \forall i.$$

The parameter t controls the sparsity of the loading vectors $\mathbf{b}_k$. The addition of this constraint was inspired by the LASSO [157] regression method described below. However, this necessitates the use of a numerical optimization method. The problem formulation contains p parameters which is a potentially large number, and the cost function contains several local minima. The authors use a simulated annealing approach for optimization, which adds a number of tuning parameters in itself.

The following section presents the theory of the present method of sparse PCA, hereafter simply denoted SPCA. Section 6.3 shows results on shape data from three different data sets along with results on the general properties of SPCA. Section 6.4 discusses the obtained results, debates the advantages and drawbacks of SPCA and proposes a range of different possibilities for ordering of modes. Section 6.5 concludes the paper.

6.2 Methods

This section gives a brief description of the SPCA algorithm and discusses its relation to variable selection methods in regression. For a complete treatment, consult [177] and the preliminary papers [40, 157] and [175].

6.2.1 Regression Techniques

The regression methods presented here all originate from ordinary least squares (OLS) approximations. The *response* variable $\mathbf{y}$ is approximated by the *predictors* in $\mathbf{X}$. The coefficients for each variable (column) of $\mathbf{X}$ are contained in $\mathbf{b}$,

$$\mathbf{b}_{\mathrm{OLS}} = \arg\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2, \tag{6.2}$$

where $\|\cdot\|$ represents the L2-norm. This is the best linear unbiased estimator given a number of assumptions, such as independent and identically distributed (i.i.d.) residuals. However, if some bias is allowed, estimators can be found with lower mean square error than OLS when tested on an unseen set of observations. A common way of implementing this is by introducing some constraint on the coefficients in $\mathbf{b}$. The methods described here use constraints on either the L1-norm or the L2-norm of $\mathbf{b}$, or both. Adding the L2 constraint gives

$$\mathbf{b}_{\mathrm{ridge}} = \arg\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \lambda\|\mathbf{b}\|^2. \tag{6.3}$$

This is known as ridge regression. Any positive λ will shrink the coefficients of $\mathbf{b}$; if λ is chosen carefully, this may lead to improved prediction accuracy and better numerical properties. Replacing the L2-norm in the constraint with the L1-norm gives

$$\mathbf{b}_{\mathrm{LASSO}} = \arg\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \delta\|\mathbf{b}\|_1, \tag{6.4}$$

where $\|\mathbf{b}\|_1 = \sum_{i=1}^{p} |b_i|$. This method is coined LASSO [157], the least absolute shrinkage and selection operator. As the name implies, using the L1-norm not only shrinks the coefficients, but drives them one by one to exactly zero as δ grows. This implements a form of variable selection, as minor coefficients will be set to zero in a controllable fashion, while the remaining coefficients will be altered to mimic the response in the best possible way. The relation to the problem of setting small PCA loadings to zero is already evident, but some more theory is needed before this can be properly handled.

LASSO has proven to be a very powerful regression and variable selection technique, but it has a few limitations. If $p > n$, i.e. there are more variables than observations, LASSO chooses a maximum of n variables. If there is a group of strongly correlated predictors, LASSO tends to choose a single predictor from that group only. The elastic net regression method [175] was developed to address these shortcomings. It uses a combination of the constraints from ridge regression and LASSO,

$$\mathbf{b}_{\mathrm{nEN}} = \arg\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \lambda\|\mathbf{b}\|^2 + \delta\|\mathbf{b}\|_1, \tag{6.5}$$

where nEN is short for naive elastic net for reasons described below. The elastic net can be formulated as a LASSO problem on augmented variables,

$$\mathbf{b}^*_{\mathrm{nEN}} = \arg\min_{\mathbf{b}^*} \|\mathbf{y}^* - \mathbf{X}^*\mathbf{b}^*\|^2 + \frac{\delta}{\sqrt{1+\lambda}}\|\mathbf{b}^*\|_1, \tag{6.6}$$

where

$$\mathbf{X}^*_{(n+p)\times p} = \frac{1}{\sqrt{1+\lambda}}\begin{bmatrix} \mathbf{X} \\ \sqrt{\lambda}\mathbf{I}_p \end{bmatrix}, \quad \mathbf{y}^*_{n+p} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_p \end{bmatrix}, \quad \mathbf{b}^* = \sqrt{1+\lambda}\mathbf{b}.$$

The authors argue that the formulation in Equation 6.6 incurs a double amount of coefficient shrinkage, which is why the solution of Equation 6.5 is referred to as naive. The excessive shrinkage is compensated for in the final solution for $\mathbf{b}_{\mathrm{EN}}$ which is

$$\mathbf{b}_{\mathrm{EN}} = \sqrt{1+\lambda}\mathbf{b}^*_{\mathrm{nEN}} = (1+\lambda)\mathbf{b}_{\mathrm{nEN}}. \tag{6.7}$$

The resulting LASSO problem has more observations $(p+n)$ than variables (p), which is why cases where $p > n$ are handled gracefully. If $\lambda > 0$, the elastic net constraint function $\lambda\|\mathbf{b}\|^2 + \delta\|\mathbf{b}\|_1$ is strictly convex. It can be shown [175] that the difference between coefficients of highly correlated variables in such a system is very small. The elastic net therefore has a tendency of grouping variables, contrary to the LASSO. These are two properties that are desirable in a PCA framework. Problems where there are more variables than observations are common, and principal components built from highly correlated and significant variables are easier to interpret.

Ordinary least squares and ridge regression have closed-form solutions, that is, $\mathbf{b}_{\mathrm{OLS}}$ and $\mathbf{b}_{\mathrm{ridge}}$ can be expressed as simple functions of $\mathbf{X}$, $\mathbf{y}$ and λ. This is not true for the LASSO and elastic net methods. For many years, LASSO solutions were found using standard optimization techniques, which made for long computation times. In 2002, Efron et al. [40] published a report on a new regression method called least angle regression (LARS). The terminal S in LARS refers to its close relation to stagewise regression and LASSO. Although conceptually different, the method is shown to be very similar to LASSO, and through a small modification, the exact LASSO solution can be computed. The method is built on a powerful geometric framework, through which a computationally thrifty algorithm is conceived. The algorithm starts with all coefficients at zero, and successively adds predictors until all variables are active and the ordinary least squares solution is reached. In other words, LARS returns the solutions for all possible values of δ. What remains is to pick a suitable solution, a proper value of δ. This can for instance be done using cross-validation or prior knowledge of the desired number of non-zero coefficients. In the elastic net setting, LARS returns the solutions corresponding to all possible values of δ given a value of λ. Zou and Hastie [175] describes a further development of the LARS algorithm tailor made to suit the elastic net framework. This extension is called LARS-EN.

In summary, a regression approach has been presented through which a relevant subset of variables can be selected, which handles the case of more variables than observations gracefully, and which can be computed efficiently. We now turn to the problem of calculating sparse PCs. Note that ”sparse PCs” refers to principal components formed by linear combinations of sparse sets of variables.

6.2.2 Sparse Principal Component Analysis (SPCA)

The simplest approach to SPCA using regression is by treating each principal component as a response vector and regressing this on the p variables. Denoting the ith PC and loading vector by $\mathbf{z}_i$ and $\mathbf{b}_i$ respectively, and inserting this into the elastic net framework gives

$$\hat{\mathbf{b}}_i = \arg\min_{\mathbf{b}_i} \|\mathbf{z}_i - \mathbf{X}\mathbf{b}_i\|^2 + \lambda\|\mathbf{b}_i\|^2 + \delta\|\mathbf{b}_i\|_1. \tag{6.8}$$

The principal component $\mathbf{z}_i$ is calculated using regular PCA. The regression procedure will calculate a loading vector $\mathbf{b}_i$ such that the resulting PC is close to $\mathbf{z}_i$ while being sparse. The weakness of this approach is that all solutions are constrained to the immediate vicinity of a regular PCA. A better approach would be to approximate the *properties* of PCA, rather than its exact results. Specifically, the loading matrix $\mathbf{B}$ should be near orthogonal, and the correlations between the PCs of the scores matrix $\mathbf{Z}$ should be kept low. Zou and Hastie propose a problem formulation called the *SPCA criterion* [177] to address this.

$$\begin{gathered}(\hat{\mathbf{A}}, \hat{\mathbf{B}}) = \arg\min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i\|^2 + \lambda\sum_{j=1}^{k} \|\mathbf{b}_j\|^2 + \sum_{j=1}^{k} \delta_j\|\mathbf{b}_j\|_1 \\ \text{subject to} \qquad \mathbf{A}^T\mathbf{A} = \mathbf{I}_k \end{gathered} \tag{6.9}$$

To clarify this expression, it will be broken down into components. First, $\mathbf{B}^T\mathbf{x}_i$ takes the variables of observation i and projects them onto the principal axes (loading vectors) of $\mathbf{B}$. Note that $\mathbf{x}_i$ denotes the ith column of $\mathbf{X}^T$. Only k PCs are retained, meaning that some information is lost in this transformation. Further, $\mathbf{A}\mathbf{B}^T\mathbf{x}_i$ takes the scores of $\mathbf{B}^T\mathbf{x}_i$ and transforms them back into the original space. The orthogonality constraint on $\mathbf{A}$ makes sure $\mathbf{B}$ is near orthogonal. The whole term $\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i\|^2$ measures the reconstruction error. The remaining constraints are the same as for elastic net regression, driving the columns of $\mathbf{B}$ towards sparsity. The constraint weight λ must be chosen beforehand, and has the same value for all PCs, while δ may be set to different values for each PC, offering good flexibility. It can be shown [177] that for $\delta_j = 0 \ \forall j$, the SPCA criterion is minimized by setting $\mathbf{A}$ and $\mathbf{B}$ equal to the loading matrix of ordinary PCA. Hence, the solutions of the present formulation of SPCA conveniently range from ordinary PCA on one end, to the (maximally sparse) zero matrix on the other.

Equation 6.2.2 resembles the elastic net formulation but there is a significant difference. Instead of estimating a single coefficient vector, this problem has two matrices of coefficients, $\mathbf{A}$ and $\mathbf{B}$. A reasonably efficient optimization method for minimizing the SPCA criterion is presented in [177]. First, assume that $\mathbf{A}$ is known. By expanding and rearranging Equation 6.2.2, it is shown that $\mathbf{B}$ can be estimated by solving k independent naive elastic net problems, one for each column of $\mathbf{B}$. Referring to Equation 6.5, the data matrix is $\mathbf{X}$ as usual while $\mathbf{y} = \mathbf{X}\mathbf{a}_i$ for the ith loading vector. On the other hand, if $\mathbf{B}$ is known, $\mathbf{A}$ can be calculated using a singular value decomposition; if $\mathbf{X}^T\mathbf{X}\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, then $\mathbf{A} = \mathbf{U}\mathbf{V}^T$. Since both matrices are unknown, an initial guess is made and $\mathbf{A}$ and $\mathbf{B}$ are estimated alternately until convergence. Zou et al. [177] suggests initializing $\mathbf{A}$ to the loadings of k first ordinary principal components.

6.2.3 Ordering of principal components

One goal of PCA is to recover latent variables that are as descriptive as possible. This is done by maximizing the variance of each PC subject to being orthogonal to higher order PCs. The performance of PCA methods is commonly measured by the amount of variance explained by each PC, and the total amount of variance for k modes. Regular PCA is the only linear transformation that produces both orthogonal loadings and uncorrelated scores [72]. For methods that produce correlated scores, variances cannot be calculated directly, as some of the variance explained by one PC will be present in others. This calls for a fair evaluation method. Several such methods are presented in [50], where it is concluded that the most powerful method is to measure *adjusted variance*, a term used by Zou and Hastie who suggest the same method in [177]. The idea is that the variance of each PC should be adjusted for the variance already explained by higher order components. For mean centered variables, such as those derived by PCA, correlation is equivalent to the cosine of the angle between vectors. Zero correlation corresponds to a 90° angle between vectors while fully correlated variables are parallel. Adjustment of a PC therefore amounts to a transformation such that the resulting vector is at right angles with all higher order PCs. This is also known as Gram-Schmidt orthogonalization.

The variance of the jth PC is proportional to its squared length, $\operatorname{var} \mathbf{z}_j = \frac{\mathbf{z}_j^T\mathbf{z}_j}{n} \propto \mathbf{z}_j^T\mathbf{z}_j$. Any ordering that maximizes the total variance therefore also maximizes the sum of squared lengths. For ease of notation, squared lengths are considered in the following equations.

A vector $\mathbf{z}_j$ may be orthogonalized, or *adjusted* with respect to another vector $\mathbf{z}$ using orthogonal projection by

$$\hat{\mathbf{z}}_j = \mathbf{z}_j - \mathbf{z}(\mathbf{z}^T\mathbf{z})^{-1}\mathbf{z}^T\mathbf{z}_j.$$

The jth PC should be adjusted for all higher order PCs. Assume that the variables, the columns of $\mathbf{Z}$, have been sorted according to decreasing order. The adjustment can then be carried out for all higher order PCs simultaneously by

$$\hat{\mathbf{z}}_j = \mathbf{z}_j - \mathbf{Z}_{(j-1)}(\mathbf{Z}_{(j-1)}^T\mathbf{Z}_{(j-1)})^{-1}\mathbf{Z}_{(j-1)}^T\mathbf{z}_j,$$

where $\mathbf{Z}_{(j)} = [\mathbf{z}_1 \ldots \mathbf{z}_j]$ [50].

The SPCA criterion (6.2.2) keeps the loading matrix near orthogonal by forcing $\mathbf{A}$ to be orthogonal, but does nothing to encourage uncorrelated scores. This makes an orthogonalization process central to SPCA. Zou and Hastie argue that the order of the components is not an issue; the order is left unaltered, making it possible for lower order modes to explain more variance than higher order modes. Furthermore, the amount of total adjusted variance is dependent on the ordering of the PCs, and may not be maximal in this case.

Formally, the variable ordering that maximizes the total variance can be established by maximizing $\sum_j \hat{\mathbf{z}}_j^T\hat{\mathbf{z}}_j$ and allowing for permutations,

$$\arg\max_{\mathbf{P}\in\mathcal{P}_k} \tilde{\mathbf{z}}_1^T\tilde{\mathbf{z}}_1 + \sum_{j=2}^{k} \tilde{\mathbf{z}}_j^T\tilde{\mathbf{z}}_j - \tilde{\mathbf{z}}_j^T\tilde{\mathbf{Z}}_{(j-1)}(\tilde{\mathbf{Z}}_{(j-1)}^T\tilde{\mathbf{Z}}_{(j-1)})^{-1}\tilde{\mathbf{Z}}_{(j-1)}^T\tilde{\mathbf{z}}_j, \qquad (6.10)$$

where $\tilde{\mathbf{Z}} = \mathbf{ZP}$ is the permuted scores matrix and $\mathcal{P}_k$ is the set of permutation matrices of size k. Note that the supremum of Equation 6.10 is the sum of unadjusted variances which is equal to the maximum iff $\mathbf{Z}$ is orthogonal. The simplest way of finding the optimal permutation is by trying all $k!$ possible permutations, which is feasible for a low number of PCs. This paper proposes a forward selection-type rule for picking an ordering with two properties; the variance of a PC is less than or equal to the variances of higher order PCs, and the expression in Equation 6.10 is maximized in most cases. The rule is simple. Treat one PC at a time. At each step, choose the PC with largest (adjusted) variance, and adjust the scores matrix for this PC. This means that in the first step, we calculate the variances of all (unadjusted) PCs and choose the one with greatest variance. All PCs ($\mathbf{Z}$) are then adjusted with respect to the chosen PC ($\mathbf{z}_j$) using

$$\hat{\mathbf{Z}} = \mathbf{Z} - \mathbf{z}_j(\mathbf{z}_j^T\mathbf{z}_j)^{-1}\mathbf{z}_j^T\mathbf{Z}. \qquad (6.11)$$

In the second step, the adjusted variances from the first step are considered. Again, the PC with greatest variance is chosen, and all PCs are updated using Equation 6.11. This process is repeated for all PCs. This results in a zero $\mathbf{Z}$ matrix, however, a sensible ordering has been established. This ordering is finally applied to the loading vectors and the original scores matrix.

The first property of this rule, which states that variances are decreasing, is easily realized since the longest vector is chosen in each step and since the squared length cannot grow as the vector is adjusted for some other vector. The second property of maximal total variance is empirically shown below to be fulfilled to a large extent, but as shall be seen, there are counter examples, e.g. $\mathbf{Z} = [[0\ 1.5]^T\ [1\ 1]^T\ [1\ \ {-1}]^T]$.

6.3 Results

The SPCA algorithm has been applied to medical shape analysis. The shape data is contained in a data matrix $\mathbf{X}$ $(n \times p)$ where each shape corresponds to one row (observation) and the variables consist of the different landmark positions. Landmarks are defined by two coordinates (2D data); these are treated separately such that one coordinate is one variable. This project is concerned with 2D data only, although the techniques described herein are directly applicable to data of any dimensionality.

Three data sets were used in this study. The first consists of 37 annotations of the human face. Each face is represented by 58 landmarks. The second data set is a shape model of the lungs, the heart and the clavicles. The set contains 247 observations, each with 166 landmarks. The final data set represents the *corpus callosum* brain structure. This is the bundle of nerve fibers connecting the two cerebral hemispheres of the brain. The structure is well defined in the *mid-sagittal plane*, the plane that separates the left hemisphere from the right [144]. Further away from this plane, the structure dissolves into separate fibers, which is why it is best analyzed in 2D. The set has 62 observations, each with 78 landmarks.

Figure 6.1 shows regular and sparse decompositions of the face data set. Each set of figures shows the first 12 *modes of variation*[1], ordered by the method described above. It is evident that regular PCA produces holistic modes of variation, each describing a series of effects at once, making interpretation difficult. SPCA, on the other hand, manages to display more or less separate effects for

[1] Modes of variation is a commonly used term where the jth mode denotes movements along the axis defined by the jth loading vector. The mean shape defines the origin and perturbations are measured offset to this.

each mode. SPCA modes 2, 8 and 12 correspond to mouth opening/closing, upper lip thickness and smile/frown respectively. SPCA modes 4, 6 and 7 show differing eyebrow configurations. Figure 6.2 shows corresponding images for the lungs, heart and clavicles data set. SPCA mode 2 depicts the length of the clavicles, while most other modes are concerned with either lung or heart geometry, or both (e.g. SPCA mode 5). Figure 6.3 presents results for the corpus callosum data set.

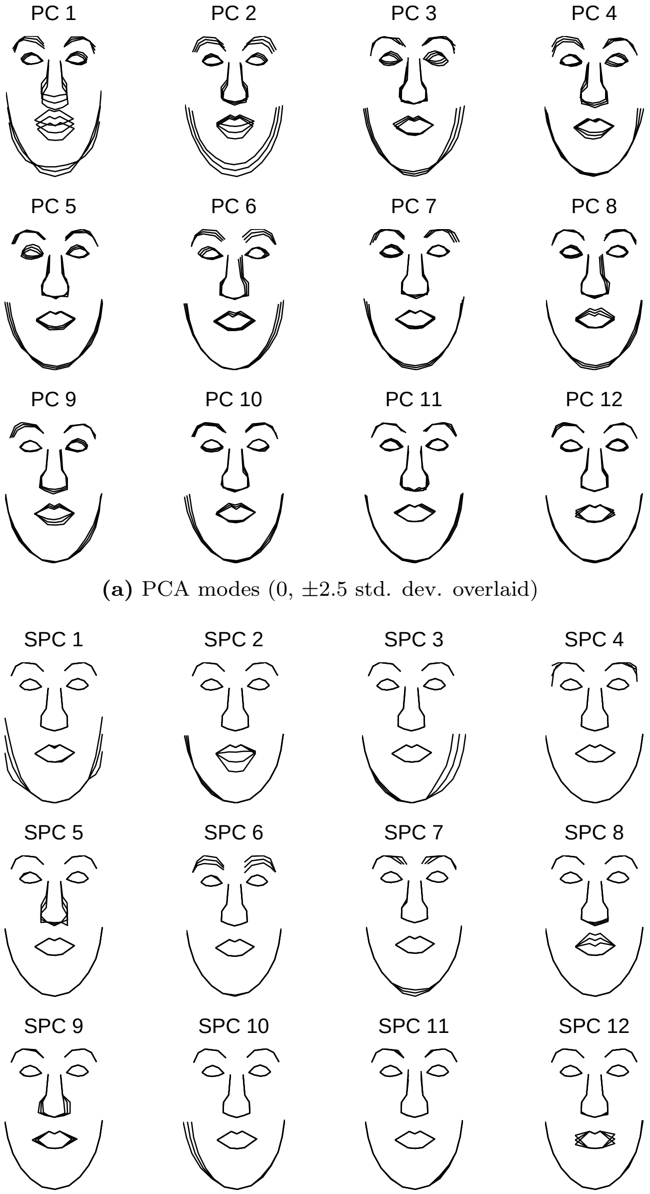
Table 6.1 shows variance proportions for ten modes of variation of the corpus callosum data set. The top row contains results for a regular PCA, while the second row represents sparse PCA using thresholding. The third row presents the adjusted variances of SPCA with no reordering of modes, and the results in the bottom row are for SPCA using the proposed forward selection-type rule for mode ordering. It is seen that reordering the modes increases the total explained variance and ensures that the variances are decreasing.

Variance (%)	PC 1	PC 2	PC 3	PC 4	PC 5	PC 6	PC 7	PC 8	PC 9	PC 10	$\sum$
PCA	43.27	18.55	13.74	7.71	4.93	2.01	1.58	1.36	0.98	0.78	94.92
threshold PCA	14.36	8.86	4.89	2.46	3.10	0.90	0.66	0.57	0.32	0.24	36.37
SPCA	13.21	7.51	5.18	3.44	2.27	0.60	0.26	0.90	0.11	0.03	33.50
reordered SPCA	15.12	7.88	7.10	3.37	1.35	1.06	0.42	0.32	0.11	0.03	36.77

Table 6.1: Explained proportion of variance for each mode and method for the corpus callosum data set. The last column shows the cumulative variance for all ten modes. Each sparse mode is set to affect 20 coordinates exactly (total 78), explaining the low proportions of variation.

The simplest alternative to SPCA is straight-forward truncation of loadings as described in the introduction. Some results of this scheme is found in Figure 6.4. The difficulties of this method are clear from these images; the modes of variation are merely pruned versions of those of regular PCA. Hence, the effects are scattered and hard to interpret.

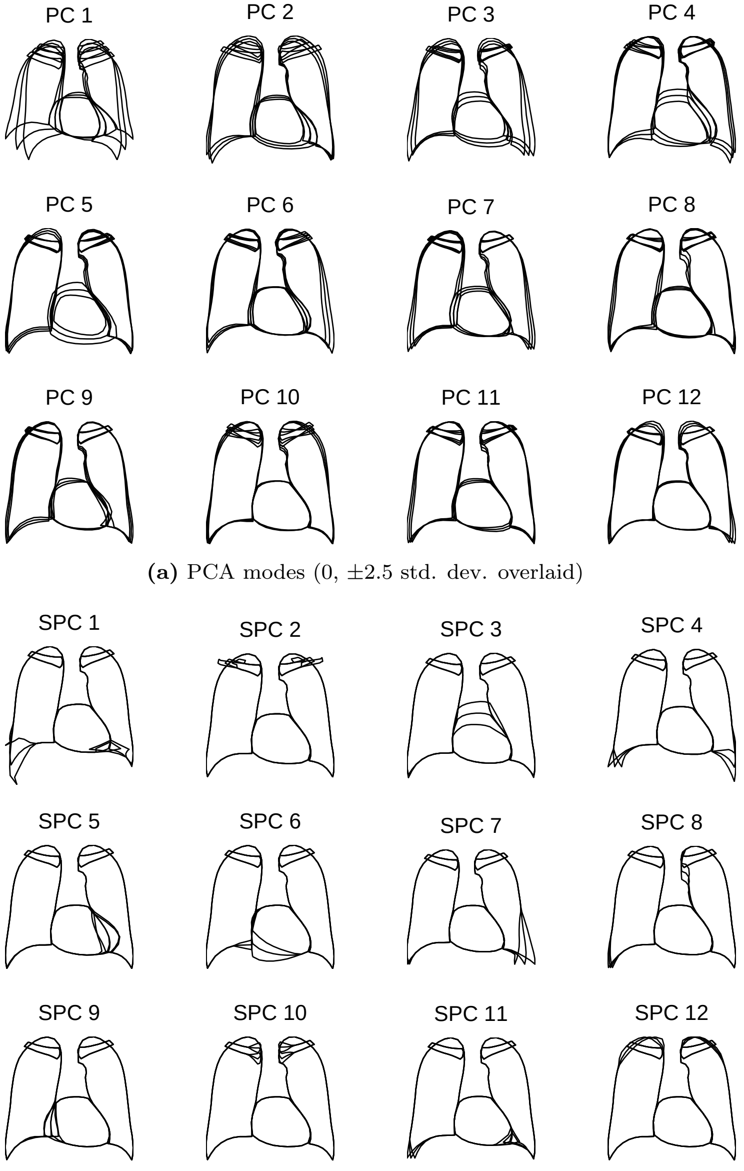
Figure 6.5 shows an important property of SPCA. The results vary slowly with values of λ, the weighting term on the L2 norm of the loadings. Here, vastly different values are chosen, but with similar results.



(a) PCA modes (0, ±2.5 std. dev. overlaid)

(b) SPCA modes (0, ±2.5 std. dev. overlaid)

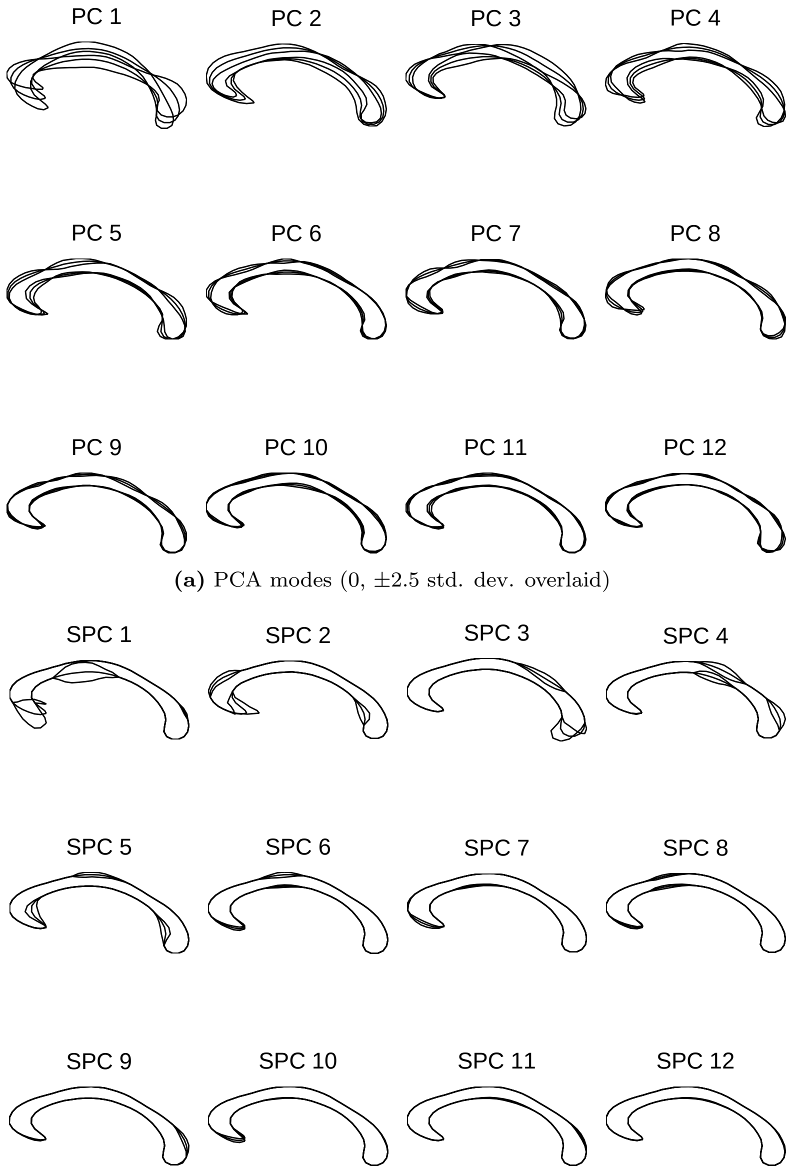
Figure 6.1: PCA (left) versus SPCA (right) shape models of the human face. Each mode describes an identifiable effect, such as smile/frown, nose size and shape, and eyebrow configurations.



(a) PCA modes (0, ±2.5 std. dev. overlaid)

(b) SPCA modes (0, ±2.5 std. dev. overlaid)

Figure 6.2: Lungs, heart and clavicles. Mode 3, 5, 6 and 9 depict the heart geometry while mode 8 describes the position of the aortic arch.



(a) PCA modes (0, ±2.5 std. dev. overlaid)

(b) SPCA modes (0, ±2.5 std. dev. overlaid)

Figure 6.3: PCA (left) and Sparse PCA (right) models of the corpus callosum brain structure.

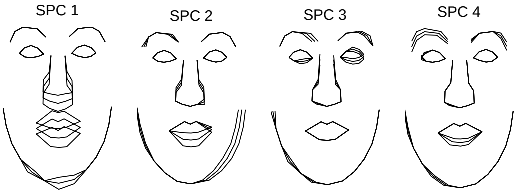


Figure 6.4: SPCA using simple thresholding. Although the same L1 constraint has been used, these images do not show the same amount of separation as those in Figure 6.1(b).

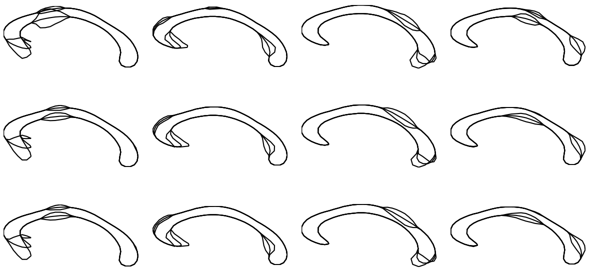
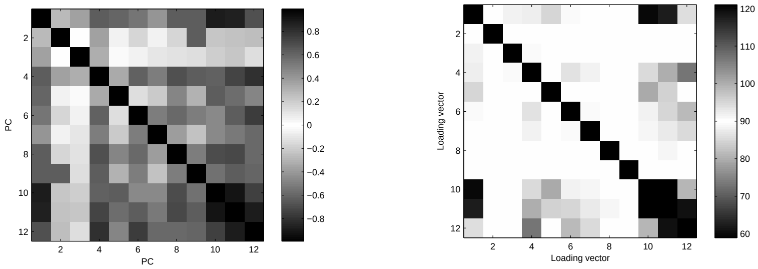


Figure 6.5: The first four modes of variation for the corpus callosum data set. Rows correspond to λ-values 0.001, 1, and 1000 (top to bottom). Note the insensitivity to values of λ.

The relatively strong correlations among the PCs produced by SPCA are evident in Figure 6.6 where correlations are plotted for the PCs, next to the angles between the loading vectors. The correlations become considerable, while most angles are in the vicinity of 90°, although with a few clear exceptions. These properties follow from the definition of the SPCA criterion as discussed earlier. The implication of the high correlations is that it becomes impossible to refer to one PC without referring to others. This is what motivates the discussion on ordering of modes.

The proposed method for ordering the principal components and the corresponding loading vectors proved successful in the majority of cases. To test the performance of the method, 100 random scores matrices were used as input and the average total amount of adjusted variance was measured in three different ways; using no reordering, the proposed method, and, by trying all possible combinations, the average maximal adjusted variance. This test was carried out for a number of combinations of the number of observations n, and the number of PCs k. Table 6.2 shows the complete set of results. The test matrices were all random, but to produce relevant scores matrices, a predefined covariance



(a) Correlations among the sparse principal components.

(b) Angles (degrees) between the sparse loading vectors.

Figure 6.6: Correlations of PCs and angles between loading vectors for the lungs data set, using the SPCA method. Regular PCA produces an orthonormal loading matrix and uncorrelated principal components. SPCA typically results in significant correlations, while angles are relatively close to 90°.

structure was used, and all variables had zero mean. The covariance structure from the SPCA calculations on the face data set was used. Similar results were obtained using other SPCA covariance matrices.

The **A** matrix is initialized to the first k loading vectors of a regular PCA. In the first SPCA iteration, the values of **B** will be influenced by this. However, as Figure 6.7 shows, as the iterations progress, the values of **B** converge to very different values; the resulting **B** seems to be independent of regular PCA. Tests with initialization of **A** to the identity matrix gives slightly different, but acceptable results.

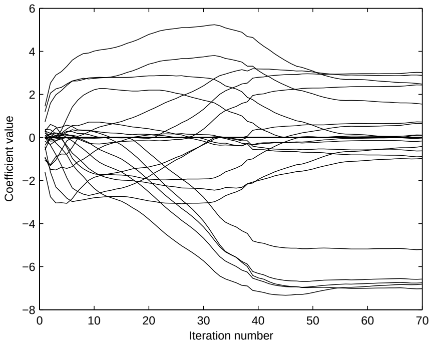


Figure 6.7: Coefficient values as functions of iteration number for the face data set. Typically, coefficients vary considerably before convergence.

	no reordering		
	$n = 10$	$n = 100$	$n = 1000$
$k = 3$	14.0 (98.8)	0	0
$k = 4$	46.0 (97.9)	16.0 (99.8)	5.0 (100.0)
$k = 5$	57.0 (97.6)	21.0 (99.8)	2.0 (100.0)
$k = 6$	76.0 (96.4)	39.0 (99.8)	4.0 (100.0)
$k = 7$	87.0 (96.8)	46.0 (99.8)	5.0 (100.0)
$k = 8$	95.0 (96.2)	50.0 (99.8)	2.0 (100.0)

	forward selection reordering		
	$n = 10$	$n = 100$	$n = 1000$
$k = 3$	1.0 (98.8)	0	0
$k = 4$	0	0	0
$k = 5$	0	0	0
$k = 6$	6.0 (98.7)	4.0 (99.8)	0
$k = 7$	9.0 (98.3)	6.0 (99.9)	0
$k = 8$	6.0 (99.1)	16.0 (99.5)	0

Table 6.2: Results of the proposed ordering method (right) versus no reordering (left) for k n-dimensional random PCs with a static covariance structure. Numbers represent the average proportion (%) over 100 trials where the optimal ordering was not found. The optimal ordering was established by an all-subsets calculation in each case. The parenthesized numbers denote the average proportion of maximal variance reached in cases of failure. Note that the average proportion of maximal variance over all trials is higher.

6.4 Discussion

The results presented in this article provide evidence that the presented SPCA algorithm is able to produce separate and easily identifiable modes of variation. We anticipate that SPCA will find good use in many clinical applications. In particular, the ability of SPCA to extract latent variables that are easily interpreted and visualized may help to understand the present variability. For instance, studies of atrophic processes in the human brain due to aging, dementia, Alzheimer's disease etc. may benefit from this treatment.

The algorithm requires $k + 1$ parameters, λ and one δ_i for each PC. From the results, it can be seen that the resulting loading vectors vary slowly with λ. The values of δ_i are, however, crucial. In this project, δ is set such that precisely 20 coordinates are affected in each mode, but any other choice is equally valid, and results would differ greatly. This makes the algorithm flexible, but parameter tuning requires knowledge of the problem at hand.

The computational complexity of SPCA for $n > p$ is at most $np^2 + mO(p^3)$ where m is the number of iterations before the algorithm converges. If $p > n$, the complexity is of order $mkO(pln + l^3)$ where l is the number of non-zero loadings, see [177] for a more thorough discussion. Typical computation times for the examples in this article are less than one minute on a standard laptop computer. However, the number of iterations grows rapidly with the number of PCs, and computation times for each elastic net problem grow with the number of non-zero loadings. Memory consumption depends mostly on the number of non-zero loadings, as the algorithm creates an $(l \times l)$ matrix in each iteration. This makes it difficult to handle e.g. texture data in this setting, where thousands of non-zero loadings may be of interest. The SPCA article [177] presents a designated SPCA algorithm where λ is set to infinity. Each elastic net computation is replaced by a single matrix multiplication, allowing for much lower memory consumption and computational complexity. Results on this extension are, however, yet to come.

This article presents one simple way of ordering principal components. This method sorts the PCs according to descending variance and maximizes the total explained (adjusted) variance in most cases. Table 6.2 shows that the fail rate increases dramatically with increasing k if no reordering is performed, especially for a low number of observations. With reordering, this effect is considerably lower. It is also apparent that the negative impact of a failure drops with the number of dimensions, n. The shape data used in this article has approximately $k = 12$ and $37 \leq n \leq 247$. Without reordering the PCs, there is a considerable risk that the resulting total adjusted variance is sub-maximal, and, as shown in Table 6.1, the individual variances may not be sorted in descending order.

The proposed method of measuring the performance of SPCA is convenient, as it resembles the results of a regular PCA. However, other ways of ordering modes can be beneficial.

Sparsity Modes can be ordered according to the amount of sparsity of the corresponding loading vectors. Several SPCA calculations may be carried out, each with different sparsity constraints. The modes are then ordered according to sparsity, e.g. from highly local modes to more global effects.

Spatially Modes may also be ordered according to spatial locality. The center of attention is calculated for each mode. These are then ordered along the contour of the object.

Entropy Although the resulting loading vectors are sparse, each mode may describe more than one effect. Using results from information theory, the *entropy* of each mode can be calculated, effectively giving a measure on the amount of clustering. Modes can be ordered accordingly, for instance going from low entropy, where a mode describes a single effect and has limited spatial extent, to high entropy, where effects are scattered and/or affect a larger proportion of the contour.

Combinations To obtain a more thorough library of modes, they may be ordered according to two criteria simultaneously and put in a two-dimensional grid. For instance, a combination of sparsity and a spatial ordering may be useful, especially in an exploratory setting. It is plausible that an examiner has some idea of the spatial location and extent of the relevant effect. The search may then be constrained by isolating relevant modes by defining for instance a rectangle in the two-dimensional grid.

6.5 Conclusion

This article has introduced sparse principal component analysis (SPCA) to medical shape modeling. Results, shown on three different data sets, provide some evidence that SPCA manages to isolate relevant sparse effects in each mode of variation. The inherent design of SPCA keeps loading vectors near orthogonal, while correlations between principal components are typically high. This motivates a discussion on the ordering of PCs. A method that orders the modes according to descending variance was discussed in detail and shown to improve the estimates of adjusted variances notably, while a few other possibilities where mentioned briefly. The convergence of SPCA was shown to be irregular and slow at times, but results are superior to those of the more straight-forward approaches, such as thresholding of loading vectors.

Future work includes using SPCA for other applications, such as exploratory analysis of fMRI data. The main obstacle in such analyses is the large number of variables. An examination of the discriminative power of SPCA calculations in medical shape modeling is also planned.

Source code for the statistics software S-Plus and its freeware sibling R has been written and made available by H. Zou and T. Hastie, see `www.r-project.org`. The first author of this article has made a corresponding implementation for Matlab, available on `www.imm.dtu.dk/~kas/software/spca/`.

Acknowledgments

Dr. Bram van Ginneken, Image Sciences Institute, University Medical Center Utrecht, kindly provided the lung annotations used in this study. Dr. Ginneken also assisted with the anatomical interpretation. Charlotte Ryberg and Egill Rostrup, The Danish Research Centre for Magnetic Resonance, Copenhagen University Hospital, Hvidovre, kindly provided the MRIs used to produce the corpus callosum annotations. Hui Zou and Trevor Hastie are acknowledged for making the source code of the SPCA algorithm publicly available. K. Sjöstrand was supported by The Technical University of Denmark, DTU. M. B. Stegmann was supported by The Danish Research Agency, grant no. 2059-03-0032.

Chapter 7

Sparse Decomposition and Modeling of Anatomical Shape Variation

Karl Sjöstrand, Egill Rostrup, Charlotte Ryberg, Rasmus Larsen, Colin Studholme, Hansjoerg Baezner, Jose Ferro, Franz Fazekas, Leonardo Pantoni, Domenico Inzitari, and Gunhild Waldemar

Abstract

Recent advances in statistics have spawned powerful methods for regression and data decomposition that promote sparsity, a property that facilitates interpretation of the results. Sparse models use a small subset of the available variables, and may perform as good as or better than their full counterparts if constructed carefully. In most medical applications, models are required to have both good statistical performance and a relevant clinical interpretation to be of value. Morphometry of the corpus callosum is one illustrative example. This paper presents a method for relating spatial features to clinical outcome data. A set of parsimonious variables is extracted using sparse principal component analysis, producing simple yet characteristic features. The relation of these variables with clinical data is then established using a regression model. The result may be visualized as patterns of anatomical variation, related to clinical outcome. In the present application, landmark-based shape data of the corpus callosum is analyzed in relation to age, gender, and clinical tests of walking speed and verbal fluency. To put the data-driven sparse principal component method into perspective we consider two alternative techniques, one where features are derived using a model-based wavelet approach, and one where the original variables are regressed directly on the outcome.

7.1 Introduction

Traditional morphometric investigations in medicine make use of simple metrics such as volume, area, length and various ratios to evaluate relations between structure and function. The outcomes of such studies provide the examiner with an indication of the characteristic anatomy of a clinical population, or spatial features related to for example pathology. More intricate features provide more information for interpretation, but require a more detailed hypothesis of the process under study. For a clinical investigation that is exploratory in nature, it makes sense to use an exploratory method to extract features. Such variables should ideally have a clear relation to the relevant morphology, while imposing as few assumptions on the data as possible. During the last two decades, methods for extracting more complex representations of anatomy from image data of increasingly higher resolution have evolved. This has led to the development of methods that allow for the computation of more abstract features such as the mean shape and typical deformation patterns according to the latent shape distribution. Derived variables may be concretized as examples of anatomy, which allows for more detailed investigation and interpretation. Furthermore, the relationship between structural and clinical variables can be analyzed in a formal statistical framework, making the investigation of certain clinical hypotheses possible.

The challenge posed by increasingly complex anatomical representations is to extract physically intuitive parameterizations of spatial variation. Conventional statistical techniques tend to extract global decompositions of spatial data. However, the effects of many biological processes of interest are expected to be anatomically localized, even if the particular location, extent and frequency are usually unknown.

This paper presents a methodology in which a statistically defined spatially localized representation of anatomy is automatically extracted. The approach is built on a generic statistical method known as sparse principal component analysis. The paper further describes a way of relating these spatial variables to some clinical outcome variable, producing a characteristic deformation of the present anatomy and indicating its statistical relevance.

Related Work

Increasingly advanced techniques for analyzing the shape of anatomical structures have emerged during the last two decades [11]. A suitable choice of shape parameterization is crucial to ensure a correct and efficient analysis, and several techniques have been developed to accurately describe the variability of human anatomy. These techniques include corresponding landmarks [12, 27, 38], representations in the frequency domain in two [136] and three [15] dimensions, skeleton-based techniques [10, 51], distance transforms [13, 92], and deformation fields resulting from the registration of a set of images to a common reference [3, 147].

Most of these methods produce a large number of spatial features. To devise a more manageable model, the features are often arranged into groups according to some spatial or statistical criterion. [27] pioneered the use of principal component analysis (PCA) to decompose sets of landmarks. This provides compact and powerful models for shape-driven segmentation and registration. A more recent example is [35], who decomposed sets of landmarks with optimized correspondences using PCA, and used the resulting shape features in a classification study of the hippocampus. PCA has also been used to decompose other shape descriptors. For instance, [77] presented a framework similar to that of [27] for frequency domain descriptors applied to the segmentation of the hippocampus, and [87] applied PCA to deformation fields extending throughout the entire brain.

The use of PCA as an explanatory basis for interpretation in clinical applications has been limited ([114] is one exception). While PCA is an excellent tool for efficient data representation, the global nature of the derived variables makes interpretation difficult. This motivates the use of an extension to PCA known as sparse PCA (SPCA). While the variables derived by PCA consist of linear combinations of *all* original variables, SPCA forces the weights on some variables towards zero, while others are adjusted to uphold the variance-maximizing properties of PCA. The idea in studies of anatomy is that each variable will describe a spatial pattern of variation that has a simple structure and a clinically relevant interpretation [134]. Although conceptually simple, the calculation of SPCA has proved difficult and several algorithms have been proposed [18, 31, 61, 74, 99, 123, 166]. The approach advocated here was developed by [177] and formulates PCA as a regression problem, using a recent variable selection algorithm [175] to achieve sparsity. The selection of important variables is achieved by penalization of the weights on each variable using the ℓ_1 norm, a methodology introduced with the LASSO regression framework [157], along with a method for its efficient computation [40].

Examples of other statistical decomposition techniques used in shape analysis are factor analysis [95], varimax rotated principal components [139], and independent component analysis [161]. The latter two typically produce approximately sparse representations, but lack the flexibility of most SPCA implementations.

In medical image analysis, the use of variable selection algorithms to aid interpretation is gaining momentum. [171], employed a support vector machine classification algorithm that incorporates variable selection to select subregions of the hippocampus that separates schizophrenic patients from normal controls. A similar algorithm was used by [146] on SPECT imagery to find regions of the brain that differentiate between healthy subjects and patients with Alzheimer's disease. [43] used variable selection on deformation field data in a study of schizophrenia.

The methodology introduced in this paper is applied to a data set of 569 outlines of the corpus callosum (CC) brain structure, obtained from a study on atrophy in an elderly population [110]. The CC provides an illustrative example of a structure that may benefit from a localized analysis. The white matter fibers defining the CC are organized according to an anterior-posterior topographical organization; tissue loss and discrepancies can therefore be expected to be constrained to specific regions [70]. The CC is perhaps the most popular single nervous structure for morphometric analysis and a wide range of applications in shape analysis exist. [12] characterized deformations of the CC using partial thin-plate spline warps. [33], [94] and [39] used deformation field features to find gender differences in the CC. [52, 53] takes a classification approach to finding anatomical discrepancies between populations where group differences are characterized by the gradient of the classifier function and applies the method to a study of the CC in affective disorder. [75] extract predefined global and local shape features of the CC using a multi-scale medial shape representation. The features are used for classification of schizophrenic and normal subjects.

The advantage of the method presented in this paper over previous work is the extraction of interpretable localized features governed by few and weak assumptions. The central assumption is on the extent of the deformations, however, we propose to alleviate this assumption by extracting features on several scales.

To put the SPCA method into perspective, we provide a comparison with two alternative analysis methods, one where the original shape features (landmarks) are analyzed directly to provide a sparse representation of anatomy. The second method challenges a potential shortcoming of a data-driven process such as PCA or SPCA in that minor but clinically relevant variation may be omitted. We therefore include a model-based method for decomposition based on the wavelet transform. Multi-scale representation of curves using the wavelet transform has found applications in both computer graphics [118] and image analysis [34].

The wavelet transform decomposes the anatomy into coefficients of both scale and localization [32] and offers a sparse orthogonal shape basis with acceptable interpretability.

Characteristic deformation patterns of the CC are derived for four different clinical variables. Focus is on shape differences of the CC due to gender [2, 9, 20, 33, 39, 94, 167]), but results are also given for age effects, verbal fluency and walking speed. Using the same data set, atrophy of the CC has previously been shown to correlate with general cognitive and physical decline [70, 126].

7.2 Methods

To understand and quantify a complex process such as the variability of anatomy, one has to balance a trade-off between a model that is both general and compact. The first property means that it should be possible to model any conceivable deformation pattern, while the second property ensures that the number of variables used to do so is kept small, allowing more power to the subsequent statistical analysis. If the intended use of the model goes beyond prediction, interpretability adds to this list of requirements. Many anatomical processes are expected to be localized, leading to high correlations between spatially neighboring features. This property can be used to derive variables where a single variable may describe deformations across several features in an anatomically plausible fashion. Furthermore, restricting the analysis to relevant variation only, the number of variables can be reduced. In the following, we will review two methods for deriving such variables.

7.2.1 Principal Component Analysis

The first method is perhaps the most well-known and widely used method for data decomposition in general; principal component analysis (PCA). To introduce the method, as well as the notation and terminology used throughout the rest of this paper, a brief explanation will be given here.

PCA takes a mean centered $(n \times p)$ data matrix $\mathbf{X}$, n being the number of observations and p being the number of variables, and transforms it by $\mathbf{Z} = \mathbf{XB}$ such that the derived variables (the columns of $\mathbf{Z}$) are uncorrelated and correspond to directions of maximal variance in the data. The derived coordinate axes are the columns of $\mathbf{B}$, called *loading vectors* with individual elements known as *loadings*. These are at right angles with each other; PCA is simply a rotation of the original coordinate system, and the $(p \times p)$ loading matrix $\mathbf{B}$ is the rotation matrix. The new variables (the columns of $\mathbf{Z}$) are known as principal components (PCs). Usually only the first k components, $k < p$, are retained since these explain the majority of the sample set variance. This makes $\mathbf{Z}$ $(n \times k)$ and $\mathbf{B}$ $(p \times k)$. The loading matrix can be calculated using singular value decomposition of the data matrix $\mathbf{X}$ or by eigenanalysis of the corresponding covariance or correlation matrix.

7.2.2 Sparse Principal Component Analysis

Sparse PCA (SPCA) can be described as an extension of PCA, where a constraint on the number of non-zero loadings is added. The recent development in statistical methods for variable selection in regression has resulted in an SPCA approach described by Zou and Hastie [177]. This method is used throughout this paper and the idea will be described here in brief. For a complete treatment, consult [177] and the preliminary papers [40, 157] and [175]. Refer to [134] for an introduction on using SPCA to decompose shape data.

The regression methods used in the calculation of SPCA all originate from ordinary least squares (OLS) approximations. The independent variable $\mathbf{y}$ is approximated by a linear combination of the dependent variables in $\mathbf{X}$. The coefficients for each variable (column) of $\mathbf{X}$ are contained in $\mathbf{b}$.

$$\mathbf{b}_{\text{OLS}} = \arg\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2, \tag{7.1}$$

where $\|\cdot\|$ represents the ℓ_2 norm. This is the best linear unbiased estimator given a number of assumptions, such as independent and identically distributed (i.i.d.) residuals. However, if some bias is allowed, estimators can be found with lower mean square error than OLS when tested on an unseen set of observations. A common way of implementing this is by introducing some constraint on the coefficients in $\mathbf{b}$. The methods described here use constraints on either the ℓ_1 norm or the ℓ_2 norm of $\mathbf{b}$, or both. Adding the ℓ_2 constraint gives

$$\mathbf{b}_{\text{ridge}} = \arg\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \lambda\|\mathbf{b}\|^2. \tag{7.2}$$

This is known as ridge regression [65]. Sufficiently large values of λ will shrink the coefficients of $\mathbf{b}$. The shrinkage introduces bias, but lowers the variance of the estimates. Careful selection of λ may lead to improved prediction accuracy, but of more interest here are the improved numerical properties, making estimation in cases where $p > n$ feasible [59]. Replacing the ℓ_2 norm in the constraint with the ℓ_1 norm gives

$$\mathbf{b}_{\text{LASSO}} = \arg\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \delta\|\mathbf{b}\|_1, \tag{7.3}$$

where $\|\mathbf{b}\|_1 = \sum_{i=1}^{p} |b_i|$. This is the LASSO method [157]. Using the ℓ_1 norm not only shrinks the coefficients, but drives them one by one to exactly zero as δ increases. This implements a form of variable selection, as minor coefficients will be set to zero in a controllable fashion, while the remaining coefficients will be used to minimize the size of the regression residuals.

A third possibility is to use a combination of the constraints from ridge regression and the LASSO. This approach is known as the elastic net [175] and has the form

$$\mathbf{b}_{\text{EN}} = \arg\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \lambda\|\mathbf{b}\|^2 + \delta\|\mathbf{b}\|_1. \tag{7.4}$$

The main benefit of the elastic net is that it better handles cases where $p > n$. The elastic net can be formulated as a LASSO problem on augmented variables, and is solved using the same algorithm, outlined below.

Ordinary least squares and ridge regression have closed-form solutions, that is, $\mathbf{b}_{\text{OLS}}$ and $\mathbf{b}_{\text{ridge}}$ can be expressed as functions of the random variable $\mathbf{y}$; $\mathbf{b}_{\text{OLS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ and $\mathbf{b}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$. This is not true for the LASSO and elastic net methods. For many years, LASSO solutions were found using standard optimization techniques, which made for long computation times. In 2002, [40] published a report on a new regression method coined least angle regression (LARS). Although conceptually different, the method is shown to be very similar to LASSO, and through a small modification, the exact LASSO solution can be computed. The method is built on a powerful geometric framework, through which a computationally thrifty algorithm is conceived. The paper shows that the coefficients $\mathbf{b}$ are piecewise linear with respect to the regularization parameter δ, with breakpoints as variables enter or leave the model. The breakpoints can be established using standard linear algebra. Using this property, the entire regularization path can be computed. Starting with the empty model ($\mathbf{b} = \mathbf{0}$), variables are added and occasionally subtracted as δ grows until all variables are non-zero and the full least squares solution is reached. Hereby, the LARS path algorithm returns the solutions for all possible values of δ. The computational cost for obtaining the entire LASSO regularization path is the same as for a single least squares fit.

PCA and SPCA are strongly related to these regression algorithms. One way of describing PCA using regression is by treating each principal component as a response vector and regressing this on the p variables using ridge regression,

$$\hat{\mathbf{b}}_i = \arg\min_{\mathbf{b}_i} \|\mathbf{z}_i - \mathbf{X}\mathbf{b}_i\|^2 + \lambda\|\mathbf{b}_i\|^2. \tag{7.5}$$

The minimizing coefficient vector $\hat{\mathbf{b}}_i$ normalized to unit length is exactly the ith principal loading vector, independent of the choice of λ [177]. A direct approach to sparse PCA is obtained by adding the ℓ_1 (LASSO) constraint,

$$\hat{\mathbf{b}}_i = \arg\min_{\mathbf{b}_i} \|\mathbf{z}_i - \mathbf{X}\mathbf{b}_i\|^2 + \lambda\|\mathbf{b}_i\|^2 + \delta\|\mathbf{b}_i\|_1. \tag{7.6}$$

The regression procedure will calculate a loading vector $\hat{\mathbf{b}}_i$ such that the resulting PC is close to $\mathbf{z}_i$ while being sparse. The weakness of this approach is that all solutions are constrained to the immediate vicinity of a regular PCA. A better approach would be to approximate the *properties* of PCA, rather than its exact results. Specifically, the columns of the loading matrix $\mathbf{B}$ should be near orthogonal and describe directions of high variance in the data set. Zou and Hastie propose a problem formulation called the *SPCA criterion* [177] to address this.

$$(\hat{\mathbf{A}}, \hat{\mathbf{B}}) = \arg\min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i\|^2 + \lambda \sum_{j=1}^{k} \|\mathbf{b}_j\|^2 + \sum_{j=1}^{k} \delta_j \|\mathbf{b}_j\|_1$$

$$\text{subject to} \quad \mathbf{A}^T\mathbf{A} = \mathbf{I}_k \tag{7.7}$$

To clarify this expression, it will be broken down into components. First, $\mathbf{B}^T\mathbf{x}_i$ takes the variables of observation i and projects them onto the principal axes (loading vectors) of $\mathbf{B}$. Note that $\mathbf{x}_i$ denotes the ith column of $\mathbf{X}^T$. Only k PCs are retained, meaning that some information is lost in this transformation. Next, $\mathbf{A}\mathbf{B}^T\mathbf{x}_i$ takes the scores of $\mathbf{B}^T\mathbf{x}_i$ and transforms them back into the original space. The orthogonality constraint on $\mathbf{A}$ makes sure $\mathbf{B}$ is near orthogonal. The whole term $\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i\|^2$ measures the reconstruction error. The remaining constraints are the same as for elastic net regression, driving the columns of $\mathbf{B}$ towards sparsity and ensuring good numerical properties in cases where $p > n$. Some further insight into this criterion is given by considering the loss function alone, with the additional constraint $\mathbf{B} = \mathbf{A}$,

$$\hat{\mathbf{A}} = \arg\min_{\mathbf{A}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{A}^T\mathbf{x}_i\|^2 \quad \text{subject to} \quad \mathbf{A}^T\mathbf{A} = \mathbf{I}_k. \tag{7.8}$$

The minimizer of this function is given by the first k loading vectors of a standard PCA; this equation is in fact the basis for a derivation of PCA [59] other that the standard variance-maximization approach. One of the key results of the SPCA paper [177] is that the constraint $\mathbf{B} = \mathbf{A}$ can be omitted given the addition of an ℓ_2 penalty term,

$$(\hat{\mathbf{A}}, \hat{\mathbf{B}}) = \arg\min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i\|^2 + \lambda \sum_{j=1}^{k} \|\mathbf{b}_j\|^2 \quad \text{subject to} \quad \mathbf{A}^T\mathbf{A} = \mathbf{I}_k \tag{7.9}$$

The columns of $\mathbf{B}$ (normalized to unit length) will still give the exact PCA solution. The SPCA criterion then augments this formulation by the addition of the ℓ_1 term, making it possible to estimate loading vectors that range from the results of a standard PCA to various sparse approximations.

The constraint weight λ must be chosen beforehand and has the same value for all PCs, while δ may be set to different values for each PC, offering good flexibility. The level of sparsity can also be defined by specifying a target number of active variables. This is done by terminating the elastic net estimation when a suitable number of variables have entered the model. This stopping criterion is very useful in practice.

Equation 7.7 resembles the elastic net formulation, but there is a significant difference. Instead of estimating a single coefficient vector, this problem has two matrices of unknown coefficients, $\mathbf{A}$ and $\mathbf{B}$. A reasonably efficient optimization method for minimizing the SPCA criterion is presented in [177]. First, assume $\mathbf{A}$ is known. By expanding and rearranging Equation 7.7, it is shown that $\mathbf{B}$ can be estimated by solving k independent elastic net problems, one for each column of $\mathbf{B}$ (loading vector). Referring to the elastic net formulation in Equation 7.4, the predictor matrix is $\mathbf{X}$ as usual while $\mathbf{y} = \mathbf{X}\mathbf{a}_i$, where $\mathbf{a}_i$ is the ith column of $\mathbf{A}$. On the other hand, if $\mathbf{B}$ is known, $\mathbf{A}$ can be calculated using a singular value decomposition; if $\mathbf{X}^T\mathbf{X}\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, then $\mathbf{A} = \mathbf{U}\mathbf{V}^T$. Since both matrices are unknown, an initial guess is made and $\mathbf{A}$ and $\mathbf{B}$ are estimated alternately until convergence. The standard option is to initialize $\mathbf{A}$ to the loadings of the k first ordinary principal components.

7.2.3 Statistical Analysis

The goal of the analysis is to determine the relationship between the derived variables (loading vectors) and some clinical outcome variable. Clinical variables are here assumed to consist of a single score for each patient (e.g. age) and are therefore n-dimensional. However, methods such as PCA and SPCA derive new variables that are p-dimensional, that is, each variable can be interpreted as a perturbation of the mean observation. As a preliminary step, the presence of each PCA/SPCA variable in each subject must be measured. We propose to do this via univariate regression. The following model formulates the idea, where the presence z of deformation mode j is determined for the shape corresponding to subject i (row vector) by,

$$\mathbf{x}_i^T = z\mathbf{b}_j + \boldsymbol{\varepsilon}. \tag{7.10}$$

The loading vectors $\mathbf{b}_j$ have unit length for both PCA and SPCA, yielding the least squares estimate $z = \mathbf{x}_i\mathbf{b}_j$. This is simply the (i, j)th entry of the $n \times k$

scores matrix $\mathbf{Z}$, which, as described in Section 7.2.1, is estimated by $\mathbf{Z} = \mathbf{XB}$, also for SPCA. The presence z can be interpreted as a measure of correlation between shape i and deformation j.

The scores matrix provides k n-dimensional variables that can be related to clinical outcome. In this paper, we propose to establish this relation via a series of univariate tests. This approach is similar to those used in e.g. analysis of functional images and deformation/tensor based analysis [39], where separate tests are performed at each voxel of an image volume. The statistical properties of the scores vectors are often better suited for a regression analysis than clinical variables, which may be categorical or ordinal (ordered categorical). We therefore assign the scores vector as the outcome variable. The test for a relationship between spatial variable i and the clinical outcome $\mathbf{y}$ becomes

$$\mathbf{z}_i = \beta_i \mathbf{y} + \boldsymbol{\varepsilon}. \tag{7.11}$$

Confounding variables enter the model on the right hand side as covariates. This simple regression model is solved using the least squares criterion, providing access to a range of statistical properties, most notably t-scores with corresponding p-values, measuring the probability that a significant relation is declared when the variables are in fact unrelated.

Using the above analysis, the relationship between the outcome and each spatial variable is established. A complication with this approach is that significance levels should be adjusted for the number of comparisons performed. Bonferroni correction provides one simple procedure, where any test probabilities (p-values) are multiplied by the number of tests performed. This provides strong control over the family-wise (type-I) error rate – the probability that one or more tests are falsely rejected is less than the nominal significance level α. However, this procedure is generally too conservative, leading to unnecessarily high p-values. A more powerful alternative, also with strong control over type-I errors, is provided by nonparametric permutation testing procedures. The specific method used here is described in detail in e.g. [104], and is based on finding the empirical distribution of a *maximal statistic*. First, we will review that basics of permutation testing, and then briefly explain how this may be used to adjust a set of p-values for multiple comparisons.

The idea of permutation testing is that if two variables are in fact unrelated, then the results (for instance from a correlation or regression analysis) should not change notably even though the elements of one of the variables have been randomly shuffled around [36]. By permuting the dependent variable in the regression analysis in Equation 7.11 R times, where R is some large integer number ($R > 999$), an estimate of the empirical distribution function (EDF) under the null hypothesis is obtained as the histogram of the corresponding t-statistics of the independent variable of interest. Calculating the proportion of t-values exceeding the t-value obtained from the original (non-permuted) regression analysis provides a nonparametric estimate of the p-value of the independent variable. Providing that the standard assumptions of the regression analysis in (7.11) hold, these p-values will be in close agreement with those obtained from a classical, parametric analysis.

One advantage of this non-parametric approach is that it provides additional information that can be used to adjust the obtained p-values for multiple comparisons. This information comes in the form of the distribution of the maximal statistic. This statistic consists of the maximal absolute t-value over all tests for each permutation. For the ith repetition, we denote this value $t_i^{\max}$. After R repetitions, an approximation of the EDF for the maximal statistic is obtained. The *critical value* is defined as the $\lfloor \alpha R \rfloor + 1$ largest member of this distribution. Any t-values exceeding this value are deemed significant at the α level. In practice, we do not need to compute the critical value. An adjusted p-value can be obtained directly from the EDF of the maximal statistic as the proportion of values exceeding the t-value t from the original regression analysis. Formally this corresponds to $p_{\text{adjusted}} = (1 + \#\{t_i^{\max} > t\})/(R+1)$, where # denotes the number of elements in a set [36].

7.2.4 Application to Shape Analysis

In this section, we will describe more specifically how the methods outlined above are applied to landmark based shape analysis. We adopt the definition of shape of Kendall [78], stating that shape information is what remains in a data set, when translational, rotational and scaling effects have been filtered out. The shapes are therefore aligned using a general Procrustes analysis [38]. The removal of scale differences deserves some attention in this application. Many anatomical discrepancies, age related changes is one example, are likely to include a component of pure scale. Obviously, a sparse decomposition is not suitable for describing global properties with preserved interpretability, which is why we recommend removing such differences. In the subsequent analysis of the results, this fact must be taken into consideration. A separate analysis of area/volume differences may be used to complement the study of local shape variability.

PCA Application

Global patterns of shape variability are obtained through a principal component analysis, performed by a singular value decomposition of the centered data matrix $\mathbf{X}$. This matrix consists of the Procrustes aligned shapes, where the mean shape has been subtracted from each row. Typically, all landmarks contribute to the variance of the data set, meaning that each new variable $\mathbf{b}_i$ (column of $\mathbf{B}$) will affect the entire outline at once. The usual practice is to truncate the set of variables to account for e.g. 95 % of the total variation. This can be done easily, since the variance explained by $\mathbf{b}_i$ is given directly by the ith eigenvalue of $\text{cov}(\mathbf{X})$. This reduction has a number of advantages. For instance, it excludes noisy (wiggly) deformation modes and it simplifies and strengthens the subsequent statistical analysis.

SPCA Application

As for PCA, SPCA is applied to the aligned and centered shapes contained in the data matrix $\mathbf{X}$. A number of parameters govern the results. Also akin to PCA, a choice must be made on the number of variables to retain. Unlike PCA, this must be done in advance here. A rough number is provided by the number of variables deemed significant in the PCA analysis, since when estimating an excess of variables, the SPCA algorithm tends to produce highly correlated variables. The next parameter to set is λ in relation to the ℓ_2 constraint. Empirical evidence [134] supported by some theoretical results [177] suggest that the results are largely independent on the specific choice of this parameter. Typically, it is set to a small positive value to ensure good numerical properties. Finally, the parameters δ_j must be set, governing the amount of sparsity of the decomposition. This choice is dependent on the anatomical scale of interest, and must be carefully chosen for each application. For many purposes, δ_j will be equal for all j, resulting in the same deformation size for each $\mathbf{b}_i$.

Tabulation and Visualization

The most thorough way of presenting the results is a table showing each deformation mode, and the significance level for each of these associated with each tested clinical outcome variable. Such a presentation minimizes the risk of misleading the reader, but may also become time consuming and complex to draw conclusions from. In order to construct a sample anatomy related to a specific outcome variable, we suggest creating a compound deformation of a template shape (for most purposes, the mean shape). Each deformation mode exceeding the nominal significance level α contributes to this deformation with strength proportional to its corresponding β (regression coefficient) value. If both the spatial and clinical variables are standardized (zero mean, unit variance) prior to the regression analysis, the coefficients can be interpreted as the change (in standard deviations) in the spatial (response) variable introduced by a unit change in the clinical variable. For interpretational purposes, the use of the β values directly as weights on the various deformations may not produce an anatomically meaningful pattern. Therefore, we instead choose to normalize the β values within the group of spatial variables being tested such that the maximal point-to-point distance is set to an appropriate value. The relative sizes of the deformations will still be correct using this method, but the absolute strengths of the relationships are lost, a fact that must be taken into consideration when analyzing the results. This approach is used in the display of deformations in this paper.

7.2.5 Alternative Methods

This section provides a brief explanation of two alternative methods for relating clinical outcome to localized representations of anatomy. One represents a simple and direct analysis, while the other provides a model-based alternative to the data-driven decomposition of PCA/SPCA.

Direct Analysis of Original Variables

PCA derives variables that capture global properties of the relevant anatomy, while SPCA provides a more localized alternative. If the analysis is made increasingly localized, the derived variables will in the limit consist of a single component (x or y coordinate in the case of 2D shape analysis). This results in an immediate and simple approach where the original spatial variables enter Equation 7.11 one by one on the left hand side, and their individual relation to the clinical outcome is established.

Decomposition using the Wavelet Transform

The pitfall of using subspace techniques such as PCA is that subtle but interesting information may be lost. A minor deformation may be strongly related to a clinical variable, but since the contribution to the sample variance is low, the effect may not be modeled or simply discarded. It is therefore of interest to find a basis where each variable is clinically relevant and all the variance of the original data set is preserved. The wavelet transform may provide one such basis.

A wavelet is a waveform of limited duration. The wavelet transform breaks the original signal into scaled and translated versions of a predefined *mother wavelet* [32]. The original signal is first divided into two parts of low and high scale. These representations are known as the approximation (coarse scale) and the detail (fine scale). The approximation is then further divided in an equivalent fashion, and the process is repeated a suitable number of times. This yields a hierarchy of coefficients organized in a tree structure according to scale and location depicted in Figure 7.1. Each wavelet coefficient represents a deformation across several landmarks that is localized in both scale (spatial extent) and position along the outline. The first order *coiflet* wavelet is used here, which was determined suitable for describing local shape changes because of its low complexity and high symmetry. This particular wavelet is orthogonal, meaning that the variance and structure of the original shape data are preserved. In the present analysis, x- and y-coordinates are treated separately as one-dimensional periodic functions. The two resulting wavelet coordinate vectors are concatenated into a single observation. This process is repeated for all shapes in the data set, producing a set of variables of the same size as the original data.

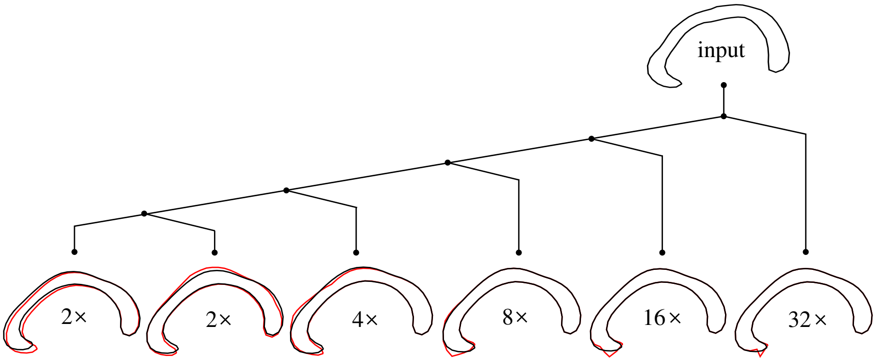


Figure 7.1: The hierarchical representation of a shape in the wavelet domain. Numbers represent the number of wavelet coefficients on each level. The leftmost branch represents the approximation, while other branches correspond to detail at different scales. At each branch, one example of the resulting shape deformation is shown in red with the mean shape (black) as reference.

7.3 Results

The proposed method was applied to a large data set of two-dimensional outlines of the corpus callosum (CC) brain structure. The corpus callosum is the band of fibers connecting the hemispheres of the brain. These fibers are organized in the approximate anterior to posterior topographical organization depicted in Figure 7.2. The data set is part of the longitudinal LADIS (Leukoaraiosis And DISability in the elderly) study, involving twelve European countries and more than 700 patients. Refer to [110] for a complete description of this study and the project protocol. This paper presents a cross-sectional study based on baseline data with 569 (312 female) subjects. The shape data was extracted from the baseline MR images (3D sagittal or coronal T1-weighted MPRAGE, voxel size 1×1×1 mm). In the mid-sagittal plane, the CC was registered using a learning-based active appearance model [29, 140], trained on 62 CC examples, each manually annotated with 78 corresponding landmarks. The automatic registration was followed by manual inspection and correction by an expert reviewer, unaware of any clinical status [126].

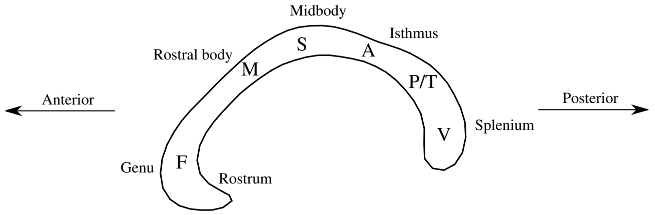


Figure 7.2: Subregions and approximate fiber connectivity of the corpus callosum. The connectivity labels are F (frontal), M (motor), S (somatosensory), A (auditory), P/T (parieto-temporal), and V (visual). This image is adapted from [172] and is based on a post-mortem study [167].

Initially, as simple test was performed to see whether the shape of the male and female corpus callosum differed significantly. The full Procrustes distance was used to measure the discrepancy between two shapes. This measure is a normalized sum of point-to-point distances between the aligned shapes $\mathbf{w}$ and $\mathbf{y}$ in complex notation [38],

$$d_F(\mathbf{y}, \mathbf{w}) = \sqrt{1 - \frac{\mathbf{y}^*\mathbf{w}\mathbf{w}^*\mathbf{y}}{\mathbf{w}^*\mathbf{w}\mathbf{y}^*\mathbf{y}}}, \tag{7.12}$$

where $\mathbf{w}^*$ is the transpose complex conjugate of $\mathbf{w}$. The Procrustes distance between the male and female mean shapes was found to be $d_F(\bar{\mathbf{x}}_{\text{male}}, \bar{\mathbf{x}}_{\text{female}}) = 0.0167$. Placing this value on the null distribution estimated by calculation of the Procrustes distance based on a large number of permutations of the data set (cf. [36, 104] and Section 7.2.3), the shapes were found to differ significantly ($p = 0.0015$, $R = 9999$ repetitions). Figure 7.3 shows the female versus the male mean CC shapes and the corresponding null distribution. The red dashed line indicates the nominal Procrustes distance $d_F(\bar{\mathbf{x}}_{\text{male}}, \bar{\mathbf{x}}_{\text{female}})$.

The described algorithm for sparse principal component decomposition was applied to the Procrustes aligned shape data. The anatomical scale of any deformations related to the clinical outcome variables of interest is unknown. Three decompositions on three different scales were therefore calculated. The extent of

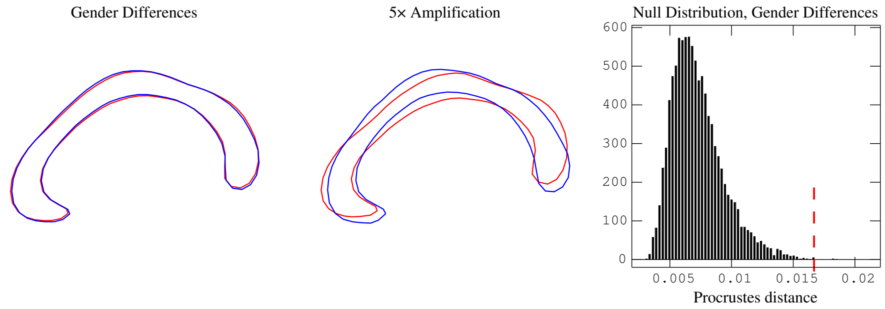


Figure 7.3: The mean female CC shape (red) versus the male (blue). Also shown is the empirical null distribution function of the Procrustes distance between the two shapes. The observed distance is represented by the red dashed vertical line, corresponding to $p = 0.0015$.

the deformations were set to 5, 20 and 50 non-zero components, corresponding to 3%, 13% and 32% of the total number of components ($2 \cdot 78 = 156$). This choice of scales provides a relatively large span of deformations, while interpretability is maintained. A standard PCA was also applied, obviously corresponding to 100% non-zero components. Figure 7.4 shows the resulting deformations. Note the coherence of the sparse deformation patterns. This property is in no way enforced by the algorithm and neither are such assumptions desired from a fully exploratory method. Instead, the coherence is a result of the high correlations between adjacent landmarks. In theory there is, however, nothing to keep the deformations from breaking up into an arbitrary number of separate effects, and this is seen to occur to some extent for SPCA(20) and SPCA(50).

The deformations for each SPCA scale and for PCA were related to four clinical outcome variables using the univariate regression scheme outlined in Section 7.2.3. The variables are gender (male/female), age (years), walking speed (meters/second) and verbal fluency (words/minute). In the tests for gender and age, no confounding variables were identified. For walking speed and verbal fluency, the results were adjusted for age, gender, level of education and the logarithm of the volume of white matter hyperintensities, as suggested by previous studies on the same data set [70, 126].

The results for each clinical variable are given in Figure 7.5. As described in Section 7.2.4, the deformations shown for each scale and variable are the compounded results for each deformation mode corresponding to an adjusted p-value below $\alpha = 0.05$. To provide more specific results in the case of gender differences, Table 7.1 lists the resulting coefficient values for each deformation mode and scale with corresponding significance levels.

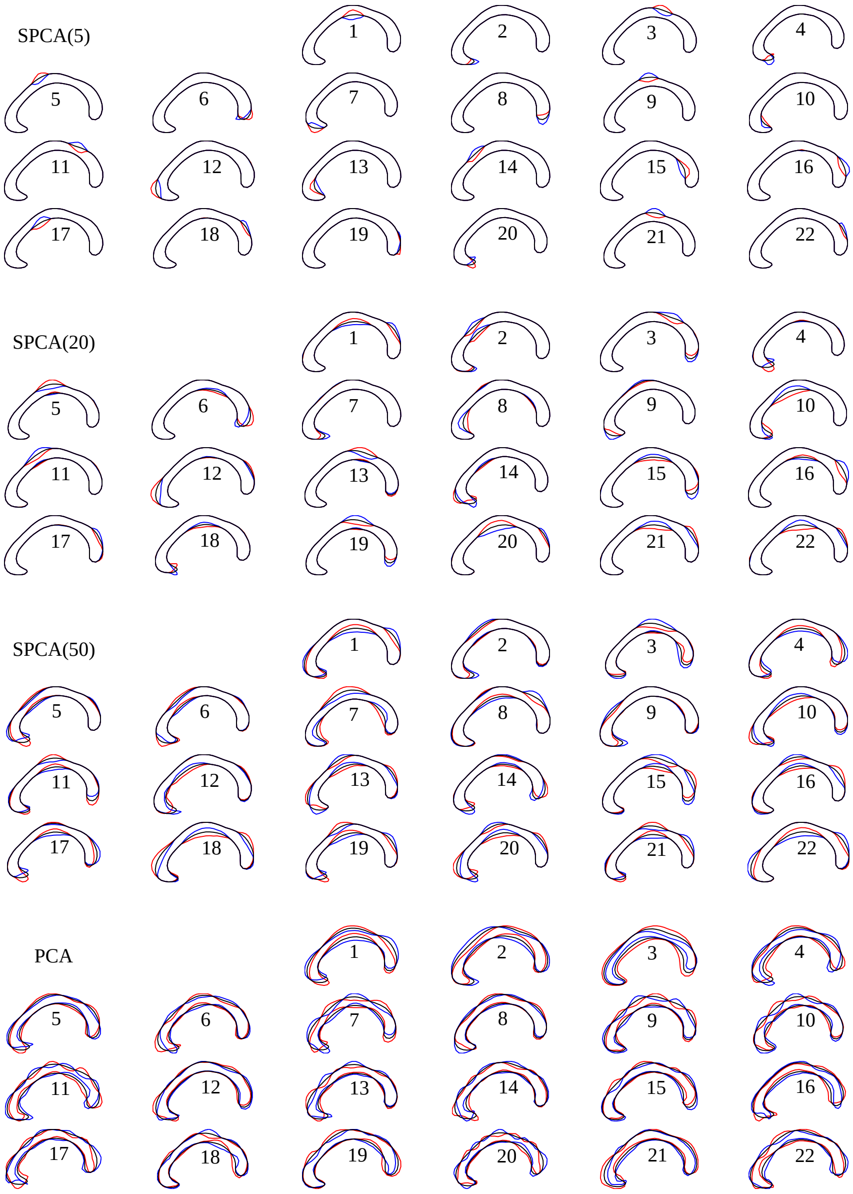


Figure 7.4: Example deformation modes. Each group of deformations represents one scale. The notation SPCA(k) denote a sparse decomposition with k nonzero components. The mean shape is shown in black, while blue and red lines represent deformations in the positive and negative direction respectively. The deformations have been appropriately scaled for visualization.

	SPCA(5)	SPCA(20)	SPCA(50)	PCA
1	−0.0034	−0.0075	−0.0092	−0.0131
2	0.0043	0.0074 **	0.0100 **	0.0077
3	−0.0017	0.0043	0.0067	−0.0007
4	0.0025	−0.0036	−0.0032	−0.0041
5	−0.0038	−0.0044	−0.0099 *	0.0018
6	−0.0032	−0.0022	−0.0086 **	0.0027
7	−0.0030	0.0058 *	−0.0083 **	−0.0020
8	0.0040	−0.0039 *	−0.0081 *	−0.0012
9	0.0024	0.0048 *	0.0068 *	0.0001
10	0.0010	0.0056	0.0105 **	0.0004
11	0.0009	0.0061 *	−0.0089 *	−0.0010
12	0.0032	0.0061	0.0077 *	0.0008
13	0.0028	−0.0034	0.0107 *	−0.0009
14	0.0041 *	0.0069 *	0.0016	−0.0000
15	0.0011	0.0070	0.0080	0.0007
16	−0.0011	−0.0059	0.0096 *	−0.0005
17	0.0041	−0.0064 *	−0.0053	−0.0005
18	−0.0048	0.0041	0.0085	−0.0007
19	−0.0042 *	0.0053	−0.0110 *	0.0004
20	−0.0020	−0.0074	0.0099 *	0.0009
21	0.0017	0.0069	−0.0085	−0.0006
22	−0.0047 *	0.0071	−0.0092	0.0004

Table 7.1: Regression coefficients β_i (cf. Equation 7.11) from the investigation of CC gender differences. Significance levels are indicated by * ($p < 0.05$), ** ($p < 0.01$), and *** ($p < 0.001$), corrected for multiple comparisons using permutation testing. Row numbers refer to the deformation modes shown i Figure 7.4.

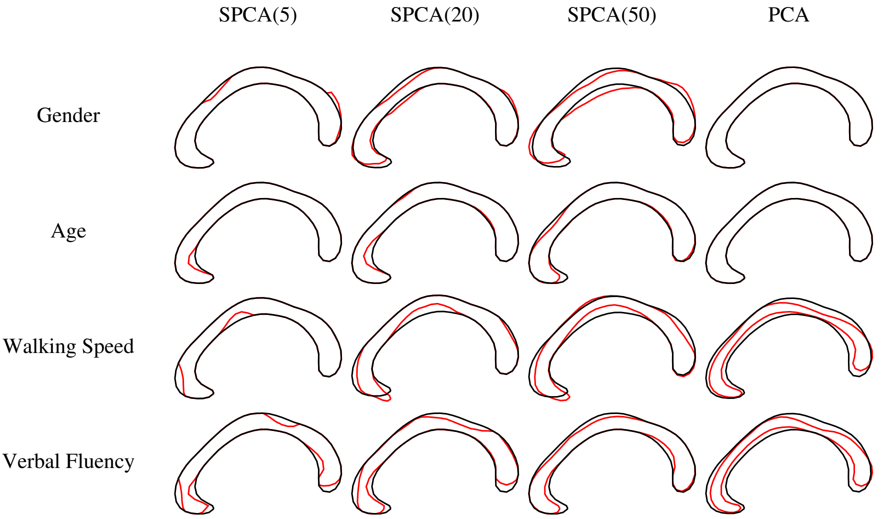


Figure 7.5: Results for each clinical outcome variable and scale of decomposition. The mean shape is drawn using black lines, while red lines represent a more female CC, old age, and lower scores for walking speed and verbal fluency. The results for verbal fluency have not been corrected for multiple comparisons. The deformations show a high degree of consistency over different scales and are sufficiently coherent and regular for clinical interpretation.

To put the data-driven SPCA method into perspective, tests for each clinical outcome variable were also investigated through a direct analysis of the original variables and by using the model-based wavelet approach. Figure 7.6 shows the results from these tests.

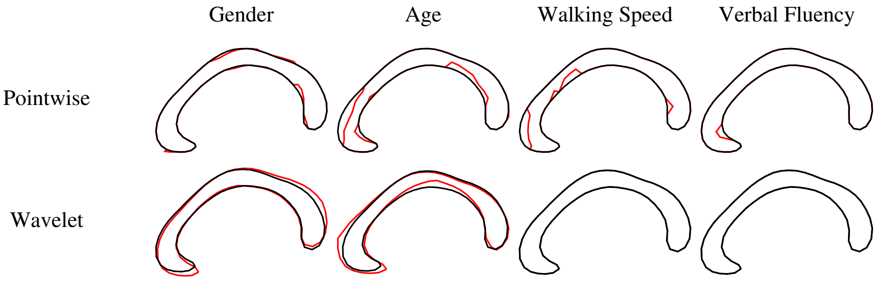


Figure 7.6: Results for all four clinical outcome variables using the direct component-wise approach (top row) and the wavelet coefficient approach (bottom row), showing the mean shape (black) versus a more female shape (red). The methods seem inferior to the proposed method in terms of statistical power, specificity, and interpretability.

7.4 Discussion

This paper has introduced a method for relating localized, anatomically meaningful patterns of variation to clinical outcome using a method for the estimation of sparse principal components.

7.4.1 Method

The results presented in Figure 7.4 suggests that the SPCA method is a useful method for deriving localized and interpretable patterns of variability. The computational complexity is reasonable in the present case of relatively many observations, but limited dimensionality. Computation times varied from seconds for low scale deformations, to minutes for more complex cases. Convergence seems to vary considerably as well, with almost immediate convergence in some cases, and slower and more irregular convergence in others. Alternative or approximate optimization schemes for the SPCA criterion in (7.7) should be a focus of future work. For application to higher dimensional data, we supply a discussion below.

Splitting the testing procedure performed to relate spatial deformations to clinical outcome data into a series of univariate tests comes with both benefits and drawbacks. Most importantly, it provides a strong form of regularization. Each model contains a low number of variables (one plus any covariates), making the analysis more stable in cases with few observations. The main disadvantage is that this analysis disregards the correlation structure between variables. However, PCA scores are uncorrelated and are therefore unaffected by this property. SPCA scores generally show stronger patterns of correlation and the SPCA analysis may be more notably influenced by this limitation. Estimation methods that take the correlation structure between spatial variables into consideration is another topic for further investigation.

Two alternative methods for a localized analysis of anatomy was outlined. Arguably, the results obtained using these methods (cf. Figure 7.6) were inferior to those of the proposed method. The point based method suffers from two apparent disadvantages. The high number of degrees of freedom makes the method prone to overfitting. Disparate results may be obtained for adjacent points, leading to variational patterns that are scattered or irregular, and therefore difficult to interpret. The SPCA method circumvents this problem by making sure that each variable represents an anatomically meaningful pattern over several data points. The second problem is the high number of variables. Procedures for adjustment for multiple comparisons such as Bonferroni correction or the permutation method outlined in Section 7.2.3 tend to adjust more for more high-dimensional models, effectively resulting in lower levels of significance. The discouraging results obtained using the wavelet representation seems to be due to the spatial appearance of the derived variables, which look implausible from an anatomical viewpoint (cf. Figure 7.1). The poor results may therefore be due to an improper choice of mother wavelet. The first order coiflet was used here, because of its low complexity and high degree of symmetry. Reissell present a type of wavelet called pseudocoiflet [118], which are custom designed for curve and surface representation, and may be a more suitable choice. Further, the wavelet representation also suffers from multiple testing problems, as the number of variables involved is equal to the number of variables in the original data. To alleviate this, the wavelet representation can either be truncated, or separate analyses can be performed at each wavelet scale. Preliminary tests using the latter approach did not point to an improvement in the results.

There exists a few interesting alternatives to SPCA to construct sparse representations of anatomy, most notably independent component analysis (ICA) [161] and varimax rotated principal components [139]. Some experiments using these bases have been carried out, with results similar to those of SPCA. One disadvantage shared by both ICA and factor rotation is that the patterns produced are only approximately sparse. The residual variation makes the results more difficult to interpret.

Extension to 3D and Higher Dimensions

The corpus callosum outlines used here to validate the method are represented by planar shape data. However, the outline of the method, from the extraction of spatially sparse and meaningful features to the subsequent analysis of the relation of these to clinical data, is applicable to data of any dimension, modality and topology, given that its distribution is suitable for linear modeling. With an increasing number of variables, such as for shape data in three dimensions, comes an increase in computational burden and memory requirements. The core problem for most SPCA algorithms is the need to calculate and store the $p \times p$ covariance matrix of the variables involved. The algorithm presented here uses sequential up- and downdating of the Cholesky factorization of the covariance matrix [54], such that only currently active variables are being considered. With k active variables, this limits the storage requirement to a $k \times k$ matrix. The complexity of the algorithm is therefore more due to the number of non-zero components, than to the total number variables involved.

In cases where a very large number of variables must be considered, such as for complex shape representations in three dimensions or for functional MRI analyses, the optimization problem in (7.7) becomes too complex and the alternating estimation algorithm will not converge. It turns out that the criterion in (7.7) is valid for any positive value of λ and that the solutions are not particularly dependent on the choice of this parameter [134, 177]. Specifically, a computationally efficient algorithm emerges for $\lambda = +\infty$. In this case, the complex elastic net process to estimate $\mathbf{B}$ can be replaced by a simpler soft-thresholding rule,

$$\mathbf{b}_j = \left(|\mathbf{a}_j^T \mathbf{X}^T \mathbf{X}| - \frac{\delta_j}{2} \right)_+ Sign(\mathbf{a}_j^T \mathbf{X}^T \mathbf{X}). \tag{7.13}$$

where $(\cdot)_+ = \max(0, \cdot)$ and $\mathbf{a}_j$ is the jth column of $\mathbf{A}$. Note that the $p \times p$ matrix $\mathbf{X}^T\mathbf{X}$ does not need to be explicitly calculated and stored if the matrix operations are properly ordered. Some preliminary results on using this method for exploratory analyses of fMRI data can be found in [133].

SPCA and its related methods for regression are available as add-on packages for the statistical environment R. Similar implementations for the Matlab platform are available from the web page of the first author, `www.imm.dtu.dk/~kas/software/spca`.

7.4.2 Clinical Application

We will now comment on the results for the application of the method on the corpus callosum data. These comments are provided to support the method only, a more thorough clinical investigation with subsequent interpretation is deferred to a separate paper.

The sexual dimorphism of the CC is a closely investigated subject that has yielded disparate results. However, several authors [2, 20, 33, 39] report on a more bulbous splenium for females. The present results clearly agree with this finding. The results can also be seen to agree with the male/female mean shape differences depicted in Figure 7.3. The advantage of using the proposed method is the additional information on localization. In a number of limited regions along the boundary, the method quantifies the strength of the relevant discrepancies, giving more detailed anatomical information. Moreover, any global method such as measures of callosal area or the Procrustes distance measure used in this paper may not prove to be significant if the differences are small and highly localized. Using sparse decomposition, such differences can be identified and quantified correctly.

The deformation of the CC corresponding to the measure of walking speed provides an example that nicely demonstrates the potential of the method. In the third row of Figure 7.5, some thinning can be seen in the genu area, but more interestingly, a clear deformation is also present in the rostral body, corresponding well to the area of the CC containing fibers related to the motor cortex (cf. Figure 7.2). All SPCA scales show this effect to some extent.

The results for verbal fluency did not reach significant levels when corrected for multiple comparisons. In Figure 7.5, the corresponding unadjusted deformations for $p < 0.05$ are shown. Although not highly significant, the results again make anatomical sense. On scales SPCA(5) and SPCA(20), a thinning of the isthmus subregion occurs. Referring to Figure 7.2, this seems to correspond to atrophy of fiber tissue connecting to brain regions involved in auditory tasks. This result is also in accordance with previous results based on the same data set [70], where verbal fluency was found to correlate exclusively with the rostrum and isthmus regions. The latter paper used measures of callosal area based on a partitioning of the CC into subregions, and declared significance at level $\alpha = 0.01$, not corrected for multiple comparisons.

The deformation modes extracted using PCA did not provide much interpretational value in this application. For gender and age, no deformations correlated significantly with the outcome. For walking speed and verbal fluency, PCA yielded some significant results, but the limited interpretational power becomes apparent in the results. Effects are present throughout the entire boundary, and inference of structure-function relationships become difficult.

7.5 Conclusions

Sparse principal component analysis is introduced as an attractive method for extracting strictly sparse and anatomically meaningful variables from a data set. While the results may be interesting for direct analysis, this paper shows how to relate these spatial variables to clinical outcome data, making it possible to derive typical deformation patterns related to e.g. pathology. As an illustrative example, results are presented based on a large data set of corpus callosum outlines for several clinical target variables, demonstrating the capabilities of the method. The method has been compared to both a simple point-based alternative, as well as decomposition using a wavelet transform. The results suggest that these methods are either less precise, or offer inferior interpretability compared to the sparse principal component analysis approach.

Acknowledgments

This study was supported by the European Union (grant QLRT-2000-00446, Impact of age-related brain white matter changes on transition to disability in the elderly – ”Leukoaraiosis And DISability”). We would also like to acknowledge the financial support from the Technical University of Denmark, the Danish Velux Foundation, and the Danish Alzheimer Research Foundation. The authors would like to thank the group of anonymous reviewers and associate professor Bjarne Ersbøll, Technical University of Denmark, for inspirational comments.

Chapter 8

Sparse Statistical Deformation Model for the Analysis of Craniofacial Malformations in the Crouzon Mouse

Hildur Ólafsdóttir, Michael Sass Hansen, Karl Sjöstrand, Tron A. Darvann, Nuno V. Hermann, Estanislao Oubel, Bjarne K. Ersbøll, Rasmus Larsen, Alejandro F. Frangi, Per Larsen, Chad A. Perlyn, Gillian M. Morriss-Kay and Sven Kreiborg.

Abstract

Crouzon syndrome is characterised by the premature fusion of cranial sutures. Recently the first genetic Crouzon mouse model was generated. In this study, Micro CT skull scannings of wild-type mice and Crouzon mice were investigated. Using nonrigid registration, a wild-type craniofacial mouse atlas was built. The atlas was registered to all mice providing parameters controlling the deformations for each subject. Our previous PCA-based statistical deformation model on these parameters revealed only one discriminating mode of variation. Aiming at distributing the discriminating variation over more modes we built a different model using Independent Component Analysis (ICA). Here, we focus on a third method, sparse PCA (SPCA), which aims at approximating the properties of a standard PCA while introducing sparse modes of variation. The results show that SPCA outperforms both ICA and PCA with respect to the Fisher discriminant, although many similarities are found with respect to ICA.

8.1 Introduction

Crouzon syndrome was first described nearly a century ago when calvarial deformities, facial anomalies, and abnormal protrusion of the eyeball were reported in a mother and her son [30]. Later, the condition was characterised as a constellation of premature fusion of the cranial sutures (craniosynostosis), orbital deformity, maxillary hypoplasia, beaked nose, crowding of teeth, and high arched or cleft palate. Identification of heterozygous mutations in the gene encoding *fibroblast growth factor receptor type 2* (*FGFR2*) have been found responsible for Crouzon syndrome [117]. Recently a mouse model was created to study one of these mutations ($FGFR2^{Cys342Tyr}$)[41]. Incorporating advanced small animal imaging techniques such as Micro CT, allows for detailed examination of the craniofacial growth disturbances. Studying the craniofacial shape differences in detail contributes to the understanding of the syndrome, surgery planning and diagnosis in humans. A recent study, performing linear measurements on Micro CT scans, proved the mouse model applicable to reflect the craniofacial deviations occurring in humans with Crouzon syndrome [113]. Previously, we have extended this study to assess the local deformations between the groups by constructing a deformable shape and intensity-based atlas of wild-type (normal) mouse skulls. Deforming this atlas to all mice, the craniofacial shape differences can be analyzed [108].

To analyse and interpret these deformations in a meaningful way, it is desirable to reduce the large number of dimensions and at the same time localise the growth deviations with respect to the atlas. This leads us to statistical deformation models (SDMs). These are closely related to statistical shape models but the fact that the whole correspondence field is modelled makes them more powerful. A standard PCA has been a popular approach to build SDMs (e.g. [93, 100, 125]) but recently different techniques have been applied, e.g. wavelet-based PCA [168].

With respect to the mouse study, PCA was previously performed [107]. This analysis revealed only one discriminating mode of variation, mainly reflecting global differences between the groups. This kind of variation can be hard to interpret and in a recent study, we showed that applying Independent Component Analysis (ICA) to the deformation fields resulted in several discriminating modes, revealing the local differences between the groups. Sparse Principal Components Analysis (SPCA) [177] has proven successful when applied in shape modelling [134]. In this paper we introduce the use of SPCA to build a Sparse Statistical Deformation Model and provide a comparison to a standard PCA and ICA with focus on the discriminative ability. We believe this is the first time SPCA is applied to statistically model deformation fields.

8.2 Data Material

Production of the $Fgfr2^{C342Y/+}$ and $Fgfr2^{C342Y/C342Y}$ mutant mouse (Crouzon mouse) has been previously described [41]. All procedures were carried out in agreement with the United Kingdom Animals (Scientific Procedures) Act, guidelines of the Home Office, and regulations of the University of Oxford.

For three-dimensional (3D) CT scanning, 10 wild-type and 10 $Fgfr2^{C342Y/+}$ specimens at six weeks of age (42 days) were sacrificed using Schedule I methods and fixed in 95% ethanol. They were sealed in conical tubes and shipped to the Micro CT imaging facility at the University of Utah. Images of the skull were obtained at approximately 46μm × 46μm × 46μm resolution using a General Electric Medical Systems EVS-RS9 Micro CT scanner. Fig. 8.1 shows an example of the living mice and the imaging data appearance.

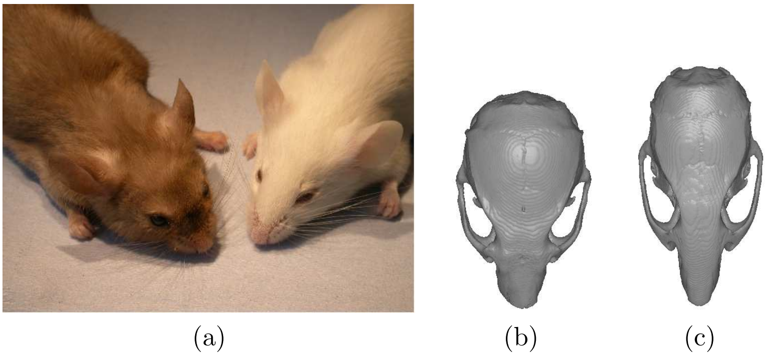


Figure 8.1: (a) Photo of a Crouzon mouse (left) and a wild-type mouse (right). Skulls Extracted from CT images of (b) a Crouzon mouse, (c) wild-type mouse.

8.3 Methods

The steps taken to automatically assess the local shape deviations between groups, statistically, from the Micro CT images are the following.

1. Build a craniofacial wild-type mouse atlas from the Micro CT's using nonrigid image registration

2. Match atlas to all 20 cases (wild-type and Crouzon mice) using nonrigid image registration

3. Use the resulting deformation parameters as input to a SPCA

8.3.1 Atlas Building and Registration

The first two steps of the procedure were presented in [108]. The nonrigid registration algorithm based on B-splines [124, 127] was applied. This algorithm uses a transformation model which is a combination of a global and a local transformation model, $\mathbf{T}(\mathbf{x}) = \mathbf{T}_{\text{global}}(\mathbf{x}) + \mathbf{T}_{\text{local}}(\mathbf{x})$. The global transformation model consists in our case of a rigid transformation matrix (with 6 degrees of freedom). The local transformation model describing the nonrigid part of the model is written by the tensor product of the 1D cubic B-splines,

$$\mathbf{T}_{local}(x, y, z) = \sum_{l=0}^{3} \sum_{m=0}^{3} \sum_{n=0}^{3} B_l(u) B_m(v) B_n(w) \mathbf{c}_{i+l,j+m,k+n} \tag{8.1}$$

where $\mathbf{c}$ are the parameters of the B-splines ordered in a $p_x \times p_y \times p_y$ lattice. u, v and w are the (x, y, z) image coordinates translated into the lattice coordinates.

8.3.2 A Sparse Statistical Deformation Model

The third step of the procedure listed above is the main focus of this paper. The control points (parameters) of the B-splines in Equation 8.1 provide a compact representation of the correspondence fields. As shown in [125] it is sufficient to perform a statistical analysis on these control points to obtain a compact description of the deformations. Using a common reference frame, e.g. an atlas, as the origin of the registrations, the control points for a subject reflect its local deviation from this reference frame. Concatenating the 3D control points for subject i into a row vector $\mathbf{C}_i = [c_1, ..., c_p]$, where $p = 3p_x p_y p_z$, gives the ith row of the $n \times p$ data matrix to analyse (n is the number of observations).

SPCA approximates the properties of a standard PCA while introducing sparsity in the modes of variation. Zou et al. [177] take advantage of formulating PCA as a regression problem leading to the *SPCA criterion*

$$\begin{gathered}(\hat{\mathbf{A}}, \hat{\mathbf{B}}) = argmin_{\mathbf{A},\mathbf{B}} \textstyle\sum_{i=1}^{n} ||\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i||^2 + \lambda \sum_{j=1}^{k} ||\mathbf{b}_j||^2 + \sum_{j=1}^{k} \delta_j ||\mathbf{b}_j||_1 \\ s.t.\ \mathbf{A}^T\mathbf{A} = \mathbf{I}\end{gathered} \tag{8.2}$$

Here $\mathbf{x}_i$ denotes the ith column of $\mathbf{X}^T$. This formulation assumes k modes to be retained in the model. The columns of $\mathbf{B}$ represent the principal axes (loading vectors $\mathbf{b}_j$, $j = 1, ..., k$) and $\mathbf{B}$ projects observation i onto those axes. The matrix $\mathbf{A}$ takes the observation back to the original space. Hence, the first term measures the reconstruction error of the model. The second term, the L2 penalty is included to ensure a unique solution, also in cases where $p > n$, and the third term, L1 penalty, introduces sparsity. These two latter terms are adopted from Elastic Net regression [175]. The constraint weight, λ, must be chosen beforehand, and has the same value for all PCs, while δ may be set to different values for each PC, providing good flexibility.

The problem in Equation 8.2 is usually solved iteratively by fixing $\mathbf{A}$ in each iteration, solving for $\mathbf{B}$ using the LARS-EN algorithm [175] and recalculating $\mathbf{A}$. However, when we have $p \gg n$ as in our case, Zou et al. [177] have shown that by letting $\lambda \rightarrow \infty$, $\mathbf{B}$ can be determined by soft thresholding[1]

$$\mathbf{b}_j = (|\mathbf{a}_j^T \mathbf{X}^T \mathbf{X}| - \frac{\delta_j}{2})_+ \cdot \operatorname{sign}(\mathbf{a}_j^T \mathbf{X}^T \mathbf{X}), \qquad j = 1, 2, ..., k \tag{8.3}$$

where k is the number of modes and $\mathbf{a}_j$ is the jth column of $\mathbf{A}$. This approach was taken here enforcing the same fixed level of sparsity in each loading vector by dynamically changing (δ_j) in each iteration. To maximise the total adjusted variance [177] explained by the SPCA, the modes were ordered allowing for perturbations as suggested in [134].

Since the aim of our sparse deformation model is to discriminate between the two groups of mice the final ordering of modes was defined with respect to the Fisher discriminant. That is, the observations were projected onto the principal directions, the Fisher discriminant between the groups calculated for each mode and the principal directions ordered with respect to decreasing Fisher discriminant score. In general, for class 1 and 2, the Fisher discriminant is defined as

$$F = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}, \tag{8.4}$$

where μ_i is the mean of class i and σ_i^2 is the variance of class i.

8.4 Experimental Results

The accuracy of the image registration algorithm (registering the atlas to each of the 20 cases) is essential for the deformation model to be valid. In [108],

[1] $(z)_+$ denotes that if $z < 0$, z is set to 0 and if $z >= 0$, z is kept unchanged. The term is denoted hinge-loss.

the manual annotations from two observers were used to assess the registration accuracy. Using the optimal transformations from the image registrations, landmarks were obtained automatically. The landmark positions were statistically compared to those annotated by the human observers. This showed that the automatic method provided as good accuracy as the human observers and, moreover, it was more precise, judged from the significantly lower standard deviation.

The SPCA was applied to the matrix of control points ($p = 21675$). A threshold of 2000 points was used to obtain equal sparsity in each mode of variation. Fig. 8.2 (a-c) shows the observations projected onto the first six sparse principal directions (ordered by Fisher discriminant score). To evaluate the ability of the sparse SDM to assess the local group differences, it was compared to a standard PCA and our previous approach [56] using ICA [67]. Fig. 8.2(d-i) shows scatter plots of the first six modes for ICA and PCA, sorted with respect to the Fisher discriminant.

The score plots already give an idea about the discrimination ability of the different approaches. To give a more quantitative measure, the Fisher discriminant was assessed in a leave-one-out fashion for all three approaches. This is plotted with error bars for each of the approaches in Fig. 8.3.

With emphasis on the group differences, each mode of the sparse model was visualised by selecting the extremes from each group in model space (Fig. 8.2) and project back into the space of control points. This set of control points generated from the model was then applied to the atlas to obtain the deformed volumes of the two extremes. Subsequently the surfaces were extracted for visualisation. Fig. 8.4 shows mode 1,3,4 and 6. Mode 2 was excluded from this visualisation due to an overlap in variation with mode 1.

Deforming the atlas along the discriminating modes of the ICA model reveals many similarities between ICA and SPCA. To give an example Fig. 8.5 shows IC 5 which is closely related to SPC 4.

8.5 Discussion and Conclusions

The score plots in Figure 8.2 indicate that both SPCA and ICA are capable of discriminating between the two groups in up to six deformation modes. The standard PCA only discriminates between the groups in the first mode. Figure 8.3 confirms these speculations. It is evident that PCA is only capable of discriminating between the groups in one mode of variation. SPCA performs slightly better than the ICA, but the ICA seems to be more robust judged from

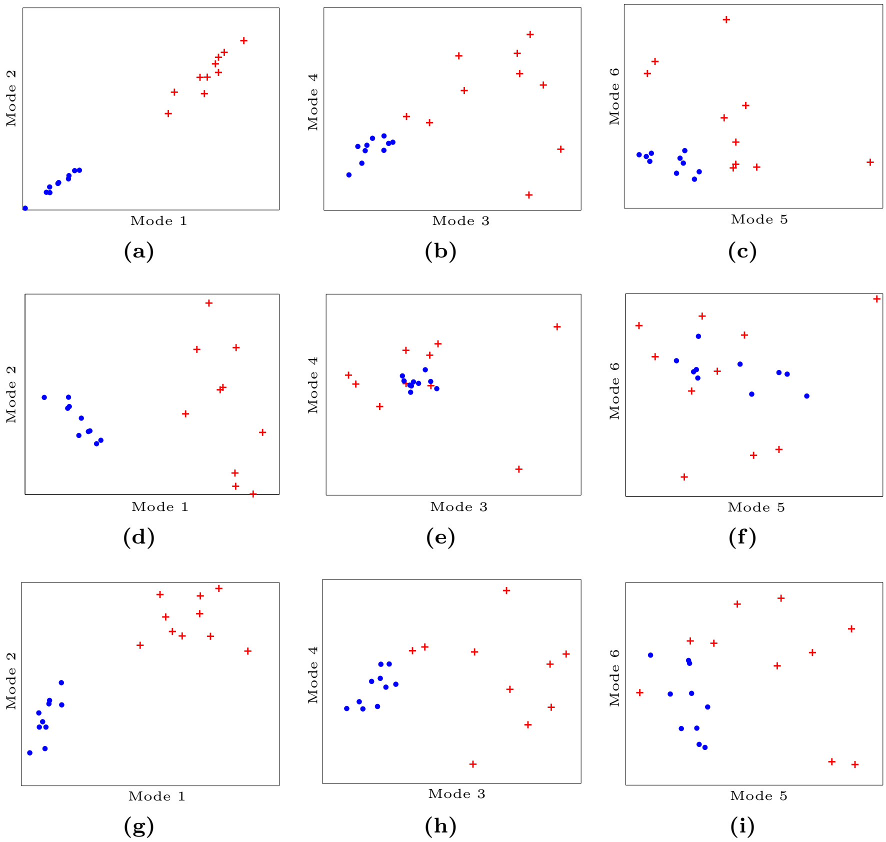


Figure 8.2: Projection of observations into the space of the first six components (ordered by Fisher discriminant) using (a-c) SPCA, (d-f) PCA and (g-i) ICA. Crosses denote Crouzon cases while circles denote wild-type cases. (a,d,g) Mode 2 vs. mode 1; (b,e,h) Mode 4 vs. mode 3; (c,f,i) Mode 6 vs. mode 5.

the error bars. Considering the low number of points in the sparse model, this is understandable.

Visualising the sparse deformation modes in Figure 8.4 indicates that compared to wild-type mice, the skulls of Crouzon mice are higher and longer (SPC 1), are asymmetric with respect to zygoma and nose (SPC 3), have different shape of the middle ear and back of the head (SPC 4), and have an angulated cranial base (SPC 6). These observations correspond up to some degree with what has previously been seen in humans using manual measurements (see e.g. [79]). The asymmetric behaviour seen in SPC 3 can be explained by the full or partial fusion of cranial sutures at different sides and different times. The different

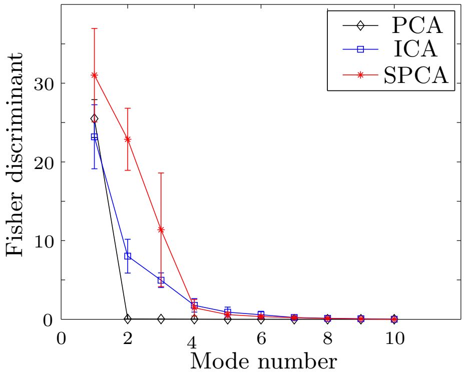


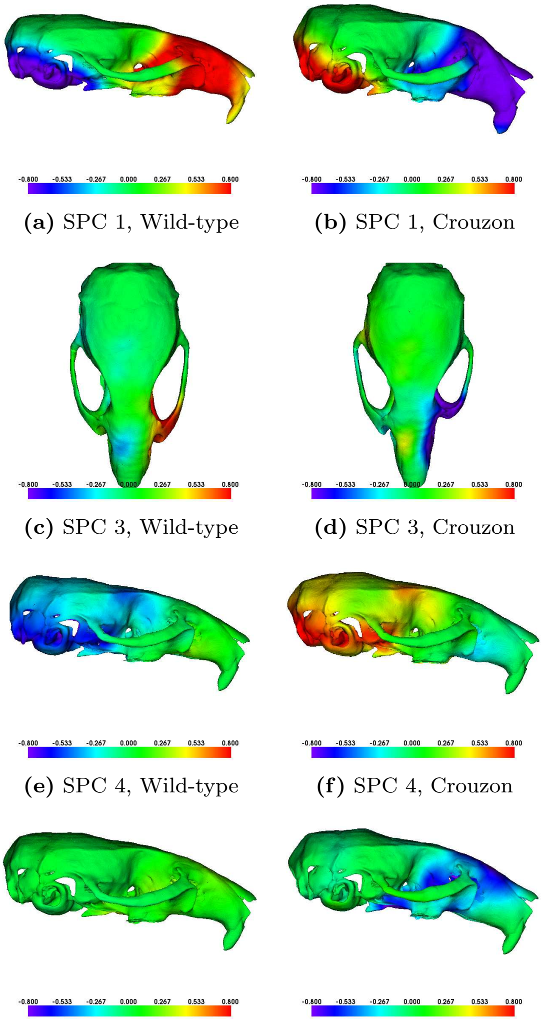
Figure 8.3: The Fisher discriminant plotted vs. deformation mode number for PCA, ICA and SPCA. The values are obtained in a leave-one-out experiment providing the error bars (one standard deviation).

shape of the middle ear and the increased angulation of the cranial base has not been reported in humans to our knowledge and may therefore be an important contribution to the understanding of the growth disturbances. The angulation was found in mice both using ICA [56] and PCA (with global transformation model extended to 9 DOFs) [107]. The difference in shape of the middle ear and back of the head was also captured by the ICA approach as seen in Figure 8.5. In fact SPC 4 and IC 5 are extremely similar, but SPCA seems to create slightly stronger evidence for the group difference. In general, the ICA modes introduce more noise than sparse PCA, since many elements are close to 0, while in SPCA, the sparsity property avoids this. Another advantage of SPCA is that it is solely based on second order statistics making it less committed than ICA, which uses higher order statistics.

In conclusion, with respect to discriminative ability, SPCA and ICA give similar results when applied to model deformations. Both of the approaches outperform a standard PCA. However, due to the simplicity and flexibility of SPCA, it should be the preferred method for this type of analysis.

Acknowledgements

For all image registrations, the Image Registration Toolkit was used under Licence from Ixico Ltd.



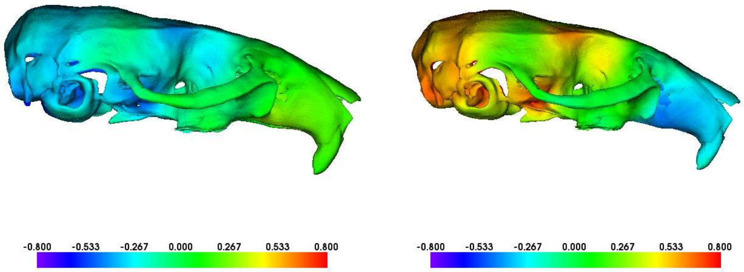
(a) SPC 1, Wild-type **(b)** SPC 1, Crouzon

(c) SPC 3, Wild-type **(d)** SPC 3, Crouzon

(e) SPC 4, Wild-type **(f)** SPC 4, Crouzon

(g) SPC 6, Wild-type **(h)** SPC 6, Crouzon

Figure 8.4: Sparse Principal Deformation modes 1,3,4 and 6, visualised on surfaces after deforming atlas to the extremes of each mode. The colors are intended to enhance the regions where changes have occurred in the deformed surfaces. The colors denote displacement with respect to atlas (in mm), with positive values (red) pointing outwards.



(a) IC 5, Wild-type **(b)** IC 5, Crouzon

Figure 8.5: Independent Deformation mode 5 visualised on surfaces after deforming atlas to the extremes of the mode. The colors are intended to enhance the regions where changes have occurred in the deformed surfaces. The colors denote displacement with respect to atlas (in mm), with positive values (red) pointing outwards.

Chapter 9

A Path Algorithm for the Support Vector Domain Description and its Application to Medical Imaging

Karl Sjöstrand, Michael Sass Hansen, Henrik B. Larsson and Rasmus Larsen

Abstract

The support vector domain description is a one-class classification method that estimates the distributional support of a data set. A flexible closed boundary function is used to separate trustworthy data on the inside from outliers on the outside. A single regularization parameter determines the shape of the boundary and the proportion of observations that are regarded as outliers. Picking an appropriate amount of regularization is crucial in most applications but is, for computational reasons, commonly limited to a small collection of parameter values. This paper presents an algorithm where the solutions for all possible values of the regularization parameter are computed at roughly the same computational complexity previously required to obtain a single solution. Such a collection of solutions is known as a regularization path. Knowledge of the entire regularization path not only aids model selection, but may also provide new information about a data set. We illustrate this potential of the method in two applications; one where we establish a sensible ordering among a set of corpora callosa outlines, and one where ischemic segments of the myocardium are detected in patients with acute myocardial infarction.

9.1 Introduction

The support vector domain description (SVDD) [151, 152] is a method for one-class classification where the aim is to obtain an accurate estimate of the support of a set of observations. Such methods differ from two or multi-class classification in that we are typically interested in a single object type and want to distinguish this from "everything else", rather than separating one class from other known classes. There are several benefits and uses for such a method. It is a natural choice for outlier and novelty detection for two reasons. First, outlier data is typically sparse and difficult to obtain, while normal data is readily available. Second, the nature of outlier data may not be known. Even a standard two-class classification task may be better suited for a one-class method when one class is sampled very well and the other is not. The SVDD is a non-parametric method in the sense that it does not assume any particular form of the distribution of the data. The support of the unknown distribution of the data points is modeled by a boundary function enclosing the data. This boundary is "soft" in the sense that atypical points are allowed outside the boundary. The proportion of exterior points is governed by a single regularization parameter λ, which must be tuned for each data set and application. This paper presents an algorithm, in which the SVDD solutions for *all possible* values of λ are calculated with roughly the same computational complexity required by standard algorithms to estimate a single solution. Such a complete set of solutions is sometimes referred to as a *regularization path*. Proper choice of λ, which previously depended on either ad-hoc rules or probing the regularization path at a sparse set of locations, is now greatly facilitated since a search through the entire solution set becomes possible. Further, the regularization path itself provides valuable information that hitherto has been impractical to obtain. Two such examples are given in this paper.

The SVDD was presented by Tax and Duin [151] and again in [152] with extensions and a more thorough treatment. The boundary function is modeled by a hypersphere, a geometry which can be made less constrained by mapping the data points to a high-dimensional space where the classification is performed. This leads to a methodology known as the *kernel trick* in the machine learning community [165]. Schölkopf et al. [128] presents a conceptually different approach to one-class classification where a hyperplane is used to separate the data points from the origin. The solutions are, however, shown to be equivalent to those of the SVDD when radial basis expansions are used. The Gaussian kernel is one such function and represents the most frequent choice in the literature.

The SVDD has found uses both in a wide range of applications and as a basis for new methodology in statistics and machine learning. Banerjee et al. [6] used the SVDD for anomaly detection in hyperspectral remote sensing imagery. Compared to standard parametric approaches, the SVDD was found to improve both accuracy and computational complexity. Lee et al. [90] suggest improving the basic SVDD by weighting each data point by an estimate of its corresponding density. The density is approximated either by a K-nearest-neighbor or a Parzen window approach. This modification is shown to improve the basic SVDD in studies of e.g. breast cancer, leukemia and hepatitis. Other applications include pump failure detection [153], face recognition [91, 130], speaker recognition [37] and image retrieval [80].

The ability of the SVDD to focus modelling of the density of a set of observations to its support makes it a natural alternative to large-margin classifiers such as the support vector machine (SVM) [165]. Lee and Lee [88] present a method for multi-class classification built on the SVDD. First, a separate boundary is estimated for each class. Second, a classifier is built using Bayes optimal decision theory where the class-conditional densities are approximated from the respective SVDD representations. The resulting classifier demonstrates similar or better performance compared to several competing classification techniques. Similar approaches have been proposed by Choi et al. [19] and Ban and Abe [5].

The kernel formulation of the SVDD may lead to boundaries that split up into two or more separate closed hypersurfaces. These were interpreted as cluster boundaries by Ben-Hur et al. [8] who developed an algorithm for the assignment of cluster labels called support vector clustering. The results are dependent on the parameters of the chosen kernel and the amount of regularization in the SVDD, pointing to the usefulness of the results presented in this paper. For instance, support vector clustering has been applied to exploratory analysis of fMRI data [159].

The path algorithm presented in this paper is one example of several recent investigations into the efficient estimation of regularized statistical methods where the coefficients are piecewise-linear functions of the regularization parameter. The increasing interest in regularization paths is in part motivated by a seminal paper by Efron et al. [40], where a novel method for penalized regression called least angle regression (LAR) is presented. It is shown that the LAR coefficient paths are piecewise-linear with respect to the regularization parameter and that these paths can be calculated at the computational cost of a single ordinary least squares estimation. Through small modifications to the algorithm, the regularization path of the least absolute shrinkage and selection operator (LASSO) [157] and a variant of forward selection can be obtained, circumventing the need for costly computational techniques such as linear and quadratic programming. Inspired by this finding, similar algorithms have been developed for other statistical methods such as generalized linear models [112] and support vector machines [60, 173]. Zou and Hastie [175] developed a new method for regression called the elastic net and suggested a path algorithm for its computation. Rosset and Zhu [122] discuss necessary and sufficient conditions for the existence of piecewise-linear regularization paths and supply several examples. The work by Hastie et al. [60] on the entire regularization path for the support vector machine was the inspiration for this paper, and we acknowledge the numerous similarities between their work and the description and derivation of the SVDD path algorithm presented here.

9.2 Methods

In this section, we will give a concise explanation of the support vector domain description and the standard algorithm for its computation. This is followed by a description of the proposed path algorithm. The section is concluded by a detailed discussion on the implementation of the method. Figure captions in this and later sections refer to the color illustrations available in the electronic version of this paper.

The support vector domain description models the distributional support of a data set using a hypersphere. Observations enclosed by this boundary function are considered trustworthy data while points outside the boundary are treated as outliers. The hypersphere is specified by its center $\mathbf{a}$ and its radius R. Let $\mathbf{X} = (\mathbf{x}_1^T \dots \mathbf{x}_n^T)^T$ denote the $(n \times p)$ data matrix with n observations and p variables. This implies that $\mathbf{a}$ is a p-dimensional variable while R is scalar. Figure 9.1 outlines the geometry of one solution for the SVDD in $p = 2$ dimensions. The variable ω_i represents the perpendicular distance from the boundary to an exterior point $\mathbf{x}_i$. For interior points, and points positioned exactly on the boundary, $\omega_i = 0$. The distance ω_i corresponding to an exterior point i can be written $\omega_i = \|\mathbf{x}_i - \mathbf{a}\| - R$, however, in the following we will use the closely related measure $\xi_i = \|\mathbf{x}_i - \mathbf{a}\|^2 - R^2$. To obtain a compact representation of the data, we wish to minimize both the hypersphere radius and the distances ξ_i to any exterior points. A formal description of this is given in the *loss-penalty* form known from penalized regression,

$$\arg\min_{R,\,\mathbf{a}} \sum_{i=1}^{n} \left(\|\mathbf{x}_i - \mathbf{a}\|^2 - R^2 \right)_+ + \lambda R^2. \tag{9.1}$$

Here, the loss function is formulated using the *hinge* loss function $(\cdot)_+$ [60] which is positive if its argument is positive, and zero otherwise. The penalty function is simply the squared radius. The trade-off between the loss and the penalty is governed by the regularization parameter λ. A large value of λ favors

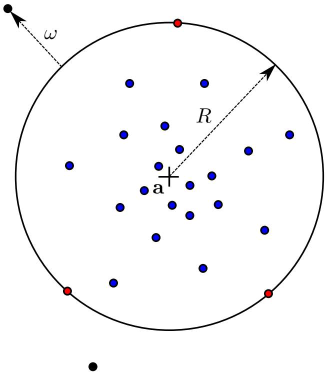


Figure 9.1: The geometry of the SVDD in two dimensions. Red, blue and black dots represent boundary points (3), data (20) and outliers (2) respectively. The hypersphere radius and center is denoted R and $\mathbf{a}$ respectively while ω is the distance from the boundary to an exterior point.

a solution with smaller radius and relatively larger ξ_i. If λ is small, the resulting hypersphere will be larger while the total distance from the boundary to exterior points shrinks.

In the following, we will reiterate the formulation and computation of the SVDD as proposed by Tax and Duin [151, 152]. To begin, Equation 9.1 is written as a constrained optimization problem using the ξ_i explicitly.

$$\min_{R,\mathbf{a},\xi_i} \sum_{i=1}^{n} \xi_i + \lambda R^2, \quad \text{subject to} \quad \|\mathbf{x}_i - \mathbf{a}\|^2 \le R^2 + \xi_i, \quad \xi_i \ge 0 \quad \forall i, \tag{9.2}$$

Fortunately, this seemingly more complex expression can be greatly simplified. First, the setup in Equation 9.2 is formulated as an unconstrained minimization problem using Lagrange multipliers $\alpha_i \ge 0$ and $\gamma_i \ge 0$,

$$L_p : \min_{R,\mathbf{a},\xi_i} \sum_{i=1}^{n} \xi_i + \lambda R^2 + \sum_{i=1}^{n} \alpha_i(\|\mathbf{x}_i - \mathbf{a}\|^2 - R^2 - \xi_i) - \sum_{i=1}^{n} \gamma_i \xi_i. \tag{9.3}$$

At the minimum, the derivative of each variable is zero, giving

$$\frac{\partial L_p}{\partial R} = 0 \quad \Leftrightarrow \quad \lambda = \sum_i \alpha_i, \tag{9.4}$$

$$\frac{\partial L_p}{\partial \mathbf{a}} = 0 \quad \Leftrightarrow \quad \mathbf{a} = \frac{\sum_i \alpha_i \mathbf{x}_i}{\sum_i \alpha_i} = \frac{\sum_i \alpha_i \mathbf{x}_i}{\lambda}, \tag{9.5}$$

$$\frac{\partial L_p}{\partial \xi_i} = 0 \quad \Leftrightarrow \quad \alpha_i = 1 - \gamma_i. \tag{9.6}$$

Carrying on, a set of useful identities are given by the Karush-Kuhn-Tucker complimentary slackness conditions,

$$\alpha_i(\|\mathbf{x}_i - \mathbf{a}\|^2 - R^2 - \xi_i) = 0, \tag{9.7}$$

$$\gamma_i \xi_i = 0. \tag{9.8}$$

Inserting Equations (9.4-9.6) into (9.3) results in the dual formulation which is to be maximized w.r.t. (9.4-9.6),

$$L_d : \max_{\alpha_i} \sum_{i=1}^{n} \alpha_i \mathbf{x}_i \mathbf{x}_i^T - \frac{1}{\lambda} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \mathbf{x}_i \mathbf{x}_j^T \quad : \quad 0 \leq \alpha \leq 1, \quad \sum_{i=1}^{n} \alpha_i = \lambda. \tag{9.9}$$

This is a quadratic optimization problem with linear constraints. As such, it can be solved for a particular value of λ using interior point methods [164].

The Lagrange multipliers α_i take on values in a limited range and have a distinct geometrical interpretation, where each α_i is connected to the behavior of a single observation $\mathbf{x}_i$. This can be inferred from the equations above in a number of steps. First, (9.7) reveals that $\alpha_i = 0$ for interior points. Second, (9.6) and (9.8) give that $\alpha_i = 1$ for exterior points. The dual problem in (9.9) is strictly concave and thus has a unique solution. This implies that each α_i is a continuous function of the regularization parameter λ. Too see this, assume that α_i is discontinuous at a point λ_l. This results in multiple optimal values of $\alpha_i(\lambda_l)$ and we have a contradiction. As a result of continuity, α_i must travel from 0 to 1 as point i passes the boundary from inside the hypersphere to the outside. To sum up, the valid range of the multipliers is $0 \leq \alpha_i \leq 1$. The range of the regularization parameter λ can now be established from Equation 9.4. The lower bound is $\lambda = 0$ for which all points are inside the boundary ($\forall \alpha_i = 0$). The upper bound is $\lambda = n$ which occurs when the hypersphere has shrunk to a point where all points are outside the boundary ($\forall \alpha_i = 1$).

The formulation in Equation 9.2 deviates slightly from the original setup of Tax and Duin [151] who use a regularization parameter $C = 1/\lambda$. As in [60] we favor the description above since $0 \leq \alpha \leq 1$ instead of $0 \leq \alpha \leq C$ which facilitates the interpretation of the coefficient paths $\alpha_i(\lambda)$.

A natural question that arises is that of the suitability of using a hypersphere to model the support of an arbitrary distribution. Clearly, this is most applicable to approximately spherical data, such as random samples drawn from a Gaussian distribution. The same question applies to support vector machines, where a hyperplane is not necessarily the best geometry for discriminating between two classes. The SVDD and SVM have the same remedy for this limitation. First, the dimensionality of the data is artificially increased using basis expansions $h(\mathbf{x})$; the classification problem is then solved in this extended space and the solution is projected back into the original domain. The result is a more flexible decision boundary with a geometry that is governed by the choice of basis function. This methodology applies to any statistical method such as regression or classification. The particular property of the SVDD and SVM is that its formulation (Equation 9.9) is specified using only the inner products $\mathbf{x}_i\mathbf{x}_j^T$. This makes it possible to replace the expanded inner product $h(\mathbf{x}_i)h(\mathbf{x}_j)^T$ by a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$, avoiding the explicit specification of the dimensionality of h. This is commonly known as the *kernel trick* in machine learning literature. Whenever possible, we use the more compact notation $K(\mathbf{x}_i, \mathbf{x}_j) = K_{i,j}$. Relevant choices of kernels are,

Linear kernel: $K_{i,j} = \mathbf{x}_i\mathbf{x}_j^T$

Polynomial kernel: $K_{i,j} = (1 + \mathbf{x}_i\mathbf{x}_j^T)^d$

Gaussian kernel: $K_{i,j} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma)$.

The linear kernel is equivalent to a first degree ($d = 1$) polynomial kernel and represents the original non-transformed formulation. Gaussian kernels are the most common choice for the SVDD as they provide a convenient generalization to the linear kernel. As σ increases, smoother and more coherent decision boundaries are obtained. For very large values of σ, the solutions approach those of a linear kernel [151]. Decreasing values of σ give more wiggly and clustered results. A polynomial kernel is not an appropriate choice for support description, as the boundary function is not sufficiently compact [151]. Hastie et al. [59] discuss a variety of kernels in more depth.

Rewriting Equation 9.9 using the more general kernel formulation yields,

$$L_d : \max_{\alpha_i} \sum_{i=1}^{n} \alpha_i K_{i,i} - \frac{1}{\lambda} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K_{i,j} \quad : \quad 0 \leq \alpha \leq 1, \quad \sum_{i=1}^{n} \alpha_i = \lambda. \tag{9.10}$$

In the remainder of this paper, this notation will be used.

In Figure 9.2, SVDD solutions are shown for $\lambda = 0$, $\lambda = n/3$ and $\lambda = 2n/3$ ($n = 50$). The top row contains the solutions obtained using a linear kernel, hence the circular boundary, while a Gaussian kernel with $\sigma = 1.8$ has been used for the results in the bottom row. In this example, the data is distributed according to a bi-modal Gaussian distribution with some overlap. The results obtained using linear kernels are obviously not satisfactory in this case. Referring to the solutions obtained using a Gaussian kernel; a note is made on the difference of the modeled geometry of the support. The solutions for $\lambda = 17$ exclude a few atypical points, resulting in a more compact and credible shape of the boundary function. At $\lambda = 33$, the boundary has separated into two clusters corresponding

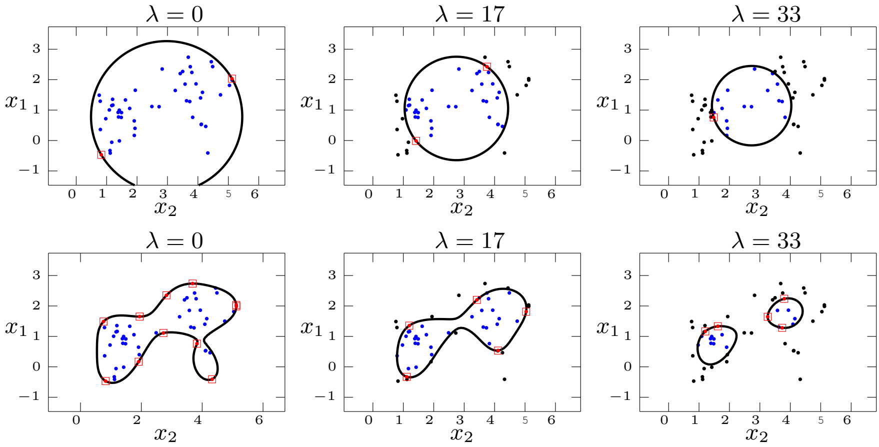


Figure 9.2: SVDD solutions for a small data set ($n = 50$) in two dimensions using different values of the regularization parameter λ. In the top row, a linear kernel is used, while a Gaussian kernel is used in the bottom row. Blue and black points denote interior and exterior points respectively, while squared red points denote points on the boundary.

to the apparent distribution of the sample. Such differences in the boundary function point to the usefulness of knowing the entire regularization path, and hence, all boundary functions.

9.2.1 The Regularization Path

In this section we will prove that the coefficient path of each α_i is a piecewise-linear function of λ, and propose an algorithm for their calculation using standard matrix algebra.

First, we define a couple of basic functions and notation that will be useful in the derivation that follows. The squared distance in feature space from the center of the hypersphere to a point $\mathbf{x}$ is,

$$f_\lambda(\mathbf{x}) = \|h(\mathbf{x}) - \mathbf{a}\|^2 = K(\mathbf{x}, \mathbf{x}) - \frac{2}{\lambda}\sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + \frac{1}{\lambda^2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j K_{i,j}. \tag{9.11}$$

The squared radius of the hypersphere can therefore be written $R^2 = f_\lambda(\mathbf{x}_k)$, where index k belongs to any point on the boundary ($\alpha_k \in (0, 1)$). Define by $\mathcal{I}$, $\mathcal{O}$ and $\mathcal{B}$ the sets containing indices i corresponding to interior, exterior and boundary points respectively, and let $n_\mathcal{I}$, $n_\mathcal{O}$, and $n_\mathcal{B}$ be the number of elements in these sets. The set $\mathcal{A} = \mathcal{I} \cup \mathcal{O} \cup \mathcal{B}$ contains the indices of all points. To determine which set a point $\mathbf{x}$ belongs to, we define a decision function,

$$\begin{aligned} g_\lambda(\mathbf{x}) &= f_\lambda(\mathbf{x}) - f_\lambda(\mathbf{x}_k) \\ &= K(\mathbf{x},\mathbf{x}) - K_{k,k} - \frac{2}{\lambda}\sum_{i=1}^{n}\alpha_i\big(K(\mathbf{x},\mathbf{x}_i) - K_{k,i}\big)k \in \mathcal{B}, \end{aligned} \tag{9.12}$$

which has $g_\lambda = 0$ for $\mathbf{x}$ on the boundary, $g_\lambda < 0$ for $\mathbf{x}$ interior and vice versa.

As discussed above, $\alpha_i = 1$ for $i \in \mathcal{O}$, $\alpha_i = 0$ for $i \in \mathcal{I}$, and $0 < \alpha_i < 1$ for $i \in \mathcal{B}$. There are four types of events where these sets change.

E_1 – Point i leaves $\mathcal{B}$ and joins $\mathcal{I}$; $\alpha_i \in (0,1) \rightarrow \alpha_i = 0$.

E_2 – Point i leaves $\mathcal{B}$ and joins $\mathcal{O}$; $\alpha_i \in (0,1) \rightarrow \alpha_i = 1$.

E_3 – Point i leaves $\mathcal{I}$ and joins $\mathcal{B}$; $\alpha_i = 0 \rightarrow \alpha_i \in (0,1)$.

E_4 – Point i leaves $\mathcal{O}$ and joins $\mathcal{B}$; $\alpha_i = 1 \rightarrow \alpha_i \in (0,1)$.

The idea of the algorithm is to start at a state where the solution is particularly simple to calculate, and then trace the solutions for changing values of λ until the entire regularization path is known. For reasons that will become clear later in this section, the most suitable starting point is at the end of the regularization interval ($\lambda = n$), corresponding to the minimal hypersphere radius. In this state, we know that $\mathcal{I} = \emptyset$, $\mathcal{B} = \emptyset$, $\mathcal{O} = \mathcal{A}$ and $\boldsymbol{\alpha} = \mathbf{1}^T$, a vector of all ones.

We will now derive a general expression for $\boldsymbol{\alpha}$ and λ at the next event, given an arbitrary configuration of $\mathcal{I}$, $\mathcal{O}$, $\mathcal{B}$, λ and $\boldsymbol{\alpha}$. From this state, λ is decreased until the next event occurs. As in [60], let λ_l be the value of the regularization parameter at step l. While $\lambda_{l+1} < \lambda < \lambda_l$, the sets remain static. Hence, $g_\lambda(\mathbf{x}_m) = 0, \forall\, m \in \mathcal{B}$ in this interval. Using this, Equation 9.12 can be expanded and rearranged into

$$\sum_{i\in\mathcal{B}}\alpha_i(K_{m,i} - K_{k,i}) = \frac{\lambda}{2}(K_{m,m} - K_{k,k}) - \sum_{i\in\mathcal{O}}(K_{m,i} - K_{k,i}) \;\; \forall m \in \mathcal{B}, k \in \mathcal{B}. \tag{9.13}$$

This results in $n_\mathcal{B}$ equations with $n_\mathcal{B}$ unknowns $\alpha_i, i \in \mathcal{B}$. However, for $m = k$, it is seen that (9.13) degenerates into $0 = 0$, making the system of equations rank deficient. We therefore replace the equation for $m = k$ with the auxiliary condition in Equation 9.4. In this way, we can find a unique solution for $\alpha_i, i \in \mathcal{B}$.

The procedure can be summarized in matrix form. Let $\mathbf{Y}$ be an $n \times n$ matrix where $\mathbf{Y}_{i,j} = K_{i,j} - K_{k,j},\ \forall\,(i,j) \in \mathcal{A}$ and let $\mathbf{y}$ be a length n vector with $y_i =$

$K_{i,i} - K_{k,k}$ $\forall i \in \mathcal{A}$. With the obvious definitions of submatrices, Equation 9.13 can be written

$$\mathbf{Y}_{\mathcal{B},\mathcal{B}}\boldsymbol{\alpha}_{\mathcal{B}} = \frac{\lambda}{2}\mathbf{y}_{\mathcal{B}} - \mathbf{Y}_{\mathcal{B},\mathcal{O}}\mathbf{1}_{n_{\mathcal{O}}}, \tag{9.14}$$

where $\mathbf{1}_{n_{\mathcal{O}}}$ is a vector of ones of length $n_{\mathcal{O}}$. This expression can be expanded to include the conditions $\boldsymbol{\alpha}_{\mathcal{I}} = 0$ and $\boldsymbol{\alpha}_{\mathcal{O}} = 1$. It also needs to be augmented to replace the degenerate equation corresponding to index k with the relation from Equation 9.4. We will now define matrices that implement this.

Let $\mathcal{B}_{-k}$ be the boundary set with index k removed. Let $\mathbf{Z}$ be the $n \times n$ identity matrix with $\mathbf{Z}_{\mathcal{B}_{-k},\mathcal{B}} = \mathbf{Y}_{\mathcal{B}_{-k},\mathcal{B}}$ and $\mathbf{Z}_{k,\mathcal{A}} = \mathbf{1}_n^T$. Let $\mathbf{z}$ be the length n zero vector with $\mathbf{z}_{\mathcal{B}_{-k}} = \mathbf{y}_{\mathcal{B}_{-k}}$ and $z_k = 2$. Finally, let $\mathbf{W}$ be the $n \times n$ zero matrix with $\mathbf{W}_{\mathcal{B}_{-k},\mathcal{O}} = -\mathbf{Y}_{\mathcal{B}_{-k},\mathcal{O}}$ and $\mathbf{W}_{\mathcal{O},\mathcal{O}} = \mathbf{I}_{n_{\mathcal{O}}}$ where $\mathbf{I}_{n_{\mathcal{O}}}$ is the identity matrix of size $n_{\mathcal{O}}$. The complete system of n equations for n unknowns is then

$$\mathbf{Z}\boldsymbol{\alpha} = \frac{\lambda}{2}\mathbf{z} + \mathbf{W}\mathbf{1}_n. \tag{9.15}$$

Providing $\mathbf{Z}$ is invertible, the resulting expression for $\boldsymbol{\alpha}$ becomes,

$$\boldsymbol{\alpha} = \frac{\lambda}{2}\mathbf{Z}^{-1}\mathbf{z} + \mathbf{Z}^{-1}\mathbf{W}\mathbf{1}_n \equiv \lambda\mathbf{p} + \mathbf{q}, \tag{9.16}$$

an expression that is linear in λ. This concludes the derivation of an expression for $\boldsymbol{\alpha}$ between two events.

Now that the functional form of each coefficient between two events is known, we need to disclose the valid range $[\lambda_{l+1}, \lambda_l]$ of λ. That is, we wish to find λ_{l+1} at which the next event occurs. We treat each of the four types of events defined above separately.

Event E_1 occurs for α_i, $i \in \mathcal{B}$ when $\alpha_i \to 0$. By setting (9.16) equal to 0 and solving for each value of λ, we get,

$$\lambda_i = -\frac{q_i}{p_i}, \quad i \in \mathcal{B}. \tag{9.17}$$

Similarly for E_2, $\alpha_i = 1$ when

$$\lambda_i = \frac{1 - q_i}{p_i}, \quad i \in \mathcal{B}. \tag{9.18}$$

For either E_3 or E_4 to occur, a point i in either $\mathcal{I}$ or $\mathcal{O}$ must join the boundary. At this stage, $g_\lambda(\mathbf{x}_i) = 0$. To find the values of λ at which each point joins the boundary, we insert $\boldsymbol{\alpha} = \lambda\mathbf{p} + \mathbf{q}$ into (9.12). The resulting expression is then set to 0 and solved for λ_i,

$$\lambda_i = \frac{2\mathbf{Y}_{i,\mathcal{A}}\mathbf{q}}{y_i - 2\mathbf{Y}_{i,\mathcal{A}}\mathbf{p}}. \tag{9.19}$$

Out of the candidates $\{\lambda_i\}$ for λ_{l+1} from (9.17), (9.18) and (9.19), the largest candidate smaller than λ_l must be the point at which the sets first change. Therefore, $\lambda_{l+1} = \max_i \lambda_i$ subject to $\lambda_i < \lambda_l$.

The boundary set $\mathcal{B}$ may at any stage of the algorithm become empty, resulting in a violation of Equation 9.4. One or more points from $\mathcal{O}$ must therefore join $\mathcal{B}$ concurrently. The calculation of candidates for λ_{l+1} in (9.19) will fail in this case, as a consequence of the new point not being placed on the current boundary. This behavior forces a discontinuity in the radius function, which must increase discretely to encompass the next point. Since $\boldsymbol{\alpha}(\lambda)$ is a continuous function, Equation 9.5 shows that the position of the hypersphere center $\mathbf{a}(\lambda)$ is also continuous. Hence, despite the discontinuity of the boundary function, the next point to join $\mathcal{B}$ can be established by finding the point in $\mathcal{O}$ with the smallest distance to the hypersphere center $\mathbf{a}$, that is, the point $i \in \mathcal{O}$ that minimizes Equation 9.11.

The entire process is summarized in Algorithm 9.1.

Algorithm 9.1 SVDD coefficient paths

```
Initialize $\lambda = n$ and $\alpha_i = 1 \ \forall i$.
while $\lambda > 0$ do
  if $n_{\mathcal{B}} = 0$ then
    Add index $i \in \mathcal{O}$ to the boundary set $\mathcal{B}$ that minimizes (9.11).
    Remove $i$ from $\mathcal{O}$.
  end if
  Given sets $\mathcal{I}$, $\mathcal{O}$ and $\mathcal{B}$, compute $\mathbf{p} = \mathbf{Z}^{-1}\mathbf{z}/2$ and $\mathbf{q} = \mathbf{Z}^{-1}\mathbf{W}\mathbf{1}_n$.
  Calculate $\lambda$ candidates according to $E_1$ using (9.17).
  Calculate $\lambda$ candidates according to $E_2$ using (9.18).
  Calculate $\lambda$ candidates according to $E_3$ using (9.19) with $i \in \mathcal{I}$.
  Calculate $\lambda$ candidates according to $E_4$ using (9.19) with $i \in \mathcal{O}$.
  Choose candidate $\lambda_{l+1}$ with the largest value smaller than $\lambda_l$.
  Calculate new coefficients, $\boldsymbol{\alpha} = \lambda_{l+1}\mathbf{p} + \mathbf{q}$.
  Update sets accordingly.
end while
```

Figure 9.3 shows the paths constructed from the data set presented in Figure 9.2, using a linear kernel (top) and a Gaussian kernel (bottom, $\sigma = 1.8$). Less rigid boundaries, such as those resulting from the use of a Gaussian kernel, tend to render more complex path patterns. Interpreting the paths for growing λ, it can be seen that the most common behavior for a point is to join the boundary from the inside and shortly after leave for the outside. However, there are several exceptions to this rule, despite the constant shrinkage of the hypersphere. This is an effect of the movement of the hypersphere center.

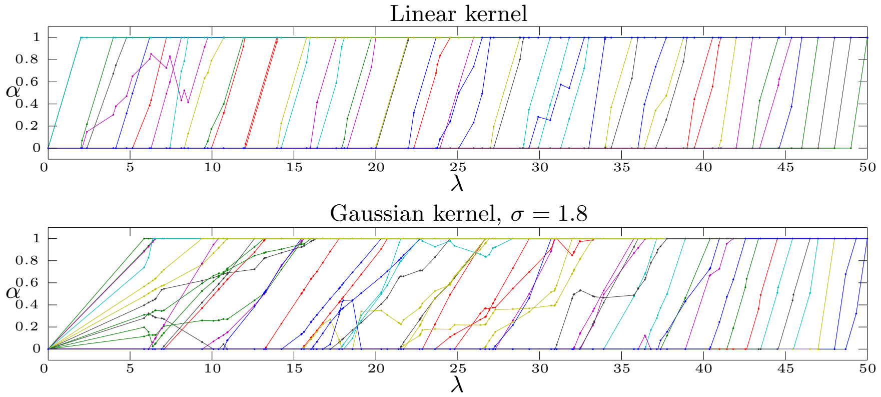


Figure 9.3: Example paths resulting from the SVDD analysis of the data set in Figure 9.2. The top figure shows the resulting path from using a linear kernel (spherical boundary), while a Gaussian kernel has been employed in the bottom figure. Each color denotes the path of a single observation.

9.2.2 Implementation

The computational complexity of computing the entire path is low. To increase efficiency, we solve (9.16) for points on the boundary only, i.e. using submatrices $\mathbf{Z}_{\mathcal{B},\mathcal{B}}$, $\mathbf{z}_{\mathcal{B}}$ and $\mathbf{W}_{\mathcal{B},\mathcal{O}}$. The remaining values of α_i $(i \in \mathcal{I} \cup \mathcal{O})$ remain static. Referring to Algorithm 9.1, Line 4 has complexity $O(n^2)$, Line 7 is $O(n_{\mathcal{B}}^3)$ while Lines 10-11 have complexity $O(n_{\mathcal{O}}n)$. Typically, $n_{\mathcal{B}} \ll n$. In our experience, the number of iterations is generally less than $2n$, although more than $5n$ iterations is possible for very dense data sets. The resulting overall complexity is $O(knn_{\mathcal{B}}^3)$, where k is some small value, usually $1 < k < 5$. Standard computation of the SVDD uses a single quadratic programming procedure for estimating a solution for a single value of λ and has complexity $O(n^3)$ [115]. A formal comparison of the complexities $O(knn_{\mathcal{B}}^3)$ and $O(n^3)$ is difficult, but our general experience is that they are roughly equal.

Due to the exclusive use of kernels, the method handles data with many variables well. The memory usage level is mainly due to the matrix $\mathbf{Y}_{\mathcal{B},\mathcal{B}}$, which can grow large for data sets with many observations and the use of very unconstrained decision boundaries.

A relevant question is whether multiple events can occur simultaneously. Starting at $\lambda = 0$ and tracing the path for increasing values of λ, we immediately come across a situation where multiple events occur. At $\lambda = 0$, 2 to $p+1$ points enter the boundary simultaneously. At least two points are necessary to completely specify a minimal hypersphere in an arbitrary space. Each additional point on the boundary removes one degree of freedom for the hypersphere, out of a total of $p+1$ degrees of freedom (specified by e.g. $\mathbf{a}$ and R). In theory, more than $p+1$ points may enter the boundary, providing that such additional points are placed at the exact same distance to the center $\mathbf{a}$, as the first $p+1$ points. We will assume that the coordinates of the observations are stochastic in some sense, and that with sufficient numerical precision, this situation does not occur. This reasoning explains why multiple events are equally unlikely along the rest of the path. There is, however, one exception. If, at $\lambda = 0$, exactly two points enter the boundary, these will be placed symmetrically on the hypersphere, with the line connecting the two observations going through $\mathbf{a}$. In this situation, $\mathbf{p}_{\mathcal{B}} = [0.5\ 0.5]^T$, meaning that both points will exit the boundary simultaneously at $\lambda = 2$. This is the only foreseeable multiple event that may occur along the path for $\lambda > 0$, and we check for it separately in our implementation (cf. Algorithm 9.2). More than two points being placed symmetrically on the boundary (such as in the corners of an equilateral triangle in $\mathbb{R}^2$) is unlikely for the same numerical reasons discussed above.

A more practical complication is described here through an example. Assume the next event was determined to be the addition of a point i from $\mathcal{O}$ to the boundary set $\mathcal{B}$ (E_4). At this instant, $\alpha_i = 1$ as the point just entered $\mathcal{B}$. In the following iteration, candidates for the next event are calculated. In particular, among the candidates for points leaving $\mathcal{B}$ for $\mathcal{O}$, point i will correspond to a value $\lambda_i = \lambda_l$, the current value of λ. This point and event (point i, $\mathcal{B} \rightarrow \mathcal{O}$) will not be considered a valid choice for λ_{l+1} since we require $\lambda_{l+1} < \lambda_l$. However, in an implementation where one must deal with finite precision, λ_i is not necessarily exactly equal to λ_l. Instead, it may be slightly lower (typically in the order of 10^{-15} in our implementation) making point i a valid candidate for the next event. In fact, this candidate will be selected as λ_i is so close to λ_l. One remedy for this problem is to treat values of λ_i that are sufficiently close to λ_l as equal. This is done by defining a threshold ε such that $\lambda_i \equiv \lambda_l \Leftrightarrow |\lambda_i - \lambda_l| < \varepsilon$. We choose to avoid this strategy as the proper size of ε may depend on the size and characteristics of the data. Instead we identify unrealizable combinations of events and avoid these explicitly in our implementation. There are four such combinations listed in the following. Note that these refer to chains of events concerning the *same point*.

1. $\mathcal{B} \rightarrow \mathcal{I}$ followed by $\mathcal{I} \rightarrow \mathcal{B}$.
2. $\mathcal{B} \rightarrow \mathcal{O}$ followed by $\mathcal{O} \rightarrow \mathcal{B}$.
3. $\mathcal{I} \rightarrow \mathcal{B}$ followed by $\mathcal{B} \rightarrow \mathcal{I}$.
4. $\mathcal{O} \rightarrow \mathcal{B}$ followed by $\mathcal{B} \rightarrow \mathcal{O}$.

These combinations of events are easily verified to be unrealizable; Equations 9.17, 9.18 and 9.19 all have single solutions for λ_i, and we know that this solution is $\lambda_i = \lambda_l$ in these cases. Thus, the conclusion is that there are no other values of $\lambda_i < \lambda_l$ such that these events can occur. By keeping track of the event that occurred in the previous iteration, points that may cause problems can be excluded from $\mathcal{I}$, $\mathcal{O}$ and $\mathcal{B}$ respectively when calculating λ_i.

Algorithm 9.2 describes a suggestion for a procedure that implements the proposed algorithm including the caveats discussed above. Here, E_P and i_P denote the previous event and the previous active point respectively.

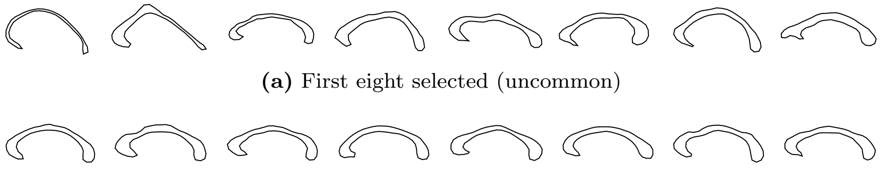
9.3 Applications

The SVDD with its original method for computation has found applications in a wide variety of fields, some of which are mentioned in Section 9.1. A survey of recent publications based on applications and extensions of the SVDD shows that interest is increasing. In such practices, the algorithm proposed in this paper makes it possible to make a more informed choice of the regularization parameter. While the improvement this offers is significant, we will not give any examples of this kind here as its application is straightforward. Instead, this section contains two examples where quantities from the regularization path are extracted which provide novel information on medical image data.

9.3.1 Commonality-based Ordering of Observations

The concept of this application is the interpretation of the SVDD as the establishment of an order among the observations of a data set. Tracing the regularization path from $\lambda = 0$ to n, the first observation to leave the boundary will be the one regarded as the least common observation of the data set by the SVDD, the next observation to become an exterior point will be regarded the second least common observation, and so on. Points that rejoin the boundary from the exterior are registered as they leave the boundary for the last time. The idea that the order of exclusion is reflecting the sample set density is the reason we refer to this method as *commonality based*; more common objects, residing in regions of feature space that are more densely populated by similar observations, will be excluded later along the regularization path than more uncommon examples. The implementation of this method is elementary, as the ordering is established directly from $\mathcal{O}$.

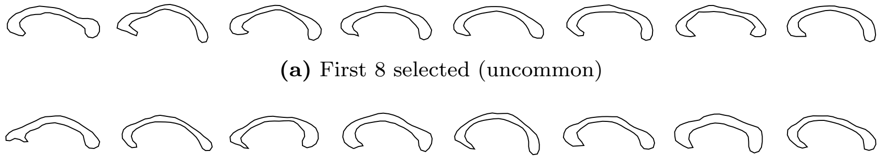
To illustrate this application, we have established an ordering in a data set consisting of 62 Procrustes-aligned outlines in two dimensions of the mid-sagittal



(a) First eight selected (uncommon)

(b) Last eight selected (common)

Figure 9.4: Ordering established by the SVDD regularization path. Note the increased dissimilarity among the outliers, as well as the increased similarity among later samples.



(a) First 8 selected (uncommon)

(b) Last 8 selected (common)

Figure 9.5: Ordering established by successive maximization of Mahalanobis distance.

cross-section of the corpus callosum brain structure [111]. Each outline is regarded as one observation, consisting of $p = 2 \cdot 78$ variables (78 landmarks in two dimensions). We use a Gaussian kernel with $\sigma = 1$ for the SVDD estimation.

To put the proposed method into perspective, we have also established an ordering using successive maximization of the squared Mahalanobis distance,

$$d_M^2 = (\mathbf{x}_i - \bar{\mathbf{x}})\Sigma^{-1}(\mathbf{x}_i - \bar{\mathbf{x}})^T. \tag{9.20}$$

Starting with the full data set, at each step, the observation with the largest distance with respect to the current data set is removed and the covariance matrix Σ and the mean $\bar{\mathbf{x}}$ is recalculated. For n shapes, this is performed $n-1$ times, thus establishing an ordering.

Figures 9.4 and 9.5 show the first (least common) and last (most common) eight ordered observations of the SVDD path method and the Mahalanobis method respectively.

The Mahalanobis distance measure is based on the shape of the covariance matrix and assumes an ellipsoidal distribution. Due to the use of kernels, the SVDD is able to model more complex distributions, giving better estimates of the commonality of each observation. This is particularly apparent among the common samples in Figure 9.4. The variance is clearly lower for the SVDD-based

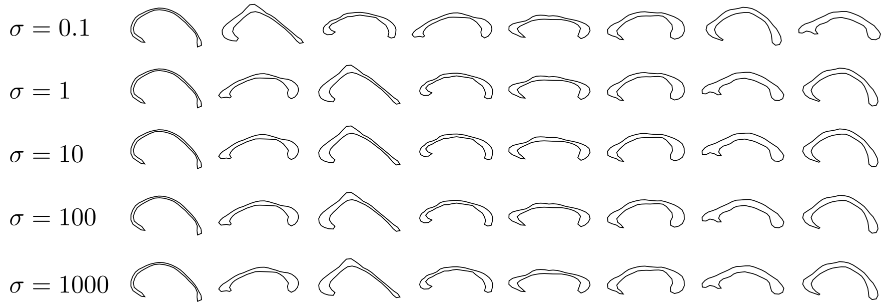


Figure 9.6: Ordering established by the SVDD using various choices of the kernel parameter σ. The results are surprisingly insensitive to this variable.

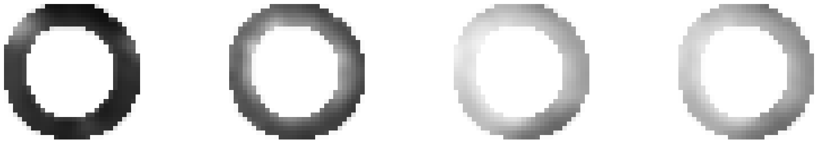
ordering than for the Mahalanobis-based counterpart. Moreover, the SVDD is significantly more efficient; the Mahalanobis-based method has approximate complexity $O(n^4)$.

The ordering is dependent on the kernel parameter σ when using a Gaussian kernel, and similar parameters for many other types of kernels. As discussed in Section 9.2 and shown in Figure 9.2, various choices of the kernel parameter σ lead to significantly different behavior of decision boundary. This invariably leads to variations in the results of any application using the SVDD together with hyper parameters. As an illustrative example, Figure 9.6 shows the eight least common corpus callosum outlines for very dissimilar values of σ. Interestingly, the results are relatively insensitive to the choice of σ in this application.

9.3.2 Ischemic Segment Detection from Cardiac MR Images

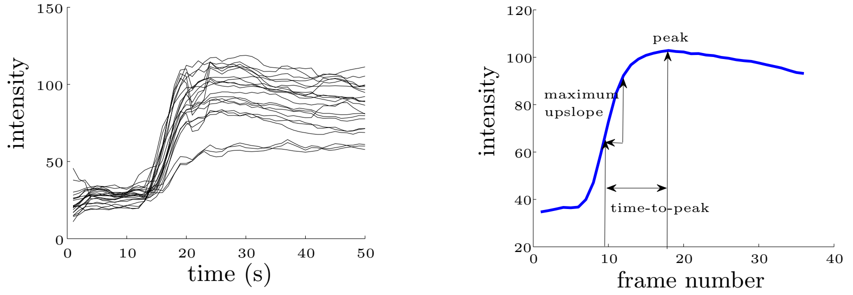
Early treatment of ischemic heart disease requires early detection. Magnetic resonance imaging (MRI) has emerged as an important tool for assessing myocardial perfusion [86]. The reduction of myocardial perfusion is a sensitive indicator for myocardial ischemia, as myocardial blood flow is directly correlated to myocardial oxygen supply. In the present study, a set of perfusion MR images is analyzed with the goal of detecting ischemic segments of the myocardium. The data set consists of a sequence of 50 myocardial perfusion short-axis MR images, obtained from ten freely breathing patients, all having acute myocardial infarction. Each image consists of four spatial slices of the myocardium.

To obtain pixel-wise correspondences between subjects, the perfusion MR images have been spatially normalized using a method for image registration based on the active appearance model with landmark correspondences optimized by a minimum description length approach [106, 143]. Four different time frames from one registered slice can be seen in Figure 9.7. Using the pixel-wise correspondence, intensity curves from a selection of pixels may be plotted, reflecting the passage of the contrast agent as shown in Figure 9.8(a). In the current setting, one observation is represented by the time-series of intensities in a given voxel ($n \approx 500, p = 50$).



(a) Frame 1. **(b)** Frame 16. **(c)** Frame 31. **(d)** Frame 46.

Figure 9.7: Different registered frames of one of the slices of the perfusion MR images.



(a) Intensities of a selection of pixels from one subject.

(b) Idealized curve with explanation of conventional perfusion parameters.

Figure 9.8: Intensity curves (experimental and theoretical) for a selection of myocardial segments.

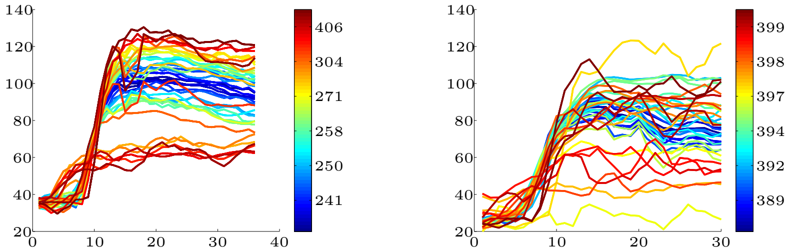
Normally perfusion is assessed using three perfusion parameters; maximum upslope, peak and time-to-peak for the detection of ischemic heart segments, as illustrated in Figure 9.8(b). We propose to replace or complement these parameters by a quantity estimated from the SVDD regularization path. This measure is referred to as a *generalized distance*, and does not require selection of a particular value of λ. An in-depth description of this method is given by Hansen et al. [55], but will be briefly reviewed in the following.

It is the hypothesis of the method that the smaller ischemic segments appear as outliers using the SVDD, and the larger healthy segments as inliers. We do not know, however, if an ischemic segment is present, nor its size. Analyzing the intensity profiles using the SVDD divides the observations into two classes, separating curves with unusual behavior from more common ones. The parameter related to the proportion of curves considered abnormal is given by λ. An appropriate value of λ is, however, unknown. If λ is too small, the ischemic segments will be considered normal data, and if it is too large, both ischemic segments and healthy segments will be considered as outliers. A distance measure that retains the contrast between inliers and outliers, while ensuring that outliers are not accepted as inliers is calculated by integrating the distance function $f_\lambda(\mathbf{x})$ from Equation 9.11 over the whole range of λ,

$$\phi(\mathbf{x}) = \int_0^n f_\lambda(\mathbf{x}) d\lambda. \tag{9.21}$$

This can be computed efficiently using the entire regularization path, and is calculated for each pixel, slice and image. The idea is that ischemic segments will exhibit larger generalized distances $\phi(\mathbf{x})$ compared to healthy tissue.

In Figure 9.9 the intensities from a collection of pixels of a single slice are shown for all time frames, colored according to their generalized distance. Notice how the curves that rapidly gain and drop in intensity are also considered outliers in spite of their large upslope. This was one motivation for the presented work, to isolate segments with a very noisy response. Such segments are mainly positioned around the ischemic segment, and they appear to form another feature in the detection.

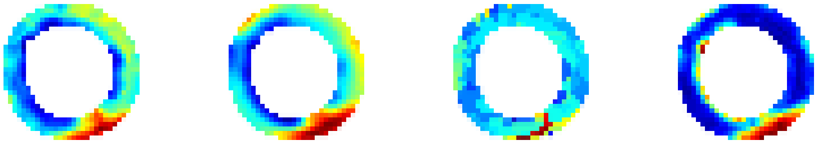


(a) Generalized distances, patient 7. **(b)** Generalized distances, patient 3.

Figure 9.9: Pixel-wise intensity plots. Different pixels are colored according to the *generalized distance*, shown in the adjacent color-bar. The pixels are evenly chosen in the whole range of distances from the 3rd slice.

The generalized SVDD-based distance shown in Figure 9.10(d) is seen to correspond well to the perfusion parameters in Figure 9.10(a-c). A benefit of the SVDD is that the noise in healthy segments is reduced. The excellent correspondence visible is not always present. In Figure 9.11, the normal (blue) area is smaller, and the ischemic segment appears less precisely defined and localized.

A caveat of the proposed distance measure is that a relatively large number of normal observations is needed for the method to work properly. However, many applications in medical imaging look for sparse effects in large volumes of data, a setting which fits the proposed method.



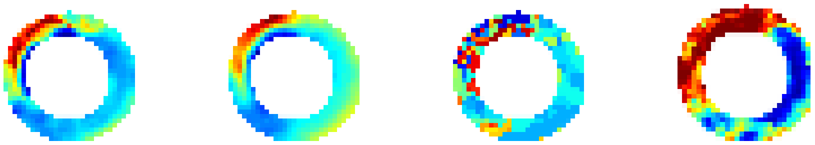
(a) Maximum upslope

(b) Peak values

(c) Time to peak

(d) Generalized distance $\phi(\mathbf{x})$

Figure 9.10: Results for patient 7. Colors are chosen to let red indicate abnormality/ischemia whereas healthy tissue is shown in blue. (a) A low maximum upslope is an indicator of ischemia. (b) A low peak value reflects ischemia. (c) A long time to peak measure indicates ischemia. (d) Blue corresponds to small generalized distance and red to higher generalized distance. This is seen to correspond well to the other measures.



(a) Maximum upslope

(b) Peak values

(c) Time to peak

(d) Generalized distance $\phi(\mathbf{x})$

Figure 9.11: Results for patient 3. Colors correspond to those in Figure 9.10.

9.4 Conclusions

This paper has presented an algorithm for efficiently calculating the entire regularization path of the support vector domain description. This means that the classification results for any conceivable choice of the regularization parameter become available. Knowledge of this path was shown to provide new tools suitable for the analysis of medical image data. First, we demonstrated how path information can be used to establish a sensible ordering among a set of clinical observations, based on their commonality. The method resulted in a visually more pleasing result compared to a Mahalanobis-based alternative, and has better computational efficiency. Second, a generalized distance measure was proposed and applied to the detection of ischemic segments of the myocardium from cardiac perfusion MR images. The generalized distance approach demonstrated unsupervised, non-parametric and computationally efficient detection of such segments in cases where the proportion of infarcted tissue is low.

The obvious application of the algorithm is the possibility of making a more informed choice of the regularization parameter, and we anticipate that our results will have an impact on any method for classification, clustering or novelty detection based on the support vector domain description.

Acknowledgements

The authors extend their gratitude to Egill Rostrup and Charlotte Ryberg from the Danish Research Center for Magnetic Resonance, Copenhagen University Hospital, Hvidovre, Denmark, for supplying the corpus callosum shape data. Karl Sjöstrand and Michael Sass Hansen are funded by the ITMAN graduate school, Technical University of Denmark.

Algorithm 9.2 Suggestion for an SVDD implementation. E_P and i_P denote the previous event and the previous active point respectively.

```
Initialize $\lambda = n$ and $\boldsymbol{\alpha} = 1$.
Initialize $\mathcal{B}$ to $i \in \mathcal{O}$ that minimizes (9.11) and initialize sets accordingly.
$E_P = E_4$
while $n_{\mathcal{O}} > 0$ do
  Given sets $\mathcal{I}$, $\mathcal{O}$ and $\mathcal{B}$, compute $\mathbf{p} = \mathbf{Z}^{-1}\mathbf{z}/2$ and $\mathbf{q} = \mathbf{Z}^{-1}\mathbf{W}\mathbf{1}_n$.
  $\mathcal{T} = \mathcal{B}$
  if $E_P = E_3$ then
    $\mathcal{T} = \mathcal{B} - i_P$
  end if
  Calculate $\lambda$ candidates according to $E_1$ using (9.17) with $i \in \mathcal{T}$.
  $\mathcal{T} = \mathcal{B}$
  if $E_P = E_4$ then
    $\mathcal{T} = \mathcal{B} - i_P$
  end if
  Calculate $\lambda$ candidates according to $E_2$ using (9.18) with $i \in \mathcal{T}$.
  $\mathcal{T} = \mathcal{I}$
  if $E_P = E_1$ then
    $\mathcal{T} = \mathcal{I} - i_P$
  end if
  Calculate $\lambda$ candidates according to $E_3$ using (9.19) with $i \in \mathcal{T}$.
  $\mathcal{T} = \mathcal{O}$
  if $E_P = E_2$ then
    $\mathcal{T} = \mathcal{O} - i_P$
  end if
  Calculate $\lambda$ candidates according to $E_4$ using (9.19) with $i \in \mathcal{T}$.
  Choose candidate $\lambda_{l+1}$ with the largest value smaller than $\lambda_l$.
  Calculate new coefficients, $\boldsymbol{\alpha} = \lambda_{l+1}\mathbf{p} + \mathbf{q}$.
  Update sets and $E_P$ accordingly.
  if $n_{\mathcal{B}} = 0$ then
    if $n_{\mathcal{O}} \leq 2$ then
      $\mathbf{B} = \mathcal{O}$, $\mathcal{O} = \emptyset$
    else
      Add index $i \in \mathcal{O}$ to the boundary set $\mathcal{B}$ that minimizes (9.11), remove
      $i$ from $\mathcal{O}$.
      $E_P = E_4$
    end if
  end if
  $\mathbf{p} = \boldsymbol{\alpha}/\lambda$, $\mathbf{q} = \mathbf{0}$, $\boldsymbol{\alpha} = \mathbf{0}$, $\lambda_{l+1} = 0$
  $\mathcal{I} = \mathcal{A}$, $\mathcal{B} = \emptyset$.
end while
```

List of Tables

List of Figures

List of Algorithms

Bibliography

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory*, pages 267–281, 1973.

[2] L.S. Allen, M.F. Richey, Y.M. Chai, and R.A. Gorski. Sex differences in the corpus callosum of the living human being. *Neuroscience*, 11(4):933–942, April 1991.

[3] John Ashburner and Karl J Friston. Nonlinear spatial normalization using basis functions. *Human Brain Mapping*, 7(4):254–266, 1999.

[4] M.A. Babyak. What you see may not be what you get: a brief nontechnical introduction to overfitting in regression-type models. *Psychosomatic Medicine*, 66:411–421, 2004.

[5] T. Ban and S. Abe. Implementing multi-class classifiers by one-class classification methods. In *International Joint Conference on Neural Networks, 2006, IJCNN '06.*, pages 327–332, 2006.

[6] A. Banerjee, P. Burlina, and C. Diehl. A support vector method for anomaly detection in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 44 (8):2282–2291, 2006.

[7] R.E. Bellman. *Adaptive control processes: a guided tour.* Princeton University Press, 1961.

[8] A. Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, Dec 2001.

[9] P. Bermudez and R.J. Zatorre. Sexual dimorphism in the corpus callosum: Methodological considerations in MRI morphometry. *NeuroImage*, 13(6):1121–1130, 2001.

[10] F.L. Bookstein. The line-skeleton. *Computer Graphics and Image Processing*, 11(2): 123–137, 1979.

[11] F.L. Bookstein. Shape and the information in medical images: a decade of the morphometric synthesis. *Mathematical Methods in Biomedical Image Analysis, 1996., Proceedings of the Workshop on*, pages 2–12, 1996.

[12] F.L. Bookstein. Landmark methods for forms without landmarks: morphometrics of group differences in outline shape. *Medical Image Analysis*, 1(3):225–243, 1997.

[13] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34(3):344–371, June 1986.

[14] R. Bowden. *Learning non-linear Models of Shape and Motion.* PhD thesis, Dept Systems Engineering, Brunel University, Uxbridge, Middlesex, UB8 3PH, UK., 1999.

[15] C. Brechbuhler, G. Gerig, and O. Kubler. Parametrization of closed surfaces for 3-d shape description. *Computer Vision and Image Understanding*, 61(2):154–170, 1995.

[16] L. Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37 (4):373–384, 1995.

[17] Jorge Cadima and Ian T Jolliffe. Loadings and correlations in the interpretation of principal components. *Journal of Applied Statistics*, 22(2):203–214, 1995.

[18] C. Chennubhotla and A. Jepson. Sparse PCA extracting multi-scale structure from data. *Proceedings of the IEEE International Conference on Computer Vision*, 1:641–647, 2001.

[19] J.Y. Choi, K.H. Im, and W.-S. Kang. SVDD-based method for fast training of multi-class support vector classifier. *Lecture Notes in Computer Science*, 3971:991–996, 2006.

[20] S. Clarke, H. van der Loos R. Kraftsik, and G.M. Innocenti. Forms and measures of adult and developing human corpus callosum: Is there sexual dimorphism? *Journal of comparative neurology*, 280:213–230, 1989.

[21] J. Cohen and P. Cohen. *Applied multiple regression / correlation analysis for the behavioral sciences.* John Wiley & sons, 1975.

[22] T. F. Cootes and C. J. Taylor. A mixture model for representing shape variation. *Image and Vision Computing*, 17(8):567–574, 1999.

[23] T.F. Cootes and C.J. Taylor. A mixture model for representing shape variation. In *Proceedings of the 8th British Machine Vision Conference, BMVC*, volume 1, pages 110–119. BMVA Press, 1997.

[24] T.F. Cootes and C.J. Taylor. *Statistical Models of Appearance for Computer Vision.* Tech. report, University of Manchester, 2001.

[25] T.F. Cootes, D. Cooper, C.J. Taylor, and J. Graham. A trainable method of parametric shape description. In *Proc. British Machine Vision Conference*, pages 54–61. Springer-Verlag, 1991.

[26] T.F. Cootes, C.J. Taylor, D. Cooper, and J. Graham. Training models of shape from sets of examples. In *Proc. British Machine Vision Conf., BMVC92*, pages 9–18, 1992.

[27] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models – their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[28] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In *Proc. of European Conf. on Computer Vision 1998*, volume 1407 of *Lecture Notes in Computer Science*, pages 484–498. Springer, 1998.

[29] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):681–685, 2001.

[30] O. Crouzon. Une nouvelle famille atteinte de dysostose cranio-faciale héréditère. *Bull Mem Soc Méd Hôp Paris*, 39:231–233, 1912.

[31] A. d'Aspremont, L. El Ghaoui, M.I. Jordan, and G.R.G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems (NIPS) 2004*, July 2005.

[32] I. Daubechies and Y. Meyer. Ten lectures on wavelets. *Bulletin of the American Mathematical Society*, 28(2):350–359, 1993.

[33] C. Davatzikos, M. Vaillant, S.M. Resnick, J.L. Prince, S. Letovsky, and R.N. Bryan. *Journal of Computer Assisted Tomography*, 20(1):88–97, 1996.

[34] C. Davatzikos, Xiaodong Tao, and Dinggang Shen. Hierarchical active shape models, using the wavelet transform. *Medical Imaging, IEEE Transactions on*, 22(3):414–423, 2003.

[35] R.H. Davies, C.J. Twining, P.D. Allen, T.F. Cootes, and C.J. Taylor. Shape discrimination in the hippocampus using an MDL model. *Information Processing in Medical Imaging, IPMI 2003.*, 18:38–50, 2003.

[36] A.C. Davison and D.V. Hinkley. *Bootstrap Methods and Their Application*. Cambridge University Press, 5th edition, 2003.

[37] X. Dong, W. Zhaohui, and Z. Wanfeng. Support vector domain description for speaker recognition. In *Neural Networks for Signal Processing XI, 2001. Proceedings of the 2001 IEEE Signal Processing Society Workshop*, pages 481–488, 2001.

[38] I.L. Dryden and K.V. Mardia. *Statistical Shape Analysis*. John Wiley & Sons, 1999.

[39] A. Dubb, R. Gur, B. Avants, and J. Gee. Characterization of sexual dimorphism in the human corpus callosum. *Neuroimage*, 20(1):512–519, Sep 2003.

[40] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–451, 2004.

[41] V.P. Eswarakumar, M.C. Horowitz, R. Locklin, G.M. Morriss-Kay, and P. Lonai. A gain-of-function mutation of fgfr2c demonstrates the roles of this receptor variant in osteogenesis. *Proc Natl Acad Sci, U.S.A.*, 101:12555–12560, 2004.

[42] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

[43] Y. Fan, D. Shen, R.C. Gur, R.E. Gur, and C. Davatzikos. Compare: Classification of morphological patterns using adaptive regional elements. *IEEE Transactions on Medical Imaging*, 26(1):93–105, 2007.

[44] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[45] J.H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.

[46] K.J. Friston, A.P. Holmes, K.J. Worsley, J.B. Poline, C. Frith, and R.S.J. Frackowiak. Statistical parametric maps in functional imaging: A general linear approach. *Human Brain Mapping*, 2:189–210, 1995.

[47] W.J. Fu. Penalized regressions: The bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7(3):397, 1998.

[48] E.I. George and D.P. Foster. Calibration and empirical bayes variable selection. *Biometrika*, 87(4):731–748, 2000.

[49] E.I. George and R.E. McCulloch. Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

[50] Daniel Gervini and Valentin Rousson. Criteria for evaluating dimension-reducing components for multivariate data. *American Statistician*, 58(1):72–76, 2004.

[51] P. Golland, W. Eric, and L. Grimson. Fixed topology skeletons. *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, 1:10–17, 2000.

[52] P. Golland, W.E.L. Grimson, M.E. Shenton, and R. Kikinis. Deformation analysis for shape based classification. *Information Processing in Medical Imaging. 17th International Conference, IPMI 2001. Proceedings (Lecture Notes in Computer Science Vol.2082)*, pages 517–30, 2001.

[53] P. Golland, W.E.L. Grimson, M.E. Shenton, and R. Kikinis. Detection and analysis of statistical differences in anatomical shape. *Medical Image Analysis*, 9(1):69–86, 2005.

[54] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.

[55] M.S. Hansen, H. Ólafsdóttir, K. Sjöstrand, H.B.W. Larsson, M.B. Stegmann, and R. Larsen. Ischemic segment detection using the support vector domain description. In *International Symposium on Medical Imaging 2007, San Diego, CA, USA*. The International Society for Optical Engineering (SPIE), Feb 2007.

[56] M.S. Hansen, H. Olafsdóttir, T.A. Darvann, N.V. Hermann, E. Oubel, R. Larsen, B.K. Ersbøll, A.F. Frangi, P. Larsen, C.A. Perlyn, G.M. Morris-Kay, and S. Kreiborg. Estimation of independent non-linear deformation modes for analysis of craniofacial malformations in crouzon mice. In Jeffrey A. Fessler Miles Wernick, editor, *2007 IEEE International Symposium on Biomedical Imaging*. IEEE, may 2007, accepted.

[57] H.H. Harman. *Modern Factor Analysis*. The University of Chicago Press, 1967.

[58] T. Hastie and R. Tibshirani. Efficient quadratic regularization for expression arrays. *Biostatistics*, 5(3):329–340, 2004.

[59] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[60] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, Oct 2004.

[61] R. Hausman. Constrained multivariate analysis. In S.H. Zanakis and J.S. Rustagi, editors, *Optimization in Statistics*, number 19 in Studies in the management sciences, pages 137–151. North-Holland, Amsterdam, 1982.

[62] T. Heap and D. Hogg. Improving specificity in PDMs using a hierarchical approach, 1997.

[63] K.B. Hilger, M.B. Stegmann, and R. Larsen. A noise robust statistical model for image representation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2002, 5th Int. Conference, Tokyo, Japan*, volume 2488 of *LNCS*, pages 444–451, sep 2002.

[64] K.B. Hilger, R. Larsen, and M. Wrobel. Growth modeling of human mandibles using non-Euclidean metrics. *Medical Image Analysis*, 7:425–433, 2003.

[65] A.E. Hoerl and R.W. Kennard. Ridge regression: Biased estimation from nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[66] P. Horst. *Factor Analysis of Data Matrices*. Holt, Rinehart and Winston, New York, 1965.

[67] A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[68] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, June 2001.

[69] J.N.R. Jeffers. Two case studies in the application of principal component analysis. *Applied Statistics*, 16(3):225–236, 1967.

[70] H. Jokinen, C. Ryberg, H. Kalska, R. Ylikoski, E. Rostrup, M.B. Stegmann, G. Waldemar, S. Madureira, J.M. Ferro, E.C.W. van Straaten, P. Scheltens, F. Barkhof, F. Fazekas, R. Schmidt, L. Pantoni, D. Inzitari, and T. Erkinjuntti. Corpus callosum atrophy is associated with mental slowing and executive deficits in subjects with age-related white matter hyperintensities. the ladis study. *Journal of neurology, neurosurgery, and psychiatry*, 2006.

[71] I. T. Jolliffe. Rotation of ill-defined principal components. *Applied Statistics*, 38(1): 139–147, 1989.

[72] Ian T Jolliffe. Rotation of principal components: Choice of normalization constraints. *Journal of Applied Statistics*, 22(1):29–36, 1995.

[73] I.T. Jolliffe. *Principal Component Analysis*. Springer, New York, 1986.

[74] I.T. Jolliffe, N.T. Trendafilov, and M. Uddin. A modified principal component technique based on the LASSO. *Journal of Computational and Graphical Statistics*, 12(3):531–547, 2003.

[75] S. Joshi, S. Pizer, P.T. Fletcher, P. Yushkevich, A. Thall, and J.S. Marron. Multiscale deformable model segmentation and statistical shape analysis using medial descriptions. *IEEE Transactions on Medical Imaging*, 21(5):538–550, 2002.

[76] H.F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23:187–200, 1958.

[77] A. Kelemen, G. Szekely, and G. Gerig. Three-dimensional model-based segmentation of brain mri. *Biomedical Image Analysis, 1998. Proceedings. Workshop on*, pages 4–13, 1998.

[78] D.G. Kendall. The diffusion of shape. *Advances in Applied Probability*, 9:428–430, 1977.

[79] S. Kreiborg. *Crouzon Syndrome - A Clinical and Roentgencephalometric Study*, 1981. Doctorate thesis, Institute of Orthodontics, The Royal Dental College, Copenhagen.

[80] C. Lai, D.M.J. Tax, R.P.W. Duin, E. Pekalska, and P. Paclik. A study on combining image representations for image classification and retrieval. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(5):867–890, 2004.

[81] R. Larsen. Shape modelling using maximum autocorrelation factors. In Ivar Austvoll, editor, *Proceedings of the Scandinavian Image Analysis Conference (SCIA'01)*, pages 98–103, Bergen, Norway, jun 2001.

[82] R. Larsen. Decomposition using maximum autocorrelation factors. *Journal of Chemometrics*, 16(8-10):427–435, 2002.

[83] R. Larsen and K.B. Hilger. Statistical 2D and 3D shape analysis using non-Euclidean metrics. *Medical Image Analysis*, 7(4):417–423, 2003.

[84] R. Larsen, H. Eiriksson, and M.B. Stegmann. Q-MAF shape decomposition. In Wiro J. Niessen and Max A. Viergever, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2001, 4th International Conference, Utrecht, The Netherlands*, volume 2208 of *LNCS*, pages 837–844. Springer, 2001.

[85] R. Larsen, K.B. Hilger, and M.C. Wrobel. Statistical 2D and 3D shape analysis using non-Euclidean metrics. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2002, 5th Int. Conference, Tokyo, Japan*, volume 2489 of *Lecture Notes in Computer Science*, pages 428–435. Springer, 2002.

[86] H.B.W. Larsson, T. Fritz-Hansen, E. Rostrup, P. Ring, and O. Henriksen. Myocardial perfusion modelling using MRI. *Magnetic Resonance in Medicine*, 35:716–726, 1996.

[87] L. Le Briquer and J.C. Gee. Design of a statistical model of brain shape. *Information Processing in Medical Imaging. 15th International Conference, IPMI'97. Proceedings*, pages 477–82, 1997.

[88] D. Lee and J. Lee. Domain described support vector classifier for multi-classification problems. *Pattern Recognition*, 40(1):41–51, 2007.

[89] D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[90] K. Lee, D.-W. Kim, K.H. Lee, and D. Lee. Density-induced support vector data description. *IEEE Transactions on Neural Networks*, 18(1):284–289, 2007.

[91] S.-W. Lee, J. Park, and S.-W. Lee. Low resolution face recognition based on support vector data description. *Pattern Recognition*, 39(9):1809–1812, 2006.

[92] M.E. Leventon, W.E.L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, 1:316–323, 2000.

[93] D. Loeckx, F. Maes, D. Vandermeulen, and P. Suetens. Temporal subtraction of thorax CR images using a statistical deformation model. *Medical Imaging, IEEE Transactions on*, 22(11):1490–1504, 2003.

[94] A.M.C. Machado and J.C. Gee. Atlas warping for brain morphometry. *Proceedings of the SPIE - The International Society for Optical Engineering*, 3338:642–51, 1998.

[95] A.M.C. Machado, J.C. Gee, and M.F.M. Campos. Structural shape characterization via exploratory factor analysis. *Artificial Intelligence in Medicine*, 30(2):97–118, 2004.

[96] C.L. Mallows. Some comments on cp. *Technometrics*, 15(4):661–675, 1973.

[97] D.W. Marquardt. Generalized inverses, ridge regression, biased linear estimation, and nonlinear estimation. *Technometrics*, 12(3):591–612, 1970.

[98] S. C. Mitchell, B. P. F. Lelieveldt, H. G. Bosch, J. H. C. Reiber, and M. Sonka. Disease characterization of active appearance model coefficients. *Proc. of SPIE - The International Society for Optical Engineering*, 5032 II:949–957, 2003.

[99] B. Moghaddam, Y. Weiss, and S. Avidan. Spectral bounds for sparse PCA: Exact & greedy algorithms. In Y. Weiss, B. Scholkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems (NIPS)*, December 2005.

[100] A. Mohamed, E.I. Zacharaki, D. Shen, and C. Davatzikos. Deformable registration of brain tumor images via a statistical model of tumor-induced deformation. *Medical Image Analysis*, 10(5):752–763, 2006.

[101] L. Molgedey and H.G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Physical Review Letters*, 72(23):3634–3637, 1994.

[102] J.E. Moody. Note on generalization, regularization and architecture selection in nonlinear learning systems. *Neural Networks for Signal Processing [1991]., Proceedings of the 1991 IEEE Workshop*, pages 1–10, 1991.

[103] H. Neudecker. On the matric formulation of kaiser's varimax criterion. *Psychometrika*, 46(3):343–345, 1981.

[104] T.E. Nichols and A.P. Holmes. Nonparametric permutation tests for functional neuroimaging: A primer with examples. *Human Brain Mapping*, 15(1):1–25, 2002.

[105] H Ojelund, H Madsen, and P Thyregod. Original research articles - calibration with absolute shrinkage. *Journal of Chemometrics*, 15(6):497–510, 2001.

[106] H. Ólafsdóttir, M.B. Stegmann, and H.B.W. Larsson. Automatic assessment of cardiac perfusion MRI. In C. Barillot, D.R. Haynor, and P. Hellier, editors, *Medical image computing and computer assisted intervention, MICCAI*, volume 36 of *Lecture Notes in Computer Science*, pages 1060–1061. Springer, Sep 2004.

[107] H. Ólafsdóttir, T.A. Darvann, B.K. Ersbøll, E. Oubel, N.V. Hermann, A.F. Frangi, P. Larsen, C.A. Perlyn, G.M. Morriss-Kay, and S. Kreiborg. A craniofacial statistical deformation model of wild-type mice and Crouzon mice. In *International Symposium on Medical Imaging 2007, San Diego, CA, USA*. The International Society for Optical Engineering (SPIE), feb 2007.

[108] H. Ólafsdóttir, T.A. Darvann, N.V. Hermann, E. Oubel, B.K. Ersbøll, A.F. Frangi, P. Larsen, C.A. Perlyn, G.M. Morriss-Kay, and S. Kreiborg. Computational mouse atlases and their application to automatic assessment of craniofacial dysmorphology caused by Crouzon syndrome. *Journal of Anatomy*, 2007 (submitted).

[109] M.R. Osborne, B. Presnell, and B.A. Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20(3):389–403, 2000.

[110] L. Pantoni, A.M. Basile, G. Pracucci, K. Asplund, J. Bogousslavsky, H. Chabriat, T. Erkinjuntti, F. Fazekas, J.M. Ferro, M. Hennerici, J. O'brien, P. Scheltens, M.C. Visser, L.O. Wahlund, G. Waldemar, A. Wallin, and D. Inzitari. Impact of age-related cerebral white matter changes on the transition to disability - the ladis study: Rationale, design and methodology. *Neuroepidemiology*, 24(1-2):51–62, 2005.

[111] L. Pantoni, A.M. Basile, G. Pracucci, K. Asplund, J. Bogousslavsky, H. Chabriat, T. Erkinjuntti, F. Fazekas, J.M. Ferro, M. Hennerici, J. O'Brien, P. Scheltens, M.C. Visser, L.O. Wahlund, G. Waldemar, A. Wallin, and D. Inzitari. Impact of age-related cerebral white matter changes on the transition to disability - the LADIS study: Rationale, design and methodology. *Neuroepidemiology*, 24(1-2):51–62, 2005.

[112] M.Y. Park and T. Hastie. L1 regularization path algorithm for generalized linear models. Technical report, Stanford University, 2006.

[113] C.A. Perlyn, V.B. DeLeon, C. Babbs, D. Govier, L. Burell, T. Darvann, S. Kreiborg, and G. Morriss-Kay. The craniofacial phenotype of the Crouzon mouse: Analysis of a model for syndromic craniosynostosis using 3D MicroCT. *Cleft Palate Craniofacial Journal*, 43(6):740–747, 2006.

[114] Bradley S Peterson, Patricia A Feineigle, Lawrence H Staib, and John C Gore. Automated measurement of latent morphological features in the human corpus callosum published online 21 february 2001. *Human Brain Mapping*, 12(4):232–245, 2001.

[115] J.C. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in kernel methods: support vector learning*, pages 185–208, Cambridge, MA, USA, 1999. MIT Press.

[116] J.O. Ramsay and B.W. Silverman. *Functional Data Analysis*. Springer Verlag, 1997.

[117] W. Reardon, R.M. Winter, P. Rutland, L.J. Pulleyn, B.M. Jones, and S. Malcolm. Mutations in the fibroblast growth factor receptor 2 gene cause Crouzon syndrome. *Nat Genet*, 8:98–103, 1994.

[118] L.-M. Reissell. Wavelet multiresolution representation of curves and surfaces. *Graphical Models and Image Processing*, 58(3):198–217, 1996.

[119] B.D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[120] S. Romdhani, S. Gong, and A. Psarrou. A multi-view nonlinear active shape model using kernel PCA. *BMVC99. Proceedings of the 10th British Machine Vision Conference*, pages 483–92 vol.2, 1999.

[121] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[122] S. Rosset and J. Zhu. Piecewise linear regularized solution paths. *Annals of Statistics (to appear)*, 2006.

[123] V. Rousson and T. Gasser. Simple component analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 53(4):539–555, 2004.

[124] D. Rueckert, L.I. Sonoda, C. Hayes, D.L.G. Hill, M.O. Leach, and D.J. Hawkes. Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Trans. on Medical Imaging*, 18(8):712–721, 1999.

[125] D. Rueckert, A.F. Frangi, and J.A. Schnabel. Automatic construction of 3D statistical deformation models of the brain using nonrigid registration. *IEEE Trans. on Medical Imaging*, 22(8):1014–1025, 2003.

[126] C. Ryberg, E. Rostrup, M.B. Stegmann, F. Barkhof, P. Scheltens, E.C.W. van Straaten, F. Fazekas, R. Schmidt, J.M. Ferro, H. Baezner, T. Erkinjuntti, H. Jokinen, L. Wahlund, J. O'Brien, A.M. Basile, L. Pantoni, D. Inzitari, and G. Waldemar. Clinical significance of corpus callosum atrophy in a mixed elderly population. *Neurobiology of Aging*, 2006.

[127] J.A. Schnabel, D. Rueckert, M. Quist, J.M. Blackall, A.D. Castellano-Smith, T. Hartkens, G.P. Penney, W.A. Hall, H. Liu, C.L. Truwit, F.A. Gerritsen, D.L.G. Hill, and D.J. Hawkes. A generic framework for non-rigid registration based on non-uniform multi-level free-form deformations. *Fourth Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI '01)*, 2208:573–581, 2001.

[128] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 2001.

[129] G. Schwartz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1979.

[130] J. Seo and H. Ko. Face detection using support vector domain description in color images. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004, ICASSP '04.*, volume 5, pages 729–732, 2004.

[131] X. Shen and J. Ye. Adaptive model selection. *Journal of the American Statistical Association*, 97(457):210–221, 2002.

[132] R.J. Sherin. A matrix formulation of Kaiser's varimax criterion. *Psychometrika*, 31(4): 535–538, 1966.

[133] K. Sjöstrand, T.E. Lund, K.H. Madsen, and R. Larsen. Sparse PCA, a new method for unsupervised analyses of fmri data. In *Proc. International Society of Magnetic Resonance In Medicine - ISMRM 2006, Seattle, Washington, USA*, Berkeley, CA, USA, may 2006. ISMRM.

[134] K. Sjöstrand, M.B. Stegmann, and R. Larsen. Sparse principal component analysis in medical shape modeling. In *International Symposium on Medical Imaging 2006, San Diego, CA, USA*, volume 6144. The International Society for Optical Engineering (SPIE), feb 2006.

[135] P.D. Sozou, T.F. Cootes, C.J. Taylor, and E.C. Di Mauro. Non-linear generalization of point distribution models using polynomial regression. *Image and Vision Computing*, 13(5):451–7, 1995.

[136] L.H. Staib and J.S. Duncan. Boundary finding with parametrically deformable models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(11):1061–1075, 1992.

[137] M.B. Stegmann. Analysis and segmentation of face images using point annotations and linear subspace techniques. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, aug 2002.

[138] M.B. Stegmann. *Generative Interpretation of Medical Images*. PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, 2004.

[139] M.B. Stegmann and K. Sjöstrand. Sparse modeling of landmark and texture variability using the orthomax criterion. International Symposium on Medical Imaging 2006, San Diego, CA, feb 2006.

[140] M.B. Stegmann, B.K. Ersbøll, and R. Larsen. FAME - a flexible appearance modelling environment. *IEEE Transactions on Medical Imaging*, 22(10):1319–1331, 2003.

[141] M.B. Stegmann, B.K. Ersbøll, and R. Larsen. FAME – a flexible appearance modelling environment. *IEEE Trans. on Medical Imaging*, 22(10):1319–1331, 2003.

[142] M.B. Stegmann, R.H. Davies, and C. Ryberg. Corpus callosum analysis using MDL-based sequential models of shape and appearance. In *International Symposium on Medical Imaging 2004, San Diego CA, SPIE*, pages 612–619, feb 2004.

[143] M.B. Stegmann, H. Olafsdottir, and H.B.W. Larsson. Unsupervised motion-compensation of multi-slice cardiac perfusion MRI. *Medical Image Analysis*, 9(4):394–410, 2005.

[144] M.B. Stegmann, K. Skoglund, and C. Ryberg. Mid-sagittal plane and mid-sagittal surface optimization in brain MRI using a local symmetry measure. In *International Symposium on Medical Imaging 2005, San Diego, CA, Proc. of SPIE vol. 5747*. SPIE, feb 2005.

[145] M.B. Stegmann, K. Skoglund, and C. Ryberg. Mid-sagittal plane and mid-sagittal surface optimization in brain MRI using a local symmetry measure. In *International Symposium on Medical Imaging 2005, San Diego, CA, Proc. of SPIE vol. 5747*, feb 2005.

[146] J. Stoeckel and G. Fung. Svm feature selection for classification of spect images of alzheimer's disease using spatial information. *Data Mining, Fifth IEEE International Conference on*, page 8 pp., 2005.

[147] C. Studholme, V. Cardenas, R. Blumenfeld, N. Schuff, H.J. Rosen, B. Miller, and M. Weiner. Deformation tensor morphometry of semantic dementia with quantitative validation. *NeuroImage*, 21(4):1387–1398, 2004.

[148] A. Suinesiaputra, A.F. Frangi, M. Üzümcü, J.H.C. Reiber, and B.P.F. Lelieveldt. Extraction of myocardial contractility patterns from short-axes MR images using independent component analysis. In *Lecture Notes in Computer Science*, volume 3117, pages 75–86. Springer-Verlag, 2004.

[149] A. Suinesiaputra, M. Üzümcü, A.F. Frangi, T.A.M. Kaandorp, J.H.C. Reiber, and B.P.F. Lelieveldt. Detecting regional abnormal cardiac contraction in short-axis MR images using independent component analysis. In *Lecture Notes in Computer Science*, volume 3216, pages 737–744. Springer-Verlag, 2004.

[150] P. Switzer. Min/max autocorrelation factors for multivariate spatial imagery. In L. Billard, editor, *Computer Science and Statistics*, pages 13–16. Elsevier Science Publishers B.V., 1985.

[151] D.M.J. Tax and R.P.W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20(11-13):1191–1199, 1999.

[152] D.M.J. Tax and R.P.W. Duin. Support vector data description. *Machine Learning*, 54 (1):45–66, 2004.

[153] D.M.J. Tax, A. Ypma, and R.P.W. Duin. Pump failure detection using support vector data descriptions. In D.J. Hand, J.N. Kok, and M.R. Berthold, editors, *Third International Symposium on Advances in Intelligent Data Analysis, IDA-99.*, volume 1642 of *Lecture Notes in Computer Science*, pages 415–25, 1999.

[154] J.M.F. ten Berge. Orthogonal procrustes rotation for two or more matrices. *Psychometrika*, 42:267–276, 1977.

[155] J.M.F. ten Berge. A joint treatment of varimax rotation and the problem of diagonalizing symmetric matrices simultaneously in the least-squares sense. *Psychometrika*, 49(3): 347–358, 1984.

[156] J.M.F. ten Berge, D.L. Knol, and H.A.L. Kiers. A treatment of the orthomax rotation family in terms of diagonalization, and a re-examination of a singular value approach to varimax rotation. *Computational Statistics Quarterly*, 3:207–217, 1988.

[157] R Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society - Series B Methodological*, 58(1):267–288, 1996.

[158] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[159] E.C.C. Tsang, T. Wong, P. Heng, D.S. Yeung, L. Shi, and D. Wang. Support vector clustering for brain activation detection. *Lecture Notes in Computer Science*, 3749: 572–579, 2005.

[160] C.J. Twining and C.J. Taylor. Kernel principal component analysis and the construction of non-linear active shape models. In *Proceedings of the British Machine Vision Conference, BMVC*, volume 1, pages 23–32, 2001.

[161] M. Üzümcü, A.F. Frangi, J. Reiber, and B. Lelieveldt. The use of independent component analysis in statistical shape models. In J.M. Fitzpatrick M. Sonka, editor, *International Symposium on Medical Imaging 2006, San Diego, CA, USA*, volume 5032. The International Society for Optical Engineering (SPIE), 2003.

[162] M. Üzümcü, A.F. Frangi, M. Sonka, J.H.C. Reiber, and B.P.F. Lelieveldt. ICA vs. PCA active appearance models: Application to cardiac MR segmentation. In *MICCAI 2003, Montréal, Canada*, volume 2878 of *LNCS*, pages 451–458, 2003.

[163] B. van Ginneken, M.B. Stegmann, and M. Loog. Segmentation of anatomical structures in chest radiographs using supervised methods: A comparative study on a public database. *Medical Image Analysis*, 2005.

[164] R.J. Vanderbei. LOQO: An interior point code for quadratic programming. *Optimization Methods and Software*, 11:451–484, 1999.

[165] V.N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.

[166] S.K. Vines. Simple principal components. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 49(4):441–451, 2000.

[167] S.F. Witelson. Hand and sex differences in the isthmus and genu of the human corpus callosum: A postmortem morphological study. *Brain*, 112(799-835), 1989.

[168] Z. Xue, D. Shen, and C. Davatzikos. Statistical representation of high-dimensional deformation fields with application to statistically constrained 3D warping. *Medical Image Analysis*, 10(5):740–751, 2006.

[169] M. Yuan and Y. Lin. On the non-negative garrotte estimator. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):143–161, 2007.

[170] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B*, 68(1):49–67, 2006.

[171] P. Yushkevich, S. Joshi, S.M. Pizer, J.G. Csernansky, and L.E. Wang. Feature selection for shape-based classification of biological objects. In C. Taylor and J.A. Noble, editors, *Information Processing in Medical Imaging (IPMI)*, pages 114–125, July 2003.

[172] E. Zaidel and M Iacoboni. *The Parallel Brain: The Cognitive Neuroscience of the Corpus Callosum.* Issues in Clinical and Cognitive Neuropsychology. Bradford Books, Jan 2003.

[173] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. L1-norm support vector machines. In L.K. Saul S. Thrun and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, June 2004.

[174] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

[175] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society - Series B Methodological*, 67(2):301–320, 2005.

[176] H. Zou, T. Hastie, and R. Tibshirani. On the degrees of freedom of the LASSO. Technical report, Stanford University, 2004.

[177] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265, 2006.