A PENALIZED MATRIX DECOMPOSITION, AND ITS APPLICATIONS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF STATISTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Daniela M. Witten

June 2010

© 2010 by Daniela Mottel Witten. All Rights Reserved.
Re-distributed by Stanford University under license with the author.

This work is licensed under a Creative Commons Attribution-Noncommercial 3.0 United States License.
http://creativecommons.org/licenses/by-nc/3.0/us/

This dissertation is online at: http://purl.stanford.edu/fw911jf5800

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Robert Tibshirani, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Balakanapathy Rajaratnam

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Jonathan Taylor

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

We present a penalized matrix decomposition, a new framework for computing a low-rank approximation for a matrix. This low-rank approximation is a generalization of the singular value decomposition. While the singular value decomposition usually yields singular vectors that have no elements that are exactly equal to zero, our new decomposition results in sparse singular vectors. This decomposition has a number of applications. When it is applied to a data matrix, it can yield interpretable results. One can apply it to a covariance matrix in order to obtain a new method for sparse principal components, and one can apply it to a crossproducts matrix in order to obtain a new method for sparse canonical correlation analysis. Moreover, when applied to a dissimilarity matrix, this leads to a method for sparse hierarchical clustering, which allows for the clustering of a set of observations using an adaptively chosen subset of the features. Finally, if this decomposition is applied to a between-class covariance matrix then it yields penalized linear discriminant analysis, an extension of Fisher's linear discriminant analysis to the high-dimensional setting.

Acknowledgements

This work would not have been possible without the help of many people. I would like to thank

- My adviser, Rob Tibshirani, for endless encouragement, countless good ideas, and for being a great friend;
- Trevor Hastie, for his contributions as a coauthor on part of this work as well as for excellent advice at many group meetings;
- Art Owen, Bala Rajaratnam, and Jonathan Taylor for serving on my thesis committee and for helpful feedback at various points;
- My husband, Ari, for being incredibly supportive;
- My parents and siblings for their help along the way;
- And the entire Department of Statistics for providing a home away from home and an intellectually stimulating atmosphere during my time as a graduate student.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Large-scale data in modern statistics

Throughout most of its history, the field of statistics has predominantly been concerned with the classical setting in which the number of observations exceeds the number of features, and the number of features is small or moderate. In the past fifteen years, things have changed. New technologies have resulted in the generation of massive data sets, and increased computational power has made possible their analysis. Large data sets are now commonplace in a variety of fields. Some examples are as follows:

1. Online advertisers are interested in characterizing a very large set of customers on the basis of many online behaviors in order to identify sets of customers who tend to buy particular products. Then ads can be targeted at specific individuals who are most likely to be interested in them. In this setting, the number of observations (individuals) is extremely large, and the number of features (online behaviors) may be quite large as well.

2. Collaborative filtering involves predicting a user's interest in a given item on the basis of a large data set consisting of that user's interests in other items, as well as other

users' interests in a set of items. The well known Netflix problem is an example of a large-scale collaborative filtering data set in which the number of observations (movie viewers) and the number of features (movies) is extremely large.

3. In the field of genomics, the microarray has become an important tool for characterizing tissue samples. In a single experiment, one can measure the expression levels of tens of thousands of genes. On the basis of these data, a researcher might want to identify latent factors, discover novel subgroups among the observations, or develop a tool to automatically classify the tissue samples into known groups. In microarray data, the number of observations (tissue samples) tends to be no more than dozens or hundreds, while the number of features (genes for which expression has been measured) is on the order of tens of thousands.

4. Functional magnetic resonance imaging (fMRI) is used to measure brain activity, or more specifically, changes in vascular oxygenation. Researchers collect fMRI measurements for a set of individuals, and attempt to answer questions such as whether activation of a particular brain region is associated with certain thoughts or behaviors. A study might contain dozens of observations (subjects) and a much larger number of features (voxels, or fMRI measurements).

These are just a few of the countless examples of large data sets that are of interest to statisticians. In the first two examples above, both the number of observations and the number of features are quite large. On the other hand, the last two examples are characterized by the fact that the number of features is very large whereas the number of observations is small or moderate. Both situations arise frequently in modern statistical applications.

In this dissertation, we are mostly concerned with examples in which the number of features greatly exceeds the number of observations. We refer to this setting as *high-dimensional*. Many classical statistical methods are not suitable in this setting. As a result,

in recent years, the statistical community has devoted a great deal of effort into developing methods for the analysis of data in the high-dimensional setting. In this chapter, we review some of the problems faced by classical statistical methods in high dimensions, as well as some of the approaches that are commonly used to address these problems. We then present an overview of the topics presented in this dissertation.

In the statistical learning literature, the term *supervised learning* is used to refer to regression, classification, and other approaches for predicting some outcome $\mathbf{y}$ on the basis of a data matrix $\mathbf{X}$. On the other hand, *unsupervised learning* refers to matrix decompositions, clustering algorithms, and other tools for finding signal in a data matrix $\mathbf{X}$ in the absence of any outcome. In Chapter 1.2, we discuss supervised methods in the high-dimensional setting, and in Chapter 1.3, we discuss unsupervised methods in the high-dimensional setting.

1.2 Supervised learning in high dimensions

In what follows, $\mathbf{X}$ denotes a $n \times p$ matrix with observations on the rows and features on the columns. For simplicity, we assume that the features have been standardized to have mean zero and standard deviation one. Most classical statistical methods are intended for the situation in which the number of observations n exceeds the number of features p, and p is small or moderate. When the situation is reversed and $p > n$, many statistical methods are not applicable. For instance, suppose that $\mathbf{y} \in \mathbb{R}^n$ is a vector of outcome measurements for each observation, and that we wish to predict the outcome for a new observation on the basis of its feature measurements. One could fit the simple linear model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{1.1}$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$ is a coefficient vector, and $\boldsymbol{\epsilon} \in \mathbb{R}^n$ is a noise vector. Ordinary least squares regression involves fitting the model (1.1) by minimizing the sum of squared errors

$$\underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\text{minimize}} \{||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2\}. \tag{1.2}$$

The problem (1.2) has solution

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{1.3}$$

However, linear regression encounters two problems when $p > n$:

1. The sample covariance matrix of the features is singular. That is, $\mathbf{X}^T\mathbf{X}$ cannot be inverted, and so one cannot calculate (1.3).

2. The fitted model is not interpretable because $\hat{\boldsymbol{\beta}}$ contains p nonzero elements, where p is quite large. There are a number of ways that regression could be made more interpretable, but most of the focus in the statistical literature has been on *sparsity*. We say that $\hat{\boldsymbol{\beta}}$ is *sparse* if many of its elements are exactly equal to zero. In this case, one can predict $\mathbf{y}$ using just a subset of the features, and it is easier to determine which features play a role in the model obtained.

We elaborate on the second point above. In the analysis of high-dimensional data, sparsity is valued for two reasons beyond just the interpretability of the resulting coefficient estimate:

1. *Parsimony.* In high dimensions one can often obtain a model using a small number of features that is as good as or better than, in terms of prediction error on an independent test set, one that involves all of the features. All else being equal, one would prefer a simpler model that states that only a subset of the features determines the outcome.

2. *Practical application.* In many applications, for a model to be practically useful it

must contain just a small number of features. For instance, suppose that one wishes to predict a quantitative outcome on the basis of a patient's gene expression measurements. In order for such a model to be clinically useful, it must involve only a small number of genes, since collecting tens of thousands of gene expression measurements on every patient is infeasible.

One well studied approach for extending (1.2) to the high-dimensional setting involves applying penalties to the elements of $\boldsymbol{\beta}$; this is known as *regularization* or *penalization*. For instance, one can apply an L_2 penalty to $\boldsymbol{\beta}$ in (1.2). This yields *ridge regression* (Hoerl & Kennard 1970),

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\text{minimize}}\{||\mathbf{y}-\mathbf{X}\boldsymbol{\beta}||^2+\lambda||\boldsymbol{\beta}||^2\}, \tag{1.4}$$

and has the effect of regularizing the sample covariance matrix of the features when the nonnegative tuning parameter λ is large. One can show that the solution to (1.4) is

$$\hat{\boldsymbol{\beta}}=(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}. \tag{1.5}$$

Ridge regression addresses the singularity issue that occurs when $p > n$, since $\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}$ has full rank when $\lambda > 0$. However, ridge regression has a major drawback: none of the elements of the solution (1.5) are exactly equal to zero, and so it is not easy to determine which features are important in the model obtained.

Tibshirani (1996) applied an L_1 or *lasso* penalty to the elements of $\boldsymbol{\beta}$ in the sum of squared errors criterion (1.2), leading to the optimization problem

$$\underset{\boldsymbol{\beta}}{\text{minimize}}\{||\mathbf{y}-\mathbf{X}\boldsymbol{\beta}||^2+\lambda||\boldsymbol{\beta}||_1\}. \tag{1.6}$$

The lasso (1.6) performs *feature selection* in an automated way. That is, when the tuning parameter λ is large, the resulting estimate $\hat{\boldsymbol{\beta}}$ is sparse. As a result, the lasso has a major advantage over ridge regression in terms of interpretability of the resulting model. The

problem is *convex*, so tools from convex optimization can be used to solve it (see e.g. Boyd & Vandenberghe 2004). Moreover, a number of specialized approaches to solving (1.6) have been proposed in the statistical literature (see e.g. Efron et al. 2004, Friedman et al. 2007). It turns out that if $\mathbf{X} = \mathbf{I}$, the problem is particularly simple, since

$$\underset{\boldsymbol{\beta}}{\text{minimize}}\{||\mathbf{y} - \boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_1\} \tag{1.7}$$

has the solution $\hat{\beta}_j = S(y_j, \frac{\lambda}{2})$ where S is the *soft-thresholding* operator, defined as

$$S(a, c) = \text{sgn}(a)\max(0, |a| - c). \tag{1.8}$$

The soft-thresholding operator arises repeatedly throughout this dissertation. When applied to a vector, the operation is performed componentwise.

A number of other proposals that address the singularity and interpretability problems of linear regression via a regularization approach take the more general form

$$\underset{\boldsymbol{\beta}}{\text{minimize}}\{||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + P(\boldsymbol{\beta})\}, \tag{1.9}$$

where $P(\boldsymbol{\beta})$ is some penalty on the elements of $\boldsymbol{\beta}$, often chosen so that the resulting coefficient estimate is sparse. Some examples include the elastic net penalty of Zou & Hastie (2005), the grouped lasso penalty of Yuan & Lin (2007), the adaptive lasso penalty of Zou (2006), and the concave penalties considered in Fan & Li (2001) and Lv & Fan (2009). In particular, the *fused lasso* penalty (Tibshirani et al. 2005)

$$P(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} |\beta_j| + \delta \sum_{j=2}^{p} |\beta_j - \beta_{j-1}| \tag{1.10}$$

arises repeatedly in the coming chapters. Here, λ and δ are nonnegative tuning parameters. When λ is large then the solution will be sparse, and when δ is large it will be piecewise

constant. The fused lasso penalty is appropriate if there is a linear ordering to the features along which smoothness is expected.

Just as linear regression suffers from a singularity problem and an interpretability problem in high dimensions, these two problems also arise in applying standard classification methods such as logistic regression and linear discriminant analysis. In the classification setting, a categorical outcome vector $\mathbf{y} \in \{1, 2, \ldots, K\}^n$ is available, and we wish to assign a new observation to one of K classes on the basis of its feature measurements. A number of authors have used regularization and penalization approaches to extend classification methods to the high-dimensional setting (see e.g. Tibshirani et al. 2002, Tibshirani et al. 2003, Guo et al. 2007, Park & Hastie 2007, Grosenick et al. 2008, Leng 2008, Friedman et al. 2010, Clemmensen et al. 2010).

It is worth noting that regularization approaches for regression and classification are often applicable even when $n > p$. A regularized model frequently yields lower test set error rates than an unregularized model, since regularization can result in decreased variance. Moreover, even in low dimensions, feature selection can be a desirable property.

1.3 Unsupervised learning in high dimensions

As mentioned in the previous section, much effort has been devoted to extending supervised methods to the high-dimensional setting. These proposals have been studied extensively from a theoretical and applied viewpoint. However, far less energy has been spent in developing and understanding unsupervised tools for $p > n$. There are two major reasons for this.

1. When $p > n$, many classical supervised methods fail in an obvious way - for instance, linear regression fails since the sample covariance matrix of the features is singular - whereas unsupervised methods encounter a problem that is more subtle. Many unsupervised tools such as matrix decompositions and clustering methods technically

can be applied in high dimensions, but the results may suffer from high variance and a lack of interpretability. For instance, suppose one wishes to approximate the matrix $\mathbf{X}$ as $\mathbf{X} \approx \mathbf{AB}$ where $\mathbf{A}$ is $n \times q$, $\mathbf{B}$ is $q \times p$, and $q < n, p$. One could do this using the truncated singular value decomposition. Then $\mathbf{A}$ and $\mathbf{B}$ each will contain nq and pq nonzero elements; as a result, they can be quite difficult to interpret when p and possibly n is very large.

2. Supervised methods have the attractive property that their performance is easily assessed, for instance by computing the error rate on an independent test set. On the other hand, it is much more difficult to assess the performance of unsupervised methods, since there is no outcome vector that serves as a "gold standard" against which the results can be measured.

In this dissertation, we develop a number of methods for unsupervised learning in the high-dimensional setting. We use a regularization approach to modify classical unsupervised methods that are intended for low-dimensional problems.

1.4 The penalized matrix decomposition for $p > n$

In this dissertation, we consider a number of classical statistical tools,

1. the singular value decomposition,
2. principal components analysis,
3. canonical correlation analysis,
4. clustering, and
5. linear discriminant analysis.

We begin by discussing how each tool can fail in the high-dimensional setting, and we then propose extensions that are intended to overcome these failures. While each individual topic

plays an important role in the field of statistics, at first glance this set of topics may seem somewhat unrelated. However, it turns out that the methods proposed in this paper share a common theme, in that they all result from the application of a simple extension of the singular value decomposition, called the *penalized matrix decomposition*.

The penalized matrix decomposition, which we introduce in Chapter 2, can be used to decompose a $n \times p$ matrix $\mathbf{X}$ as $\mathbf{X} \approx \mathbf{AB}$ in such a way that most of the elements of the $n \times q$ matrix $\mathbf{A}$ and the $q \times p$ matrix $\mathbf{B}$ are exactly equal to zero. When this method is applied to a data matrix, it can yield interpretable results.

In addition, useful results can be obtained by applying variants of the penalized matrix decomposition to other types of matrices.

1. If the penalized matrix decomposition is applied to a covariance matrix then a method for *sparse principal components analysis* results. This is discussed in Chapter 3.

2. Suppose that $\mathbf{X}_1$ and $\mathbf{X}_2$ are $n \times p_1$ and $n \times p_2$ matrices with p_1 and p_2 measurements on a single set of n observations, and that the columns of $\mathbf{X}_1$ and $\mathbf{X}_2$ are standardized to have mean 0 and standard deviation 1. Then if the penalized matrix decomposition is applied to the matrix $\mathbf{X}_1^T\mathbf{X}_2$, a method for *penalized canonical correlation analysis* results. This can be used to find an interpretable linear combination of the variables in the first data set that is correlated with an interpretable linear combination of variables in the second data set. We propose penalized canonical correlation analysis in Chapter 4.

3. Let $\mathbf{D}$ denote a $n^2 \times p$ matrix for which the (ii', j) element is the dissimilarity between X_{ij} and $X_{i'j}$. Then if the penalized matrix decomposition is applied to $\mathbf{D}$, a reweighted dissimilarity matrix will be obtained. This dissimilarity matrix can be used to perform *sparse clustering*, a clustering of the observations using a subset of the features that is appropriate for the high-dimensional setting. This is discussed in Chapter 5.

4. Consider the classification setting, in which each of n observations $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ belongs to one of K classes, and one wishes to classify a new observation on the basis of its feature measurements. Then, one can apply the penalized matrix decomposition to the between-class covariance matrix in order to obtain a method for *penalized linear discriminant analysis*. We consider this case in Chapter 6.

The Discussion is in Chapter 7.

1.5 A short note on biconvexity

In the chapters that follow, we repeatedly make use of *biconvexity*. That is, consider the optimization problem

$$\begin{aligned} &\underset{\mathbf{u},\mathbf{v}}{\text{minimize}}\{f(\mathbf{u},\mathbf{v})\} \\ &\text{subject to } g_i(\mathbf{u}) \leq 0, i = 1, \ldots, I,\ h_j(\mathbf{v}) \leq 0, j = 1, \ldots, J, \end{aligned} \tag{1.11}$$

where the functions g_i and h_j are convex, and where the function f is convex in $\mathbf{u}$ and is also convex in $\mathbf{v}$, but not necessarily *jointly* convex in $\mathbf{u}$ and $\mathbf{v}$. An example of such a function f is $f(\mathbf{u},\mathbf{v}) = \mathbf{u}^T\mathbf{v}$. One cannot optimize (1.11) directly using tools from convex optimization, because the problem is not convex. But since with $\mathbf{u}$ held fixed, (1.11) is convex in $\mathbf{v}$, and with $\mathbf{v}$ held fixed, it is convex in $\mathbf{u}$, a simple iterative algorithm is guaranteed to decrease the objective at each step:

Algorithm 1.1: An iterative algorithm for biconvex problems

1. Initialize $\mathbf{v}$ to satisfy the constraints.

2. Iterate:

(a) Update $\mathbf{u}$ to be the solution to

$$\begin{aligned} &\underset{\mathbf{u}}{\text{minimize}}\{f(\mathbf{u}, \mathbf{v})\} \\ &\text{subject to } g_i(\mathbf{u}) \leq 0, i = 1, \ldots, I. \end{aligned} \tag{1.12}$$

(b) Update $\mathbf{v}$ to be the solution to

$$\begin{aligned} &\underset{\mathbf{v}}{\text{minimize}}\{f(\mathbf{u}, \mathbf{v})\} \\ &\text{subject to } h_j(\mathbf{v}) \leq 0, j = 1, \ldots, J. \end{aligned} \tag{1.13}$$

Though the objective decreases monotonically, this iterative approach does not yield the global optimum in general. Moreover, the solution obtained depends on the initial value for $\mathbf{v}$ in Step 1. Despite its failure to yield the global optimum, Algorithm 1.1 remains an attractive option for biconvex optimization since in general, no convenient algorithms exist for finding the global solution. We refer the interested reader to Gorski et al. (2007) for an overview of biconvex problems and the theoretical properties of Algorithm 1.1.

Though Algorithm 1.1 does not yield the global solution to the problem (1.11), in many cases it is quite efficient and leads to a useful result. We repeatedly make use of such an iterative approach when faced with biconvex problems in the coming chapters. Throughout this dissertation, we will refer to Algorithm 1.1 and related approaches as tools for "solving" biconvex problems or "minimizing" biconvex functions. We ask the reader to bear in mind that we are using this terminology loosely, since we do not expect the global solution to be obtained.

1.6 Contribution to this work

All of the research described in this dissertation was primarily performed by the author, with contributions from Robert Tibshirani. Trevor Hastie also contributed to Chapters 2 and 3. Chapters 2, 3, 4, and 5 appear elsewhere in published form (Witten et al. 2009, Witten & Tibshirani 2009, Witten & Tibshirani 2010).

Chapter 2

The penalized matrix decomposition

In this chapter, we propose a penalized matrix decomposition. This work has been published in Witten et al. (2009).

2.1 General form of the penalized matrix decomposition

Let $\mathbf{X}$ denote an $n \times p$ matrix of data with rank $K \leq \min(n, p)$. Without loss of generality, assume that the overall mean of $\mathbf{X}$ is zero. The *singular value decomposition* (SVD) is a well known tool for decomposing a matrix (see e.g. Mardia et al. 1979). It can be written as follows:

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T, \mathbf{U}^T\mathbf{U} = \mathbf{I}_K, \mathbf{V}^T\mathbf{V} = \mathbf{I}_K, d_1 \geq d_2 \geq \ldots \geq d_K > 0. \tag{2.1}$$

Let $\mathbf{u}_k$ denote column k of the $n \times K$ matrix $\mathbf{U}$, let $\mathbf{v}_k$ denote column k of the $p \times K$ matrix $\mathbf{V}$, and note that d_k denotes the kth diagonal element of the $K \times K$ diagonal matrix $\mathbf{D}$. The SVD has a number of attractive properties, one of which is that for any $r \leq K$, the

problem

$$\underset{\mathbf{A}\in\mathbb{R}^{n\times p}}{\text{minimize}}\{||\mathbf{X}-\mathbf{A}||_F^2\} \text{ subject to } \text{rank}(\mathbf{A})=r \tag{2.2}$$

has solution $\sum_{k=1}^{r} d_k\mathbf{u}_k\mathbf{v}_k^T$, where $||\cdot||_F^2$ indicates the squared Frobenius norm (Eckart & Young 1936). In other words, the first r components of the SVD give the best rank-r approximation to a matrix, in the sense of the Frobenius norm.

The singular vectors obtained using the SVD will in general have no elements that are exactly equal to zero. However, if $\mathbf{X}$ contains many rows and/or columns, and if the goal is to obtain an interpretable low-rank approximation for $\mathbf{X}$, then the fact that the singular vectors obtained using the SVD are not sparse can be problematic. From an interpretation standpoint, one might prefer to obtain a low-rank approximation for $\mathbf{X}$ that has Frobenius error almost as low as the SVD (2.2), but that is composed of sparse singular vectors. In this chapter, we seek such an interpretable decomposition. We develop a generalization of the SVD by imposing additional constraints on the elements of $\mathbf{U}$ and $\mathbf{V}$. If the constraint function is chosen appropriately, then the resulting singular vectors will be sparser than those obtained using the SVD.

We start with a rank-one approximation. Consider the following optimization problem, which results from imposing constraints on the elements of $\mathbf{u}$ and $\mathbf{v}$ in the rank-1 criterion for the SVD (2.2):

$$\begin{aligned}&\underset{d,\mathbf{u},\mathbf{v}}{\text{minimize}}\{\frac{1}{2}||\mathbf{X}-d\mathbf{u}\mathbf{v}^T||_F^2\}\\ &\text{subject to } ||\mathbf{u}||^2=1, ||\mathbf{v}||^2=1, P_1(\mathbf{u})\le c_1, P_2(\mathbf{v})\le c_2, d\ge 0.\end{aligned} \tag{2.3}$$

Here, P_1 and P_2 are convex penalty functions, which can take on a variety of forms. Useful examples are

- lasso: $P_1(\mathbf{u}) = \sum_{i=1}^{n} |u_i|$, and
- fused lasso: $P_1(\mathbf{u}) = \sum_{i=1}^{n} |u_i| + \sum_{i=2}^{n} |u_i - u_{i-1}|$.

Only certain ranges of c_1 and c_2 will lead to feasible solutions, as discussed in Chapter 2.3. We now derive a more convenient form for this criterion.

The following decomposition holds:

Proposition 2.1.1. *Let* $\mathbf{U}$ *and* $\mathbf{V}$ *be* $n \times K$ *and* $p \times K$ *orthogonal matrices, and* $\mathbf{D}$ *a diagonal matrix with diagonal elements* d_k*. Then,*

$$\frac{1}{2}||\mathbf{X} - \mathbf{UDV}^T||_F^2 = \frac{1}{2}||\mathbf{X}||_F^2 - \sum_{k=1}^{K} \mathbf{u}_k^T \mathbf{X} \mathbf{v}_k d_k + \frac{1}{2}\sum_{k=1}^{K} d_k^2 \tag{2.4}$$

The proposition's proof is given in Chapter 2.7. Hence, using the case $K = 1$, we have that the values of $\mathbf{u}$ and $\mathbf{v}$ that solve (2.3) also solve the following problem:

$$\begin{aligned} &\underset{\mathbf{u},\mathbf{v}}{\text{maximize}}\{\mathbf{u}^T\mathbf{X}\mathbf{v}\} \\ &\text{subject to } ||\mathbf{u}||^2 = 1, ||\mathbf{v}||^2 = 1, P_1(\mathbf{u}) \leq c_1, P_2(\mathbf{v}) \leq c_2, \end{aligned} \tag{2.5}$$

and the value of d solving (2.3) is $\mathbf{u}^T\mathbf{X}\mathbf{v}$. The objective function $\mathbf{u}^T\mathbf{X}\mathbf{v}$ in (2.5) is bilinear in $\mathbf{u}$ and $\mathbf{v}$: that is, with $\mathbf{u}$ fixed, it is linear in $\mathbf{v}$, and vice-versa. In fact, with $\mathbf{v}$ fixed, problem (2.5) takes the following form:

$$\underset{\mathbf{u}}{\text{maximize}}\{\mathbf{u}^T\mathbf{X}\mathbf{v}\} \text{ subject to } P_1(\mathbf{u}) \leq c_1, ||\mathbf{u}||^2 = 1. \tag{2.6}$$

This problem is *not* convex, due to the L_2 equality constraint on $\mathbf{u}$.

We can finesse this as follows. We define the rank-one *penalized matrix decomposition* (PMD) as

$$\begin{aligned} &\underset{\mathbf{u},\mathbf{v}}{\text{maximize}}\{\mathbf{u}^T\mathbf{X}\mathbf{v}\} \\ &\text{subject to } ||\mathbf{u}||^2 \leq 1, ||\mathbf{v}||^2 \leq 1, P_1(\mathbf{u}) \leq c_1, P_2(\mathbf{v}) \leq c_2. \end{aligned} \tag{2.7}$$

With $\mathbf{v}$ fixed, this criterion takes the form

$$\underset{\mathbf{u}}{\text{maximize}}\{\mathbf{u}^T\mathbf{X}\mathbf{v}\} \text{ subject to } P_1(\mathbf{u}) \leq c_1, ||\mathbf{u}||^2 \leq 1, \tag{2.8}$$

which is convex. This means that (2.7) is biconvex, and this suggests an iterative algorithm for optimizing it. Moreover, it turns out that the solution to (2.8) also satisfies $||\mathbf{u}||^2 = 1$, provided that c_1 is chosen so that (for fixed $\mathbf{v}$) the solution to

$$\underset{\mathbf{u}}{\text{maximize}}\{\mathbf{u}^T\mathbf{X}\mathbf{v}\} \text{ subject to } P_1(\mathbf{u}) \leq c_1 \tag{2.9}$$

has L_2 norm greater than or equal to 1. This follows from the Karush-Kuhn-Tucker (KKT) conditions in convex optimization (see e.g. Boyd & Vandenberghe, 2004). Therefore, for c_1 chosen appropriately, the solution to (2.8) solves (2.6).

The following iterative algorithm is used to optimize the criterion for the rank-one PMD:

Algorithm 2.1: Computation of single factor PMD model

1. Initialize $\mathbf{v}$ to have L_2 norm 1.
2. Iterate:
   (a) Let $\mathbf{u}$ be the solution to
   $$\underset{\mathbf{u}}{\text{maximize}}\{\mathbf{u}^T\mathbf{X}\mathbf{v}\} \text{ subject to } P_1(\mathbf{u}) \leq c_1, ||\mathbf{u}||^2 \leq 1. \tag{2.10}$$
   (b) Let $\mathbf{v}$ be the solution to
   $$\underset{\mathbf{v}}{\text{maximize}}\{\mathbf{u}^T\mathbf{X}\mathbf{v}\} \text{ subject to } P_2(\mathbf{v}) \leq c_2, ||\mathbf{v}||^2 \leq 1. \tag{2.11}$$
3. Set $d = \mathbf{u}^T\mathbf{X}\mathbf{v}$.

In Chapter 2.2, we present an algorithm for obtaining multiple-factor solutions for the PMD. When P_1 and P_2 both are L_1 penalties, maximizations in Steps 2(a) and 2(b) are simple. This is explained in Algorithm 2.3 in Chapter 2.3.

It can be seen that without the P_1 and P_2 constraints, the algorithm above leads to the usual rank-one SVD. Starting with $\mathbf{v}^{(0)}$, one can show that at the end of iteration i,

$$\mathbf{v}^{(i)} = \frac{(\mathbf{X}^T\mathbf{X})^i\mathbf{v}^{(0)}}{||(\mathbf{X}^T\mathbf{X})^i\mathbf{v}^{(0)}||_2}. \tag{2.12}$$

This is the well known "power method" for computing the largest eigenvector of $\mathbf{X}^T\mathbf{X}$, which is the leading singular vector of $\mathbf{X}$.

In practice, we suggest using the first right singular vector of $\mathbf{X}$ as the initial value $\mathbf{v}$. In general, Algorithm 2.1 does not converge to a global optimum for (2.7); however, our empirical studies indicate that the algorithm does result in interpretable factors for appropriate choices of the penalty terms. Note that each iteration of Step 2 in Algorithm 2.1 results in an increase in the objective in (2.7). More information on iterative algorithms for optimizing biconvex problems can be found in Gorski et al. (2007).

The PMD is related to a method of Shen & Huang (2008) for identifying sparse principal components; we will elaborate on the connection between the two methods in Chapter 3.

2.2 PMD for multiple factors

In order to obtain multiple factors of the PMD, we minimize the single factor criterion (2.7) repeatedly, each time using as the $\mathbf{X}$ matrix the residuals obtained by subtracting from the data matrix the previous factors found. The algorithm is as follows:

Algorithm 2.2: Computation of K factors of PMD

1. Let $\mathbf{X}^1 = \mathbf{X}$.

2. For $k = 1, \ldots, K$:

(a) Find $\mathbf{u}_k$, $\mathbf{v}_k$, and d_k by applying the single-factor PMD algorithm (Algorithm 2.1) to data $\mathbf{X}^k$.

(b) Let $\mathbf{X}^{k+1} = \mathbf{X}^k - d_k \mathbf{u}_k \mathbf{v}_k^T$.

Without the P_1 and P_2 constraints, it can be shown that the K-factor PMD algorithm leads to the rank-K SVD of $\mathbf{X}$. In particular, the successive solutions are orthogonal. This can be seen since the solutions $\mathbf{u}_k$ and $\mathbf{v}_k$ are in the column and row spaces of $\mathbf{X}^k$, which has been orthogonalized with respect to $\mathbf{u}_j$ and $\mathbf{v}_j$ for all $j < k$. With P_1 and/or P_2 present, the solutions are no longer in the column and row spaces in general, and so the orthogonality does not hold. In Chapter 3.3 we discuss an alternative multi-factor approach, in the setting where the PMD is specialized to sparse principal components.

2.3 Forms of PMD of special interest

We are most interested in two specific forms of the PMD, which we call the "PMD(L_1,L_1)" and "PMD(L_1,FL)" methods. The PMD(L_1,L_1) criterion is as follows:

$$\begin{aligned} &\underset{\mathbf{u},\mathbf{v}}{\text{maximize}}\{\mathbf{u}^T\mathbf{X}\mathbf{v}\} \\ &\text{subject to } ||\mathbf{u}||^2 \leq 1, ||\mathbf{v}||^2 \leq 1, ||\mathbf{u}||_1 \leq c_1, ||\mathbf{v}||_1 \leq c_2. \end{aligned} \tag{2.13}$$

This method results in factors $\mathbf{u}$ and $\mathbf{v}$ that are sparse for c_1 and c_2 chosen appropriately. As shown in Figure 2.1, we restrict c_1 and c_2 to the ranges $1 \leq c_1 \leq \sqrt{n}$ and $1 \leq c_2 \leq \sqrt{p}$. When $c_1 \leq 1$ only the L_1 constraint on $\mathbf{u}$ is active, and when $c_1 \geq \sqrt{n}$ only the L_2 constraint on $\mathbf{u}$ is active.

We have the following proposition, where S is the soft-thresholding operator (1.8):

Proposition 2.3.1. *Consider the optimization problem*

$$\underset{\mathbf{u}}{\text{maximize}}\{\mathbf{u}^T\mathbf{a}\} \text{ subject to } ||\mathbf{u}||^2 \leq 1, ||\mathbf{u}||_1 \leq c. \tag{2.14}$$

Assume that $\mathbf{a}$ *has a unique element with maximal absolute value. Then, the solution is* $\mathbf{u} = \frac{S(\mathbf{a},\Delta)}{||S(\mathbf{a},\Delta)||_2}$, *with* $\Delta = 0$ *if this results in* $||\mathbf{u}||_1 \leq c$; *otherwise,* $\Delta > 0$ *is chosen so that* $||\mathbf{u}||_1 = c$.

The proof is given in Chapter 2.7. We solve the PMD criterion in (2.13) using Algorithm 2.1, with Steps 2(a) and 2(b) adjusted as follows:

Algorithm 2.3: Computation of single factor PMD(L_1,L_1) model

1. Initialize $\mathbf{v}$ to have L_2 norm 1.
2. Iterate:
   (a) Let $\mathbf{u} = \frac{S(\mathbf{Xv},\Delta_1)}{||S(\mathbf{Xv},\Delta_1)||_2}$, where $\Delta_1 = 0$ if this results in $||\mathbf{u}||_1 \leq c_1$; otherwise, Δ_1 is chosen to be a positive constant such that $||\mathbf{u}||_1 = c_1$.
   (b) Let $\mathbf{v} = \frac{S(\mathbf{X}^T\mathbf{u},\Delta_2)}{||S(\mathbf{X}^T\mathbf{u},\Delta_2)||_2}$, where $\Delta_2 = 0$ if this results in $||\mathbf{v}||_1 \leq c_2$; otherwise, Δ_2 is chosen to be a positive constant such that $||\mathbf{v}||_1 = c_2$.
3. Let $d = \mathbf{u}^T\mathbf{Xv}$.

If one wishes for $\mathbf{u}$ and $\mathbf{v}$ to have approximately the same fraction of nonzero elements, then one can fix a constant $c < 1$, and set $c_1 = c\sqrt{n}$, $c_2 = c\sqrt{p}$. For each update of $\mathbf{u}$ and $\mathbf{v}$, Δ_1 and Δ_2 are chosen by a binary search.

Figure 2.1 shows a graphical representation of the L_1 and L_2 constraints on $\mathbf{u}$ that are present in the PMD(L_1,L_1) criterion: namely, $||\mathbf{u}||^2 \leq 1$ and $||\mathbf{u}||_1 \leq c_1$. From the figure, it is clear that in two dimensions, when both the L_1 and L_2 constraints are active, then both u_1 and u_2 are nonzero. However, when n, the dimension of $\mathbf{u}$, is at least three, then the right panel of Figure 2.1 can be thought of as the hyperplane $\{u_i = 0\ \forall i > 2\}$. In this case, the small circles indicate regions where both constraints are active and the solution is sparse (since $u_i = 0$ for $i > 2$).

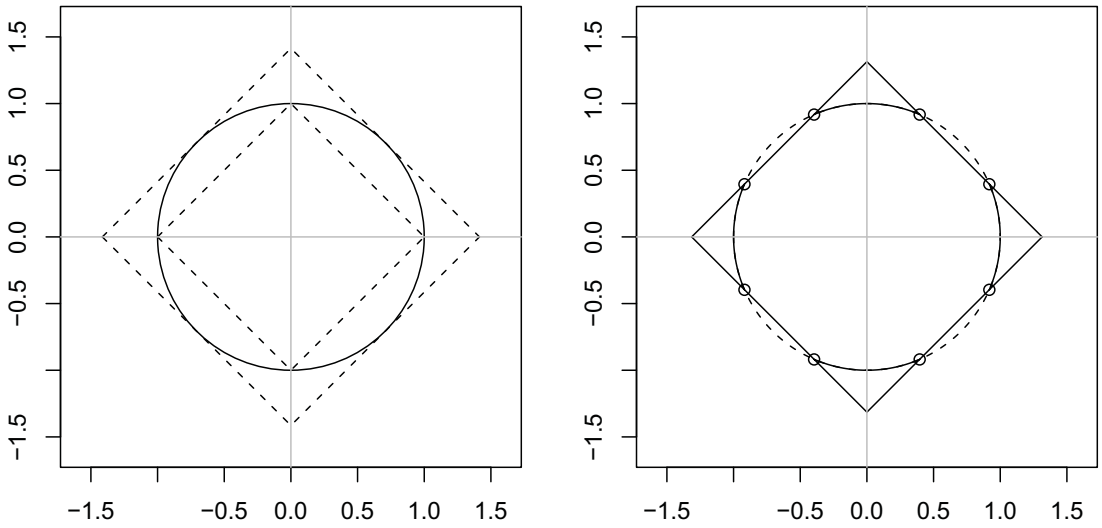

Figure 2.1: A graphical representation of the L_1 and L_2 constraints on $\mathbf{u} \in \mathbb{R}^2$ in the PMD(L_1,L_1) criterion. **Left:** The L_2 constraint is the solid circle. For both the L_1 and L_2 constraints to be active, c must be between 1 and $\sqrt{2}$. The constraints $||\mathbf{u}||_1 = 1$ and $||\mathbf{u}||_1 = \sqrt{2}$ are shown using dashed lines. **Right:** The L_1 and L_2 constraints on $\mathbf{u}$ are shown for some c between 1 and $\sqrt{2}$. Small circles indicate the points where both the L_1 and the L_2 constraints are active. The solid arcs indicate the solutions that occur when $\Delta_1 = 0$ in Algorithm 2.3.

The PMD(L_1,FL) criterion is as follows (where "FL" stands for the "fused lasso" penalty, proposed in Tibshirani et al. 2005):

$$\begin{aligned}&\underset{\mathbf{u},\mathbf{v}}{\text{maximize}}\{\mathbf{u}^T\mathbf{X}\mathbf{v}\}\\&\text{subject to } ||\mathbf{u}||^2 \leq 1, ||\mathbf{u}||_1 \leq c_1, ||\mathbf{v}||^2 \leq 1, \sum_{j=1}^{p} |v_j| + \lambda \sum_{j=2}^{p} |v_j - v_{j-1}| \leq c_2. \end{aligned} \tag{2.15}$$

When c_1 is small, then $\mathbf{u}$ will be sparse, and when c_2 is small, then $\mathbf{v}$ will be sparse. Moreover, when the tuning parameter $\lambda \geq 0$ is large, then $\mathbf{v}$ will also be piecewise constant. For simplicity, rather than solving (2.15), we solve a slightly different criterion that results from using the Lagrange form, rather than the bound form, of the constraints on $\mathbf{v}$:

$$\begin{aligned}&\underset{\mathbf{u},\mathbf{v}}{\text{minimize}}\{-\mathbf{u}^T\mathbf{X}\mathbf{v} + \frac{1}{2}\mathbf{v}^T\mathbf{v} + \lambda_1 \sum_{j=1}^{p} |v_j| + \lambda_2 \sum_{j=2}^{p} |v_j - v_{j-1}|\}\\&\text{subject to } ||\mathbf{u}||^2 \leq 1, ||\mathbf{u}||_1 \leq c. \end{aligned} \tag{2.16}$$

We can solve this by replacing Steps 2(a) and 2(b) in Algorithm 2.1 with the appropriate updates:

Algorithm 2.4: Computation of single factor PMD(L_1,FL) model

1. Initialize $\mathbf{v}$ to have L_2 norm 1.

2. Iterate:

   (a) If $\mathbf{v} = 0$, then $\mathbf{u} = 0$. Otherwise, let $\mathbf{u} = \frac{S(\mathbf{X}\mathbf{v},\Delta)}{||S(\mathbf{X}\mathbf{v},\Delta)||_2}$, where $\Delta = 0$ if this results in $||\mathbf{u}||_1 \leq c$; otherwise, Δ is chosen to be a positive constant such that $||\mathbf{u}||_1 = c$.

   (b) Let $\mathbf{v}$ be the solution to

$$\underset{\mathbf{v}}{\text{minimize}}\{\frac{1}{2}||\mathbf{X}^T\mathbf{u} - \mathbf{v}||^2 + \lambda_1 \sum_{j=1}^{p} |v_j| + \lambda_2 \sum_{j=2}^{p} |v_j - v_{j-1}|\}. \tag{2.17}$$

3. $d = \mathbf{u}^T\mathbf{X}\mathbf{v}$.

Step 2(b) is a diagonal fused lasso regression problem, and can be performed using fast software implementing fused lasso regression, as described in Friedman et al. (2007), Tibshirani & Wang (2008), Hoefling (2009*a*), and Hoefling (2009*b*).

2.4 PMD for missing data, and choice of c_1 and c_2

The algorithm for computing the PMD can be applied even in the case of missing data. When some elements of the data matrix $\mathbf{X}$ are missing, those elements can simply be excluded from all computations. Let C denote the set of indices of nonmissing elements in $\mathbf{X}$. The criterion is as follows:

$$\begin{aligned} &\underset{\mathbf{u},\mathbf{v}}{\text{maximize}}\{\sum_{(i,j)\in C} X_{ij}u_iv_j\} \\ &\text{subject to } ||\mathbf{u}||^2 \leq 1, ||\mathbf{v}||^2 \leq 1, P_1(\mathbf{u}) \leq c_1, P_2(\mathbf{v}) \leq c_2. \end{aligned} \tag{2.18}$$

The PMD can therefore be used as a method for missing data imputation. This is related to SVD-based data imputation methods proposed in the literature; see e.g. Troyanskaya et al. (2001).

The possibility of computing the PMD in the presence of missing data leads to a simple and automated method for the selection of the constants c_1 and c_2 in the PMD criterion. We can treat c_1 and c_2 as tuning parameters, and can take an approach similar to crossvalidation in order to select their values. For simplicity, we demonstrate this method for the rank-one case here:

Algorithm 2.5: Selection of tuning parameters for PMD

1. From the original data matrix $\mathbf{X}$, construct B data matrices $\mathbf{X}_1, \ldots, \mathbf{X}_B$, each of which is missing a nonoverlapping $\frac{1}{B}$ of the elements of $\mathbf{X}$, sampled at random from

the rows and columns.

2. For each candidate value of c_1 and c_2, and for each $b = 1, \ldots, B$:

   (a) Fit the PMD to $\mathbf{X}_b$ with tuning parameters c_1 and c_2, and calculate $\hat{\mathbf{X}}_b = d\mathbf{u}\mathbf{v}^T$, the resulting estimate of $\mathbf{X}_b$.

   (b) Record the mean squared error of the estimate $\hat{\mathbf{X}}_b$. This mean squared error is obtained by computing the mean of the squared differences between elements of $\mathbf{X}$ and the corresponding elements of $\hat{\mathbf{X}}_b$, where the mean is taken only over elements that are missing from $\mathbf{X}_b$.

3. The optimal values of c_1 and c_2 are those which correspond to the lowest average mean squared error across $\mathbf{X}_1, \ldots, \mathbf{X}_B$. Alternatively, the optimal values are the smallest values that correspond to average mean squared error that is within one standard deviation of the lowest average mean squared error.

Note that in Step 1 of Algorithm 2.5, we construct each $\mathbf{X}_b$ by randomly removing scattered elements of the matrix $\mathbf{X}$. That is, we are not removing entire rows of $\mathbf{X}$ or entire columns of $\mathbf{X}$, but rather individual elements of the data matrix. This approach is related to proposals by Wold (1978) and Owen & Perry (2009).

Though c_1 and c_2 can always be chosen as described above, for certain applications crossvalidation may not be necessary. If the PMD is applied to a data set as a descriptive method in order to interpret the data, then one might simply fix c_1 and c_2 based on some other criterion. For instance, one could select small values of c_1 and c_2 in order to obtain factors that have a desirable level of sparsity.

To demonstrate the performance of Algorithm 2.5, we simulate data under the model

$$\mathbf{X} = \mathbf{u}\mathbf{v}^T + \boldsymbol{\epsilon} \tag{2.19}$$

where $\mathbf{u} \in \mathbb{R}^{50}$, $\mathbf{v} \in \mathbb{R}^{100}$, and $\boldsymbol{\epsilon} \in \mathbb{R}^{50 \times 100}$ is a matrix of independent and identically

distributed Gaussian noise terms. Moreover, $\mathbf{v}$ is sparse, with only 20 nonzero elements. We apply the crossvalidation approach described above to $\mathbf{X}$. We fix $c_1 = \sqrt{50}$ since we know that $\mathbf{u}$ is not sparse; this has the effect of making the L_1 constraint on $\mathbf{u}$ inactive. We try a range of values of c_2, from 1 to $\sqrt{100} = 10$. The results are shown in Figure 2.2. As c_2 increases, the number of nonzero elements of $\mathbf{v}$ increases. When the number of nonzero elements of the estimate for $\mathbf{v}$ is less than 20, then increasing c_2 results in a reduction in the crossvalidation error. However, when more than 20 elements are nonzero in the estimate of $\mathbf{v}$, then increasing c_2 has essentially no effect on the crossvalidation error.

On a less contrived example, we would not expect Algorithm 5.1 to yield such a clear indication of the optimal tuning parameter value. However, the algorithm can often provide guidance on selection of a suitable tuning parameter value.

2.5 Relationship between PMD and other matrix decompositions

In the statistical and machine learning literature, a number of matrix decompositions have been developed. We present some of these decompositions here, as they are related to the PMD. The best-known of these decompositions is the SVD, which takes the form (2.1). The SVD has a number of interesting properties, but the vectors $\mathbf{u}_k$ and $\mathbf{v}_k$ of the SVD have (in general) no nonzero elements, and the elements may be positive or negative. These qualities result in vectors $\mathbf{u}_k$ and $\mathbf{v}_k$ that are often not interpretable.

Lee & Seung (1999, 2001) developed the nonnegative matrix factorization (NNMF) in order to improve upon the interpretability of the SVD. The matrix $\mathbf{X}$ is approximated as

$$\mathbf{X} \approx \sum_{k=1}^{K} \mathbf{u}_k \mathbf{v}_k^T, \tag{2.20}$$

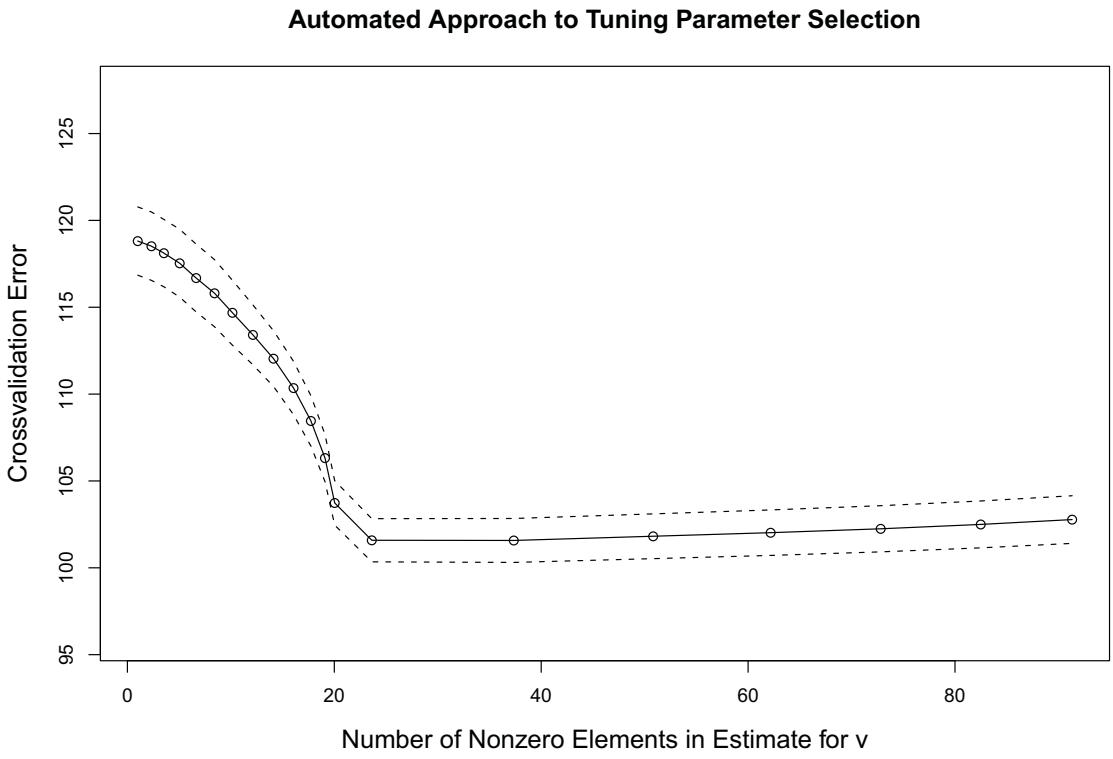

Figure 2.2: Algorithm 2.5 was applied to data generated under the simple low rank model (2.19). The solid line indicates the mean crossvalidation error rate obtained over 20 simulated data sets. The dashed lines indicate one standard error above and below the mean crossvalidation error rates. Once the estimate for $\mathbf{v}$ has more than 20 nonzero elements, there is little benefit to increasing c_2 in terms of crossvalidation error.

where the elements of $\mathbf{u}_k$ and $\mathbf{v}_k$ are constrained to be nonnegative. The resulting factors $\mathbf{u}_k$ and $\mathbf{v}_k$ may be interpretable: the authors apply the NNMF to a database of faces, and show that the resulting factors represent facial features. The SVD does not result in interpretable facial features.

Hoyer (2002, 2004) presents the nonnegative sparse coding (NNSC), an extension of the NNMF that results in nonnegative vectors $\mathbf{v}_k$ and $\mathbf{u}_k$, one or both of which may be sparse. Sparsity is achieved using an L_1 penalty. Since NNSC enforces a nonnegativity constraint, the resulting vectors can be quite different from those obtained via the PMD; moreover, the iterative algorithm for finding the NNSC vectors is not guaranteed to decrease the objective at each step.

Lazzeroni & Owen (2002) present the plaid model, which in the simplest case takes the form

$$\begin{aligned} &\underset{d_k,\mathbf{u}_k,\mathbf{v}_k}{\text{minimize}}\{||\mathbf{X} - \sum_{k=1}^{K} d_k \mathbf{u}_k \mathbf{v}_k^T||_F^2\} \\ &\text{subject to } u_{ik} \in \{0,1\}, v_{jk} \in \{0,1\}. \end{aligned} \tag{2.21}$$

Though the plaid model results in interpretable factors, it has the drawback that problem (2.21) cannot be optimized exactly due to the nonconvex form of the constraints on $\mathbf{u}_k$ and $\mathbf{v}_k$. Unlike the PMD, the problem is not biconvex.

2.6 Example: PMD applied to DNA copy number data

Comparative genomic hybridization (CGH) is a technique for measuring the DNA copy number of a tissue sample at selected locations in the genome (see e.g. Kallioniemi et al. 1992). Each CGH measurement represents the $\log_2$ ratio between the number of copies of a gene in the tissue of interest and the number of copies of that same gene in reference cells; we will assume that these measurements are ordered along the chromosome. In general,

there should be two copies of each chromosome in an individual's genome: one per parent. Consequently, CGH data tends to be sparse. Under certain conditions, chromosomal regions spanning multiple genes may be amplified or deleted in a given sample, and so CGH data tends to be piecewise constant.

A number of methods have been proposed for identification of regions of copy number gain and loss in a single CGH sample (see e.g. Picard et al. 2005, Venkatraman & Olshen 2007). In particular, the proposal of Tibshirani & Wang (2008) involves using the fused lasso to approximate a CGH sample as a sparse and piecewise constant signal:

$$\underset{\boldsymbol{\beta}}{\text{minimize}}\{\frac{1}{2}\sum_{j=1}^{p}(y_j-\beta_j)^2+\lambda_1\sum_{j=1}^{p}|\beta_j|+\lambda_2\sum_{j=2}^{p}|\beta_j-\beta_{j-1}|\}. \tag{2.22}$$

In (2.22), $\mathbf{y}$ is a vector of length p corresponding to measured log copy number gain/loss, ordered along the chromosome, and the solution $\hat{\boldsymbol{\beta}}$ is a smoothed estimate of the copy number. Here, λ_1 and λ_2 are nonnegative tuning parameters. When λ_1 is large, $\hat{\boldsymbol{\beta}}$ will be sparse, and when λ_2 is large, $\hat{\boldsymbol{\beta}}$ will be piecewise constant.

Now, suppose that multiple CGH samples are available. We expect some patterns of gain and loss to be shared between some of the samples, and we wish to identify those patterns and samples. Let $\mathbf{X}$ denote the data matrix; the n rows denote the samples, and the p columns correspond to (ordered) CGH spots. In this case, the use of PMD(L_1,FL) is appropriate, because we wish to encourage sparsity in $\mathbf{u}$ (corresponding to a subset of samples) and sparsity and smoothness in $\mathbf{v}$ (corresponding to chromosomal regions). The use of PMD(L_1,FL) in this context is related to a proposal by Nowak (2009). One could apply PMD(L_1,FL) to all chromosomes together, making sure that smoothness in the fused lasso penalty is not imposed between chromosomes, or one could apply PMD(L_1,FL) to each chromosome separately.

We demonstrate this method on a simple simulated example. We simulate 12 samples, each of which consists of copy number measurements on 1000 spots on a single chromosome.

Five of the twelve samples contain a region of gain from spots 100-500. In Figure 2.3, we compare the results of PMD(L_1,L_1) to PMD(L_1,FL). It is clear that the latter method uncovers the region of gain and the set of samples in which that gained region is present.

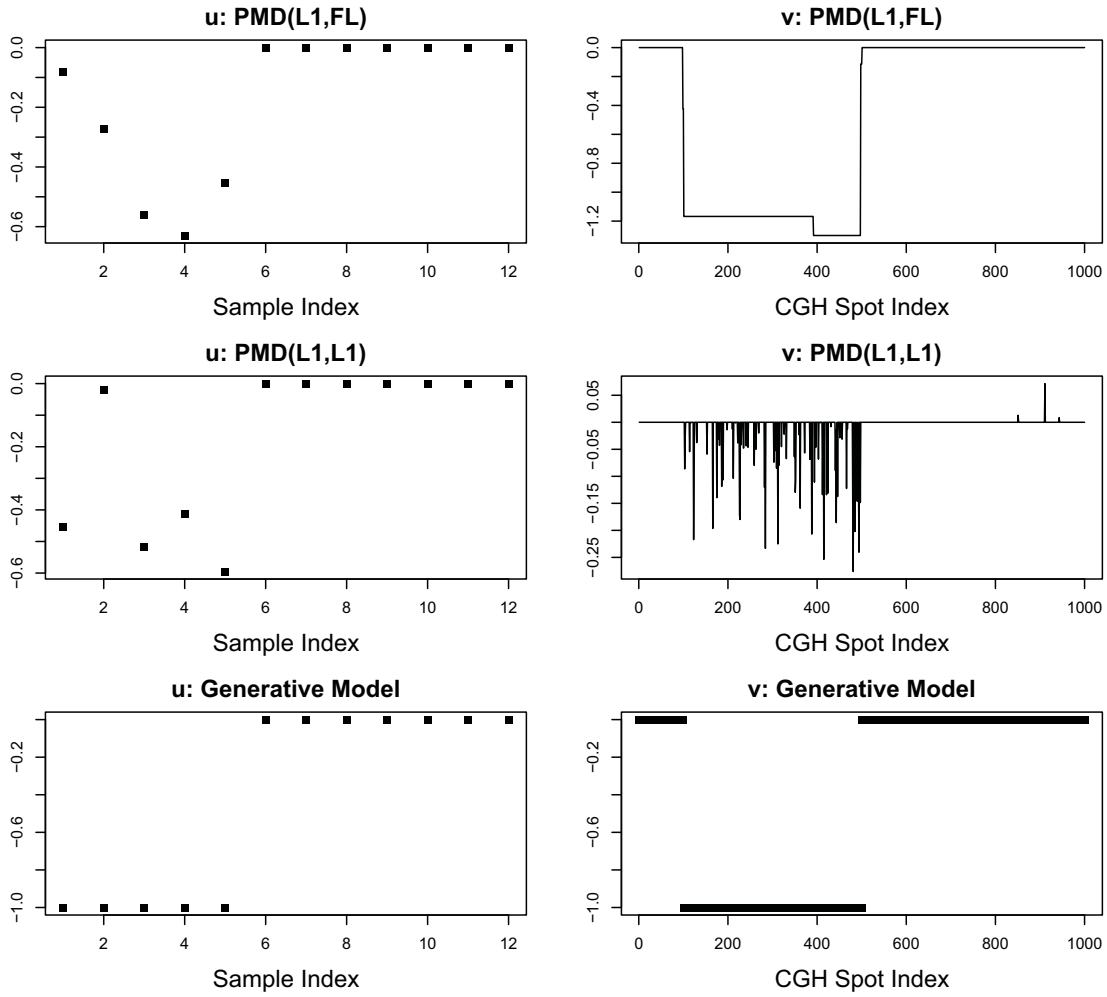

Figure 2.3: Simulated CGH data. **Top**: Results of PMD(L_1,FL). **Middle**: Results of PMD(L_1,L_1). **Bottom**: Generative model. PMD(L_1,FL) successfully identifies both the region of gain and the subset of samples for which that region is present.

2.7 Proofs

2.7.1 Proof of Proposition 2.1.1

Proof. Let $\mathbf{u}_k$ and $\mathbf{v}_k$ denote column k of $\mathbf{U}$ and $\mathbf{V}$, respectively. We prove the proposition by expanding out the squared Frobenius norm, and rearranging terms:

$$\begin{aligned}
||\mathbf{X} - \mathbf{UDV}^T||_F^2 &= \text{tr}((\mathbf{X} - \mathbf{UDV}^T)^T(\mathbf{X} - \mathbf{UDV}^T)) \\
&= \text{tr}(\mathbf{VDU}^T\mathbf{UDV}^T) - 2\text{tr}(\mathbf{VDU}^T\mathbf{X}) + ||\mathbf{X}||_F^2 \\
&= \sum_{k=1}^{K} d_k^2 - 2\text{tr}(\mathbf{DU}^T\mathbf{XV}) + ||\mathbf{X}||_F^2 \\
&= \sum_{k=1}^{K} d_k^2 - 2\sum_{k=1}^{K} d_k \mathbf{u}_k^T\mathbf{X}\mathbf{v}_k + ||\mathbf{X}||_F^2 \qquad (2.23)
\end{aligned}$$

□

2.7.2 Proof of Proposition 2.3.1

Proof. We wish to solve

$$\underset{\mathbf{u}}{\text{minimize}}\{-\mathbf{u}^T\mathbf{a}\} \text{ subject to } ||\mathbf{u}||^2 \leq 1, ||\mathbf{u}||_1 \leq c_1. \qquad (2.24)$$

The KKT conditions for optimality are as follows (Boyd & Vandenberghe 2004):

$$0 = -\mathbf{a} + 2\lambda\mathbf{u} + \Delta\Gamma, \qquad (2.25)$$

$$\lambda \geq 0,\ \Delta \geq 0, \qquad (2.26)$$

$$||\mathbf{u}||^2 \leq 1,\ ||\mathbf{u}||_1 \leq c_1, \qquad (2.27)$$

$$\lambda(||\mathbf{u}||^2 - 1) = 0,\ \Delta(||\mathbf{u}||_1 - c_1) = 0, \qquad (2.28)$$

where Γ is a subgradient of $||\mathbf{u}||_1$. That is, $\Gamma_j = \text{sgn}(u_j)$ if $u_j \neq 0$; otherwise, $\Gamma_j \in [-1, 1]$. We consider four possible cases.

1. $\lambda = 0$ and $\Delta = 0$. Then (2.25) implies that $\mathbf{a} = 0$. In this case, it is easily seen that $\mathbf{u} = 0$ is a solution to (2.24).

2. $\lambda = 0$ and $\Delta > 0$. Then (2.25) implies that $\frac{a_j}{\Delta} = \text{sgn}(u_j)$ if $u_j \neq 0$ and $\frac{a_j}{\Delta} \in [-1, 1]$ if $u_j = 0$. So $\Delta \geq \max_j |a_j|$. If $\Delta > \max_j |a_j|$ then $\mathbf{u} = 0$; this would contradict (2.28). So $\Delta = \max_j |a_j|$. We have assumed that there is a unique element of $\mathbf{a}$ with maximal absolute value. It follows that $u_j = c_1 \text{sgn}(a_j)$ if j is the element of $\mathbf{a}$ with maximal absolute value, and is 0 otherwise. This means that $||\mathbf{u}||^2 = c_1^2$. By (2.27), this can occur only if $c_1 \leq 1$. In general, we restrict c_1 to be between 1 and $\sqrt{n}$, so this case will occur only if $c_1 = 1$.

3. $\lambda > 0$ and $\Delta = 0$. Then by (2.25), $\mathbf{u} = \frac{\mathbf{a}}{2\lambda}$. By (2.28), $||\mathbf{u}||^2 = 1$. So $\mathbf{u} = \frac{\mathbf{a}}{||\mathbf{a}||_2}$. By (2.27), this case can occur only if the L_1 norm of $\frac{\mathbf{a}}{||\mathbf{a}||_2}$ is less than or equal to c_1.

4. $\lambda > 0$ and $\Delta > 0$. One can show that (2.25) yields $u_j = \frac{S(a_j, \Delta)}{2\lambda}$ where $\lambda, \Delta > 0$ are chosen so that (2.27) holds. So $\lambda = \frac{1}{2}||S(\mathbf{a}, \Delta)||_2$ and $\Delta > 0$ is chosen so that $\mathbf{u}$ has L_1 norm equal to c_1.

So we have seen that if $\mathbf{a} \neq 0$ and $c_1 > 1$ then either Case 3 or Case 4 will occur. By inspection, the two cases can be combined as follows:

$$\mathbf{u} = \frac{S(\mathbf{a}, \Delta)}{||S(\mathbf{a}, \Delta)||_2} \tag{2.29}$$

where $\Delta = 0$ if this results in $||\mathbf{u}||_1 \leq c_1$; otherwise, $\Delta > 0$ is such that $||\mathbf{u}||_1 = c_1$. □

Chapter 3

Sparse principal components analysis

In this chapter, we propose a method for sparse principal components analysis. This work also appears in Witten et al. (2009).

3.1 Three methods for sparse principal components analysis

Let $\mathbf{X}$ denote an $n \times p$ data matrix with centered columns. *Principal components analysis* (PCA) is a popular method for dimension reduction and data visualization in statistics and other fields. The principal components of $\mathbf{X}$ are simply the eigenvectors of the matrix $\mathbf{X}^T\mathbf{X}$. When p is large, the principal components of $\mathbf{X}$ can be hard to interpret because all p features have nonzero loadings. In this case, one might wish to obtain principal components that are sparse.

Several methods have been proposed for estimating sparse principal components, based on either the maximum-variance property of principal components, or the regression/reconstruction error property. In this chapter, we present two existing methods for sparse PCA from the literature, as well as a new method based on the PMD. We will then go on to show that

these three methods are closely related to each other. We will take advantage of the connection between PMD and one of the other methods in order to develop a fast algorithm for what was previously a computationally difficult formulation for sparse PCA.

The three methods for sparse PCA are as follows:

1. **SPCA:** Zou et al. (2006) exploit the regression/reconstruction error property of principal components in order to obtain sparse principal components. For a single component, their sparse principal components (SPCA) technique solves

$$\begin{aligned} &\underset{\boldsymbol{\theta},\mathbf{v}}{\text{minimize}}\{||\mathbf{X}-\mathbf{X}\mathbf{v}\boldsymbol{\theta}^T||_F^2+\lambda_1||\mathbf{v}||^2+\lambda_2||\mathbf{v}||_1\} \\ &\text{subject to } ||\boldsymbol{\theta}||_2=1, \end{aligned} \tag{3.1}$$

   where $\lambda_1, \lambda_2 \geq 0$ and $\mathbf{v}$ and $\boldsymbol{\theta}$ are p-vectors. The criterion can equivalently be written with an inequality L_2 bound on $\boldsymbol{\theta}$, in which case it is biconvex in $\boldsymbol{\theta}$ and $\mathbf{v}$. Note that when $\lambda_2 = 0$ in (3.1), then the solution $\hat{\mathbf{v}}$ is the first principal component of $\mathbf{X}$, up to a scaling. When λ_2 is large, then $\hat{\mathbf{v}}$ is sparse.

2. **SCoTLASS:** The *SCoTLASS* procedure of Jolliffe et al. (2003) uses the maximal variance characterization for principal components. The first sparse principal component solves the problem

$$\underset{\mathbf{v}}{\text{maximize}}\{\mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}\} \text{ subject to } ||\mathbf{v}||^2 \leq 1, ||\mathbf{v}||_1 \leq c, \tag{3.2}$$

   and subsequent components solve the same problem with the additional constraint that they must be orthogonal to the previous components. When c is large, then (3.2) simply yields the first principal component of $\mathbf{X}$, and when c is small, then the solution is sparse. This problem is not convex, since a convex objective must be maximized, and the computations are difficult. Trendafilov & Jolliffe (2006) provide

a projected gradient algorithm for optimizing (3.2). We will show that this criterion can be optimized much more simply by direct application of Algorithm 2.3 in Chapter 2.3.

3. **SPC:** We propose a new method for sparse PCA. Consider the PMD criterion (2.7) with $P_2(\mathbf{v}) = ||\mathbf{v}||_1$, and no P_1 constraint on $\mathbf{u}$:

$$\underset{\mathbf{u},\mathbf{v}}{\text{maximize}}\{\mathbf{u}^T\mathbf{X}\mathbf{v}\} \text{ subject to } ||\mathbf{u}||^2 \leq 1, ||\mathbf{v}||^2 \leq 1, ||\mathbf{v}||_1 \leq c. \tag{3.3}$$

Then the solution $\hat{\mathbf{v}}$ is the first sparse principal component. We will refer to (3.3) as the *sparse principal components* (SPC) criterion. When c is large, then the solution $\hat{\mathbf{v}}$ is simply the first principal component of $\mathbf{X}$, and when c is small, then $\hat{\mathbf{v}}$ is sparse.

The SPC algorithm is as follows:

Algorithm 3.1: Computation of first sparse principal component

1. Initialize $\mathbf{v}$ to have L_2 norm 1.
2. Iterate:
   (a) Let $\mathbf{u} = \frac{\mathbf{X}\mathbf{v}}{||\mathbf{X}\mathbf{v}||_2}$.
   (b) Let $\mathbf{v} = \frac{S(\mathbf{X}^T\mathbf{u},\Delta)}{||S(\mathbf{X}^T\mathbf{u},\Delta)||_2}$, where $\Delta = 0$ if this results in $||\mathbf{v}||_1 \leq c$; otherwise, Δ is chosen to be a positive constant such that $||\mathbf{v}||_1 = c$.

Now, consider the SPC criterion (3.3). It is easily shown that if $\mathbf{v}$ is fixed, and we seek $\mathbf{u}$ to maximize (3.3), then the optimal $\mathbf{u}$ will be $\frac{\mathbf{X}\mathbf{v}}{||\mathbf{X}\mathbf{v}||_2}$. Therefore, $\mathbf{v}$ that solves (3.3) also solves

$$\underset{\mathbf{v}}{\text{maximize}}\{\mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}\} \text{ subject to } ||\mathbf{v}||_1 \leq c, ||\mathbf{v}||^2 \leq 1. \tag{3.4}$$

We recognize (3.4) as the *SCoTLASS* criterion (3.2). Now, since we have a fast iterative algorithm for solving (3.3), this means that we have also developed a fast method to optimize

the *SCoTLASS* criterion (keeping in mind that we do not expect to obtain the global optimum using an iterative approach; for more information see Gorski et al. 2007). We can extend SPC to find the first K sparse principal components, as in Algorithm 2.2. Note, however, that only the first component is the solution to the *SCoTLASS* criterion, since we are not enforcing the constraint that component $\mathbf{v}_k$ be orthogonal to components $\mathbf{v}_1, \ldots, \mathbf{v}_{k-1}$.

It is also not hard to show that PMD applied to a covariance matrix with symmetric L_1 penalties on the rows and columns, as follows,

$$\begin{aligned} &\underset{\mathbf{u},\mathbf{v}}{\text{maximize}}\{\mathbf{u}^T\mathbf{X}^T\mathbf{X}\mathbf{v}\} \\ &\text{subject to } ||\mathbf{u}||^2 \leq 1, ||\mathbf{u}||_1 \leq c, ||\mathbf{v}||^2 \leq 1, ||\mathbf{v}||_1 \leq c, \end{aligned} \tag{3.5}$$

results in solutions $\mathbf{u} = \mathbf{v}$. (This follows from the Cauchy-Schwarz inequality applied to vectors $\mathbf{Xv}$ and $\mathbf{Xu}$.) As a result, these solutions solve the *SCoTLASS* criterion as well. This also means that SPC can be performed using the covariance matrix instead of the raw data in cases where this is more convenient - e.g. if $n \gg p$, or if the raw data is unavailable.

We have shown that the SPC criterion is equivalent to the *SCoTLASS* criterion for one component, and that the fast algorithm for the former can be used to solve the latter. It turns out that there also is a connection between the SPCA criterion and the SPC criterion. Consider a modified version of the SPCA criterion (3.1) that uses the bound form, rather than the Lagrange form, of the constraints on $\mathbf{v}$:

$$\underset{\boldsymbol{\theta},\mathbf{v}}{\text{minimize}}\{||\mathbf{X} - \mathbf{X}\mathbf{v}\boldsymbol{\theta}^T||_F^2\} \text{ subject to } ||\mathbf{v}||^2 \leq 1, ||\mathbf{v}||_1 \leq c, ||\boldsymbol{\theta}||^2 = 1. \tag{3.6}$$

With $||\boldsymbol{\theta}||^2 = 1$, we have

$$\begin{aligned} ||\mathbf{X} - \mathbf{X}\mathbf{v}\boldsymbol{\theta}^T||_F^2 &= \text{tr}((\mathbf{X} - \mathbf{X}\mathbf{v}\boldsymbol{\theta}^T)^T(\mathbf{X} - \mathbf{X}\mathbf{v}\boldsymbol{\theta}^T)) \\ &= \text{tr}(\mathbf{X}^T\mathbf{X}) - 2\text{tr}(\boldsymbol{\theta}\mathbf{v}^T\mathbf{X}^T\mathbf{X}) + \text{tr}(\boldsymbol{\theta}\mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}\boldsymbol{\theta}^T) \\ &= \text{tr}(\mathbf{X}^T\mathbf{X}) - 2\mathbf{v}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} + \mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}. \end{aligned} \tag{3.7}$$

So solving (3.6) is equivalent to

$$\underset{\boldsymbol{\theta},\mathbf{v}}{\text{maximize}}\{2\mathbf{v}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} - \mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}\} \text{ subject to } ||\boldsymbol{\theta}||^2 = 1, ||\mathbf{v}||^2 \leq 1, ||\mathbf{v}||_1 \leq c \tag{3.8}$$

or equivalently

$$\underset{\boldsymbol{\theta},\mathbf{v}}{\text{maximize}}\{2\mathbf{v}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} - \mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}\} \text{ subject to } ||\boldsymbol{\theta}||^2 \leq 1, ||\mathbf{v}||^2 \leq 1, ||\mathbf{v}||_1 \leq c. \tag{3.9}$$

Now, suppose we add an additional constraint to (3.6): that is, let us require also that $||\boldsymbol{\theta}||_1 \leq c$. We solve

$$\begin{aligned} &\underset{\boldsymbol{\theta},\mathbf{v}}{\text{maximize}}\{2\mathbf{v}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} - \mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}\} \\ &\text{subject to } ||\mathbf{v}||^2 \leq 1, ||\mathbf{v}||_1 \leq c, ||\boldsymbol{\theta}||^2 \leq 1, ||\boldsymbol{\theta}||_1 \leq c. \end{aligned} \tag{3.10}$$

Note that for any vectors $\mathbf{w}$ and $\mathbf{z}$, $||\mathbf{z} - \mathbf{w}||^2 \geq 0$. This means that $\mathbf{z}^T\mathbf{z} \geq 2\mathbf{w}^T\mathbf{z} - \mathbf{w}^T\mathbf{w}$. Let $\mathbf{w} = \mathbf{X}\mathbf{v}$ and $\mathbf{z} = \mathbf{X}\boldsymbol{\theta}$; it follows that $\boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} \geq 2\mathbf{v}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} - \mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}$. So (3.10) is maximized when $\mathbf{v} = \boldsymbol{\theta}$. That is, $\mathbf{v}$ that solves (3.10) also solves

$$\underset{\mathbf{v}}{\text{maximize}}\{\mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}\} \text{ subject to } ||\mathbf{v}||^2 \leq 1, ||\mathbf{v}||_1 \leq c, \tag{3.11}$$

which of course is simply the *SCoTLASS* criterion (3.2) again. Therefore, we have shown that if a symmetric L_1 constraint on $\boldsymbol{\theta}$ is added to the bound form of the SPCA criterion,

then the *SCoTLASS* criterion results. From this argument, it is also clear that the solution to the bound form of SPCA will give lower reconstruction error (defined as $||\mathbf{X} - \mathbf{X}\mathbf{v}\boldsymbol{\theta}^T||_F^2$) than the solution to the *SCoTLASS* criterion.

Our extension of PMD to the problem of identifying sparse principal components is closely related to a proposal by Shen & Huang (2008). They present a method for identifying sparse principal components via a regularized low-rank matrix approximation, as follows:

$$\underset{\mathbf{u},\mathbf{v}}{\text{minimize}}\{||\mathbf{X} - \mathbf{u}\mathbf{v}^T||_F^2 + P_\lambda(\mathbf{v})\} \text{ subject to } ||\mathbf{u}||_2 = 1. \tag{3.12}$$

They then scale the solution $\hat{\mathbf{v}}$ in order to have L_2 norm 1; this is the first sparse principal component of their method. They present a number of forms for $P_\lambda(\mathbf{v})$, including $P_\lambda(\mathbf{v}) = ||\mathbf{v}||_1$. This is very close in spirit to the SPC criterion (3.3), and in fact the algorithm is almost the same. But since Shen & Huang (2008) use the Lagrange form of the constraint on $\mathbf{v}$, their formulation does not solve the *SCoTLASS* criterion. Our method unifies the regularized low-rank matrix approximation approach of Shen & Huang (2008) with the maximum-variance criterion of Jolliffe et al. (2003) and the SPCA method of Zou et al. (2006).

To summarize, in our view, the *SCoTLASS* criterion (3.2) is the simplest, most natural way to define the notion of sparse principal components. Unfortunately, the criterion is difficult to optimize. Our SPC criterion (3.3) recasts this problem as a biconvex one, leading to an extremely simple algorithm for the solution of the first *SCoTLASS* component. Furthermore, the SPCA criterion (3.1) is somewhat complex. But we have shown that when a natural symmetric constraint is added to the SPCA criterion (3.1), it is also equivalent to (3.2) and (3.3). Taken as a whole, these arguments point to the SPC criterion (3.3) as the criterion of choice for this problem, at least for a single component.

3.2 Example: SPC applied to gene expression data

We compare the proportion of variance explained by SPC and SPCA on a publicly available gene expression data set available from `http://icbp.lbl.gov/breastcancer/`, and described in Chin et al. (2006), consisting of 19,672 gene expression measurements on 89 samples. For computational reasons, we use only the subset of the data consisting of the 5% of genes with highest variance. We compute the first 25 sparse principal components for SPC, using the constraint on $\mathbf{v}$ that results in an average of 195 genes with nonzero elements per sparse component. We then perform SPCA on the same data, with tuning parameters chosen so that each loading has the same number of nonzero elements obtained using the SPC method. Figure 3.1 shows the proportion of variance explained by the first k sparse principal components, defined as $\text{tr}(\mathbf{X}_k^T\mathbf{X}_k)$, where $\mathbf{X}_k = \mathbf{X}\mathbf{V}_k(\mathbf{V}_k^T\mathbf{V}_k)^{-1}\mathbf{V}_k^T$, and where $\mathbf{V}_k$ is the matrix that has the first k sparse principal components as its columns. This definition is proposed in Shen & Huang (2008). SPC results in a substantially greater proportion of variance explained, as expected.

3.3 Another option for SPC with multiple factors

We now consider the problem of extending the SPC method to obtain multiple components. One could extend to multiple components as proposed in Algorithm 2.2. For instance, this was done in Figure 3.1. As mentioned in Chapter 3.1, the first sparse principal component of our SPC method optimizes the *SCoTLASS* criterion. But subsequent sparse principal components obtained using Algorithm 2.2 do not, since Algorithm 2.2 does not enforce that $\mathbf{v}_k$ be orthogonal to $\mathbf{v}_1, \ldots, \mathbf{v}_{k-1}$. It is not obvious that SPC can be extended to achieve orthogonality among subsequent $\mathbf{v}_i$'s, or even that orthogonality is desirable. However, SPC can be easily extended to give something similar to orthogonality.

Instead of applying Algorithm 2.2, one could obtain multiple factors $\mathbf{u}_k, \mathbf{v}_k$ by optimizing

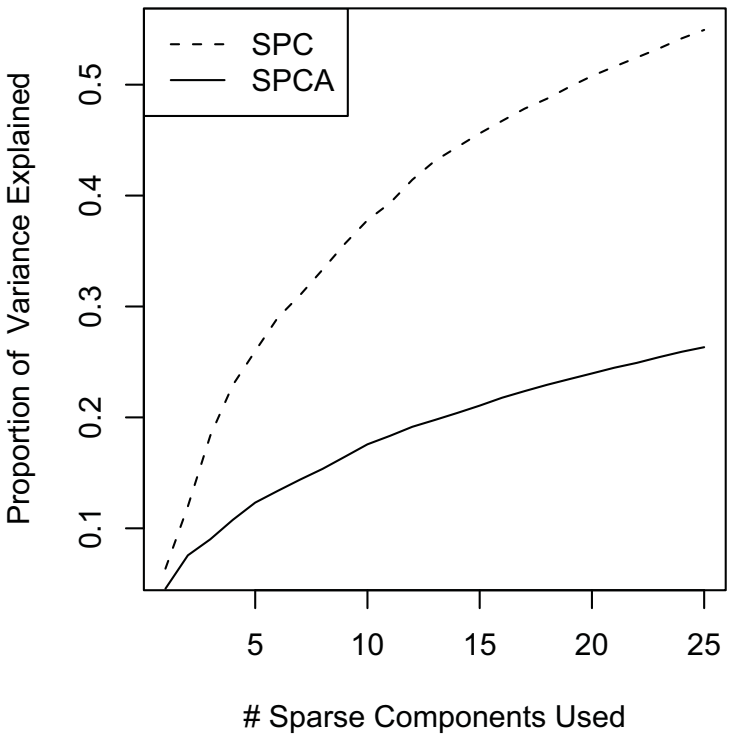

Figure 3.1: Breast cancer gene expression data. A greater proportion of variance is explained when SPC is used to obtain the sparse principal components, rather than SPCA. Multiple SPC components were obtained as described in Algorithm 2.2.

the following criterion, for $k > 1$:

$$\begin{aligned} &\underset{\mathbf{u}_k, \mathbf{v}_k}{\text{maximize}} \{\mathbf{u}_k^T \mathbf{X} \mathbf{v}_k\} \\ &\text{subject to } ||\mathbf{v}_k||^2 \leq 1, ||\mathbf{v}_k||_1 \leq c, ||\mathbf{u}_k||^2 \leq 1, \mathbf{u}_k^T \mathbf{u}_i = 0 \; \forall i < k. \end{aligned} \tag{3.13}$$

With $\mathbf{u}_k$ fixed, one can easily solve (3.13) for $\mathbf{v}_k$ (see Proposition 2.3.1). With $\mathbf{v}_k$ fixed, the problem is as follows: we must find $\mathbf{u}_k$ that solves

$$\begin{aligned} &\underset{\mathbf{u}_k}{\text{maximize}} \{\mathbf{u}_k^T \mathbf{X} \mathbf{v}_k\} \\ &\text{subject to } ||\mathbf{u}_k||^2 \leq 1, \mathbf{u}_k^T \mathbf{u}_i = 0 \; \forall i < k. \end{aligned} \tag{3.14}$$

Let $\mathbf{U}_{k-1}^{\perp}$ denote an orthonormal basis for the space that is orthogonal to $\mathbf{u}_1, \ldots, \mathbf{u}_{k-1}$. It follows that $\mathbf{u}_k$ is in the column space of $\mathbf{U}_{k-1}^{\perp}$, and so can be written as $\mathbf{u}_k = \mathbf{U}_{k-1}^{\perp} \boldsymbol{\theta}$. Note also that $||\mathbf{u}_k||_2 = ||\boldsymbol{\theta}||_2$. So (3.14) is equivalent to solving

$$\underset{\boldsymbol{\theta}}{\text{maximize}} \{\boldsymbol{\theta}^T {\mathbf{U}_{k-1}^{\perp}}^T \mathbf{X} \mathbf{v}_k\} \text{ subject to } ||\boldsymbol{\theta}||^2 \leq 1, \tag{3.15}$$

and so we find that the optimal $\boldsymbol{\theta}$ is

$$\boldsymbol{\theta} = \frac{{\mathbf{U}_{k-1}^{\perp}}^T \mathbf{X} \mathbf{v}_k}{||{\mathbf{U}_{k-1}^{\perp}}^T \mathbf{X} \mathbf{v}_k||_2}. \tag{3.16}$$

Therefore, the value of $\mathbf{u}_k$ that solves (3.14) is

$$\mathbf{u}_k = \frac{\mathbf{U}_{k-1}^{\perp} {\mathbf{U}_{k-1}^{\perp}}^T \mathbf{X} \mathbf{v}_k}{||{\mathbf{U}_{k-1}^{\perp}}^T \mathbf{X} \mathbf{v}_k||_2} = \frac{\mathbf{P}_{k-1}^{\perp} \mathbf{X} \mathbf{v}_k}{||\mathbf{P}_{k-1}^{\perp} \mathbf{X} \mathbf{v}_k||_2} \tag{3.17}$$

where $\mathbf{P}_{k-1}^{\perp} = \mathbf{I} - \sum_{i=1}^{k-1} \mathbf{u}_i \mathbf{u}_i^T$. So we can use this update step for $\mathbf{u}_k$ to develop an iterative algorithm to find multiple sparse principal components in such way that the $\mathbf{u}_k$'s

are orthogonal.

Algorithm 3.2: Alternative approach for computation of kth sparse principal component

1. Initialize $\mathbf{v}_k$ to have L_2 norm 1.
2. Let $\mathbf{P}_{k-1}^{\perp} = \mathbf{I} - \sum_{i=1}^{k-1} \mathbf{u}_i \mathbf{u}_i^T$.
3. Iterate until convergence:
   (a) Let $\mathbf{u}_k = \frac{\mathbf{P}_{k-1}^{\perp} \mathbf{X} \mathbf{v}_k}{||\mathbf{P}_{k-1}^{\perp} \mathbf{X} \mathbf{v}_k||_2}$.
   (b) Let $\mathbf{v}_k = \frac{S(\mathbf{X}^T \mathbf{u}_k, \Delta)}{||S(\mathbf{X}^T \mathbf{u}_k, \Delta)||_2}$, where $\Delta = 0$ if this results in $||\mathbf{v}_k||_1 \leq c$; otherwise, Δ is chosen to be a positive constant such that $||\mathbf{v}_k||_1 = c$.

Though we have not guaranteed that the $\mathbf{v}_k$'s will be exactly orthogonal, they are unlikely to be very correlated, since the different $\mathbf{v}_k$'s each are associated with orthogonal $\mathbf{u}_k$'s. This approach can be used to obtain multiple components of the PMD whenever a general convex penalty function is applied to either $\mathbf{u}_k$ or $\mathbf{v}_k$, but not to both. When it is applicable, Algorithm 3.2 may be preferable to Algorithm 2.2 since the former results in components that are closer to being orthogonal.

3.4 SPC as a minorization algorithm for *SCoTLASS*

Here, we show that Algorithm 3.1 can be interpreted as a *minorization-maximization* (or simply *minorization*) algorithm for the *SCoTLASS* problem (3.2). Minorization algorithms are discussed in Lange et al. (2000), Lange (2004), and Hunter & Lange (2004). We begin with a brief review of minorization algorithms.

Consider the problem

$$\underset{\mathbf{v}}{\text{maximize}}\{f(\mathbf{v})\}. \tag{3.18}$$

If f is a concave function, then standard tools from convex optimization (see e.g. Boyd & Vandenberghe 2004) can be used to solve (3.18). If not, solving (3.18) can be difficult. Minorization refers to a general strategy for this problem. The function $g(\mathbf{v}, \mathbf{v}^{(m)})$ is said to minorize the function $f(\mathbf{v})$ at the point $\mathbf{v}^{(m)}$ if

$$f(\mathbf{v}^{(m)}) = g(\mathbf{v}^{(m)}, \mathbf{v}^{(m)}), \quad f(\mathbf{v}) \geq g(\mathbf{v}, \mathbf{v}^{(m)}) \; \forall \mathbf{v}. \tag{3.19}$$

A minorization algorithm for solving (3.18) involves initializing $\mathbf{v}^{(0)}$, and then iterating:

$$\mathbf{v}^{(m+1)} = \text{argmax}_{\mathbf{v}}\{g(\mathbf{v}, \mathbf{v}^{(m)})\}. \tag{3.20}$$

Then by (3.19),

$$f(\mathbf{v}^{(m+1)}) \geq g(\mathbf{v}^{(m+1)}, \mathbf{v}^{(m)}) \geq g(\mathbf{v}^{(m)}, \mathbf{v}^{(m)}) = f(\mathbf{v}^{(m)}). \tag{3.21}$$

This means that in each iteration the objective of (3.18) is nondecreasing. However, we do not expect to arrive at the global optimum of (3.18) using a minorization approach. A good minorization function is one for which (3.20) is easily solved. For instance, if $g(\mathbf{v}, \mathbf{v}^{(m)})$ is concave in $\mathbf{v}$ then standard convex optimization tools can be applied.

Now, in the case of SPC, notice that the Cauchy-Schwarz inequality implies that

$$\mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v} \geq \frac{(\mathbf{v}^{(m)^T}\mathbf{X}^T\mathbf{X}\mathbf{v})^2}{\mathbf{v}^{(m)^T}\mathbf{X}^T\mathbf{X}\mathbf{v}^{(m)}}, \tag{3.22}$$

and equality holds when $\mathbf{v} = \mathbf{v}^{(m)}$. So $\frac{(\mathbf{v}^{(m)^T}\mathbf{X}^T\mathbf{X}\mathbf{v})^2}{\mathbf{v}^{(m)^T}\mathbf{X}^T\mathbf{X}\mathbf{v}^{(m)}}$ minorizes $\mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}$ at $\mathbf{v}^{(m)}$.

Therefore, a minorization algorithm for the *SCoTLASS* problem (3.2) is as follows:

Algorithm 3.3: Minorization algorithm for SCoTLASS

1. Initialize $\mathbf{v}^0$.

2. For $m = 1, 2, \ldots$: Let $\mathbf{v}^{(m)}$ solve

$$\underset{\mathbf{v}}{\text{maximize}}\{\frac{(\mathbf{v}^{(m-1)^T}\mathbf{X}^T\mathbf{X}\mathbf{v})^2}{\mathbf{v}^{(m-1)^T}\mathbf{X}^T\mathbf{X}\mathbf{v}^{(m-1)}}\} \text{ subject to } ||\mathbf{v}||^2 \leq 1, ||\mathbf{v}||_1 \leq c, \tag{3.23}$$

or equivalently

$$\underset{\mathbf{v}}{\text{maximize}}\{\mathbf{v}^{(m-1)^T}\mathbf{X}^T\mathbf{X}\mathbf{v}\} \text{ subject to } ||\mathbf{v}||^2 \leq 1, ||\mathbf{v}||_1 \leq c. \tag{3.24}$$

We can apply Proposition 2.1.1 to (3.24), as follows:

The solution $\mathbf{v}^{(m)}$ *to (3.24) equals* $\frac{S(\mathbf{X}^T\mathbf{X}\mathbf{v}^{(m-1)},\Delta)}{||S(\mathbf{X}^T\mathbf{X}\mathbf{v}^{(m-1)},\Delta)||_2}$ *where* $\Delta = 0$ *if this results in* $||\mathbf{v}^{(m)}||_1 \leq c$*; otherwise,* Δ *is a nonnegative constant chosen so that* $||\mathbf{v}^{(m)}||_1 = c$*.*

Indeed, comparing Algorithms 3.1 and 3.3, we see that the two are equivalent. It follows that Algorithm 3.1 can be interpreted as a minorization algorithm for the SCOTLASS problem.

Note that another minorizer of $f(\mathbf{v}) = \mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}$ at $\mathbf{v}^{(m)}$ is

$$g(\mathbf{v}, \mathbf{v}^{(m)}) = 2\mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}^{(m)} - \mathbf{v}^{(m)^T}\mathbf{X}^T\mathbf{X}\mathbf{v}^{(m)}. \tag{3.25}$$

To see why g minorizes f, notice that since f is convex, a first order Taylor approximation to f at a point $\mathbf{v}^{(m)}$ lies below the function f. That is,

$$\begin{aligned} f(\mathbf{v}) &\geq f(\mathbf{v}^{(m)}) + (\mathbf{v} - \mathbf{v}^{(m)})^T\nabla f(\mathbf{v}^{(m)}) \\ &= \mathbf{v}^{(m)^T}\mathbf{X}^T\mathbf{X}\mathbf{v}^{(m)} + 2(\mathbf{v} - \mathbf{v}^{(m)})^T\mathbf{X}^T\mathbf{X}\mathbf{v}^{(m)} \\ &= g(\mathbf{v}, \mathbf{v}^{(m)}). \end{aligned} \tag{3.26}$$

And by inspection, $f(\mathbf{v}^{(m)}) = g(\mathbf{v}^{(m)}, \mathbf{v}^{(m)})$. The minorization algorithm based on the

minorizer (3.25) is also equivalent to Algorithm 3.1.

Chapter 4

Sparse canonical correlation analysis

In this chapter, we show that the PMD can be used to develop a method for sparse canonical correlation analysis. This chapter is closely related to material that appears in Witten et al. (2009) and Witten & Tibshirani (2009).

4.1 Canonical correlation analysis and high-dimensional data

Canonical correlation analysis (CCA), due to Hotelling (1936), is a classical method for determining the relationship between two sets of variables. Given two data sets $\mathbf{X}_1$ and $\mathbf{X}_2$ of dimensions $n \times p_1$ and $n \times p_2$ on the same set of n observations, CCA seeks linear combinations of the variables in $\mathbf{X}_1$ and the variables in $\mathbf{X}_2$ that are maximally correlated with each other. That is, $\mathbf{w}_1 \in \mathbb{R}^{p_1}$ and $\mathbf{w}_2 \in \mathbb{R}^{p_2}$ solve the *CCA criterion*, given by

$$\underset{\mathbf{w}_1,\mathbf{w}_2}{\text{maximize}}\{\mathbf{w}_1^T\mathbf{X}_1^T\mathbf{X}_2\mathbf{w}_2\} \text{ subject to } \mathbf{w}_1^T\mathbf{X}_1^T\mathbf{X}_1\mathbf{w}_1 = \mathbf{w}_2^T\mathbf{X}_2^T\mathbf{X}_2\mathbf{w}_2 = 1, \tag{4.1}$$

where we assume that the columns of $\mathbf{X}_1$ and $\mathbf{X}_2$ have been standardized to have mean zero and standard deviation one. In this chapter, we will refer to $\mathbf{w}_1$ and $\mathbf{w}_2$ as the canonical vectors (or weights), and we will refer to $\mathbf{X}_1\mathbf{w}_1$ and $\mathbf{X}_2\mathbf{w}_2$ as the canonical variables.

In recent years, it has become commonplace for biomedical researchers to perform multiple assays on the same set of patient samples; for instance, DNA copy number, gene expression, and single nucleotide polymorphism data might all be available. Examples of studies involving two or more genomic assays on the same set of samples include Hyman et al. (2002), Pollack et al. (2002), Morley et al. (2004), Stranger et al. (2005), and Stranger et al. (2007). In the case of, say, DNA copy number and gene expression measurements on a single set of patient samples, one might wish to perform CCA in order to identify genes whose expression is correlated with regions of genomic gain or loss. However, genomic data is characterized by the fact that the number of features generally greatly exceeds the number of observations. For this reason, CCA cannot be applied directly. In this chapter, we propose an extension of CCA that is applicable to the high-dimensional setting and that results in interpretable canonical vectors.

4.2 A proposal for sparse canonical correlation analysis

4.2.1 The sparse CCA method

Using a simple extension of the PMD criterion (2.7), we can extend CCA to the high-dimensional setting in such a way that the resulting canonical vectors are interpretable. Consider (2.7) with the matrix $\mathbf{X}$ replaced with the matrix $\mathbf{X}_1^T\mathbf{X}_2$. This results in the criterion

$$\begin{aligned} &\underset{\mathbf{w}_1,\mathbf{w}_2}{\text{maximize}}\{\mathbf{w}_1^T\mathbf{X}_1^T\mathbf{X}_2\mathbf{w}_2\} \\ &\text{subject to } ||\mathbf{w}_1||^2 \leq 1, ||\mathbf{w}_2||^2 \leq 1, P_1(\mathbf{w}_1) \leq c_1, P_2(\mathbf{w}_2) \leq c_2 \end{aligned} \tag{4.2}$$

where P_1 and P_2 are convex penalty functions. Since P_1 and P_2 are generally chosen to yield $\mathbf{w}_1$ and $\mathbf{w}_2$ sparse, we call this the *sparse CCA criterion*. This criterion follows from the CCA criterion (4.1) by applying penalties to $\mathbf{w}_1$ and $\mathbf{w}_2$ and also assuming that the covariance matrix of the features is diagonal; that is, we replace $\mathbf{w}_1^T\mathbf{X}_1^T\mathbf{X}_1\mathbf{w}_1$ and $\mathbf{w}_2^T\mathbf{X}_2^T\mathbf{X}_2\mathbf{w}_2$ in (4.1) with $\mathbf{w}_1^T\mathbf{w}_1$ and $\mathbf{w}_2^T\mathbf{w}_2$.

The specific forms of the convex penalty functions P_1 and P_2 should be chosen based on the data under consideration. For instance, if $\mathbf{X}_1$ is a gene expression data set, then we might want $\mathbf{w}_1$ to be sparse. In this case, using an L_1 penalty for P_1 is appropriate. If $\mathbf{X}_2$ is a DNA copy number data set, then we might wish to obtain a weight vector $\mathbf{w}_2$ that is sparse and piecewise constant. In this case, P_2 could be a fused lasso penalty. In order to indicate the form of the penalties P_1 and P_2 in use, we will refer to the method as "sparse CCA(P_1,P_2)". That is, if both penalties are L_1, then we will call this "sparse CCA(L_1,L_1)", and if P_1 is an L_1 penalty and P_2 a fused lasso penalty, then we will call it "sparse CCA(L_1,FL)" (where "FL" indicates the fused lasso).

To optimize (4.2), one can simply apply Algorithm 2.1 in Chapter 2.1 with $\mathbf{X}$ replaced with $\mathbf{X}_1^T\mathbf{X}_2$. That is, the first pair of sparse canonical vectors is computed as follows:

Algorithm 4.1: Computation of first sparse CCA canonical vectors

1. Initialize $\mathbf{w}_2$ to have L_2 norm 1.

2. Iterate:

   (a) Let $\mathbf{w}_1$ solve

$$\underset{\mathbf{w}_1}{\text{maximize}}\{\mathbf{w}_1^T\mathbf{X}_1^T\mathbf{X}_2\mathbf{w}_2\} \text{ subject to } ||\mathbf{w}_1||^2 \leq 1, P_1(\mathbf{w}_1) \leq c_1. \qquad (4.3)$$

(b) Let $\mathbf{w}_2$ solve

$$\underset{\mathbf{w}_2}{\text{maximize}}\{\mathbf{w}_1^T\mathbf{X}_1^T\mathbf{X}_2\mathbf{w}_2\} \text{ subject to } ||\mathbf{w}_2||^2 \leq 1, P_2(\mathbf{w}_2) \leq c_2. \quad (4.4)$$

Methods for selecting tuning parameter values and assessing significance of the resulting canonical vectors are presented in Chapter 4.5. To obtain multiple canonical vectors, one can apply Algorithm 2.2; more details are given in Chapter 4.6. However, to simplify interpretation of the examples presented in this chapter, we will consider only the first canonical vectors $\mathbf{w}_1$ and $\mathbf{w}_2$, as given in the criterion (4.2).

4.2.2 Sparse CCA with nonnegative weights

The sparse CCA method will result in canonical vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ that are sparse, if the penalties P_1 and P_2 are chosen appropriately. However, the nonzero elements of $\mathbf{w}_1$ and $\mathbf{w}_2$ may be of different signs. In some cases, for the sake of interpretation, one might seek a sparse weighted average of the features in $\mathbf{X}_1$ that is correlated with a sparse weighted average of the features in $\mathbf{X}_2$. Then one will want to additionally restrict the elements of $\mathbf{w}_1$ and $\mathbf{w}_2$ to be nonnegative (or nonpositive). If we require the elements of $\mathbf{w}_1$ and $\mathbf{w}_2$ to be nonnegative, the sparse CCA criterion (4.2) becomes

$$\begin{aligned}&\underset{\mathbf{w}_1,\mathbf{w}_2}{\text{maximize}}\{\mathbf{w}_1^T\mathbf{X}_1^T\mathbf{X}_2\mathbf{w}_2\}\\ &\text{subject to } ||\mathbf{w}_1||^2 \leq 1, ||\mathbf{w}_2||^2 \leq 1, P_1(\mathbf{w}_1) \leq c_1, P_2(\mathbf{w}_2) \leq c_2, w_{1j} \geq 0, w_{2j} \geq 0 \; \forall j, \end{aligned} \quad (4.5)$$

and the resulting algorithm is as follows:

Algorithm 4.2: Sparse CCA with nonnegative weights

1. Initialize $\mathbf{w}_2$ to have L_2 norm 1.

2. Iterate:

(a) Let $\mathbf{w}_1$ solve

$$\underset{\mathbf{w}_1}{\text{maximize}}\{\mathbf{w}_1^T\mathbf{X}_1^T\mathbf{X}_2\mathbf{w}_2\}$$
$$\text{subject to } ||\mathbf{w}_1||^2 \leq 1, P_1(\mathbf{w}_1) \leq c_1, w_{1j} \geq 0 \; \forall j. \tag{4.6}$$

(b) Let $\mathbf{w}_2$ solve

$$\underset{\mathbf{w}_2}{\text{maximize}}\{\mathbf{w}_1^T\mathbf{X}_1^T\mathbf{X}_2\mathbf{w}_2\}$$
$$\text{subject to } ||\mathbf{w}_2||^2 \leq 1, P_2(\mathbf{w}_2) \leq c_2, w_{2j} \geq 0 \; \forall j. \tag{4.7}$$

Letting $\mathbf{a} = \mathbf{X}_2^T\mathbf{X}_1\mathbf{w}_1$, we can rewrite (4.7) as

$$\underset{\mathbf{w}_2}{\text{maximize}}\{\mathbf{a}^T\mathbf{w}_2\} \text{ subject to } ||\mathbf{w}_2||^2 \leq 1, P_2(\mathbf{w}_2) \leq c_2, w_{2j} \geq 0 \; \forall j. \tag{4.8}$$

Suppose that P_2 is an L_1 penalty. It is clear that if $a_j \leq 0$, then $w_{2j} = 0$. So using arguments from Chapter 2.7.2, one can show that when $1 \leq c_2 \leq \sqrt{p_2}$ and the maximal element of $\mathbf{a}$ is unique, the solution to (4.8) is

$$\mathbf{w}_2 = \frac{S(\mathbf{a}_+, \Delta)}{||S(\mathbf{a}_+, \Delta)||_2}, \tag{4.9}$$

where $\Delta = 0$ if this results in $||\mathbf{w}_2||_1 \leq c_2$; otherwise, $\Delta > 0$ is chosen so that $||\mathbf{w}_2||_1 = c_2$. Here, $x_+ = \max(x, 0)$, where this operation is applied componentwise to a vector. An analogous update step can be derived for $\mathbf{w}_1$ if P_1 is an L_1 penalty.

4.2.3 Example: Sparse CCA applied to DLBCL data

We apply sparse CCA to the lymphoma data set of Lenz et al. (2008), which consists of gene expression and array CGH measurements on 203 patients with diffuse large B-cell

lymphoma (DLBCL). The data set is publicly available at
`http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE11318`. We limited the analysis to genes for which we knew the chromosomal location, and we averaged expression measurements for genes for which multiple measurements were available. This resulted in a total of 17350 gene expression measurements and 386165 copy number measurements. For computational reasons, sets of adjacent CGH spots on each chromosome were averaged before all analyses were performed. In previous research, gene expression profiling has been used to define three subtypes of DLBCL, called germinal center B-cell-like (GCB), activated B-cell-like (ABC), and primary mediastinal B-cell lymphoma (PMBL) (Alizadeh et al. 2000, Rosenwald et al. 2002). For most of the 203 observations, survival time and DLBCL subtype are known.

We performed sparse CCA(L_1,FL) using $\mathbf{X}_1$ equal to expression data of genes on all chromosomes and $\mathbf{X}_2$ equal to CGH data on chromosome i. Tuning parameter values were chosen by permutations; details are given in Chapter 4.5. P-values obtained using the method in Chapter 4.5, as well as the chromosomes on which the genes corresponding to nonzero $\mathbf{w}_1$ weights are located, can be found in Table 4.1. Canonical vectors found on almost all chromosomes were significant, and for the most part, *cis* interactions were found. Cis interactions are those for which the regions of DNA copy number change and the sets of genes with correlated expression are located on the same chromosome. The presence of cis interactions is not surprising because copy number gain on a given chromosome could naturally result in increased expression of the genes that were gained, and similarly copy number loss could result in decreased gene expression.

To assess the biological importance of the canonical vectors found, we used the CGH and expression canonical variables, both of which are vectors in $\mathbb{R}^n$, as features in a multivariate Cox proportional hazards model to predict survival. We also used the canonical variables as features in a multinomial logistic regression to predict cancer subtype. The resulting p-values are shown in Table 4.1. The Cox proportional hazards models predicting survival

Chr.	P-Value	Chr. of Genes w/Nonzero Weights	P-Value w/Surv.	P-Value w/Subtype
1	0	1 1 1 1 1 1	0.009446	0.000395
2	0	2 2	0.142911	0.000352
3	0	3 3	0.00031	0
4	0	11 4 4 4 4 4 4 4	0.803672	0.111732
5	0	5 5	0.688596	0.034906
6	0	6 6 6 6 6 6 6	0.746287	0.000214
7	0	7 7	0.507885	2e-06
8	0	8 8 8 8 8	0.080686	1.2e-05
9	0	9 9	0.729718	0
10	0	10 10 10 10 10	0.066309	3e-06
11	0	11 11 11 11 11 11	0.038497	3e-06
12	0	12 12 12 12 12 12 12 12	0.186285	0
13	0	13 13	0.337291	0.000969
14	0.05	14 14	0.024711	0
15	0	15 15 15 15 15	0.018201	0.003303
16	0	16 16	0.060006	0.004777
17	0	17 17 17 17 17 17	0.029704	0.800293
18	0	18 18 18 18 18	0.006116	0
19	0	19 19	0.059882	0
20	0	20 20 2 3 20 20	0.909788	0.005293
21	0	21 21 21 21 21 21 21 21	0.246844	0.007996
22	0	22 1	0.588148	0.004283

Table 4.1: **Column 1**: Sparse CCA was performed using all gene expression measurements, and CGH data from chromosome i only. **Column 2**: In almost every case, the canonical vectors found were highly significant. **Column 3**: CGH measurements on chromosome i were found to be correlated with the expression of sets of genes on chromosome i. **Columns 4 and 5**: P-values are reported for the Cox proportional hazards and multinomial logistic regression models that use the canonical variables to predict survival and cancer subtype.

from the canonical variables were not significant on most chromosomes. However, on many chromosomes, the canonical variables were highly predictive of DLBCL subtype. This is not surprising, since the subtypes are defined using gene expression, and it was found in Lenz et al. (2008) that the subtypes are characterized by regions of copy number change. Boxplots showing the canonical variables as a function of DLBCL subtype are displayed in Figure 4.1 for chromosomes 6 and 9. For chromosome 9, Figure 4.2 shows $\mathbf{w}_2$, the canonical vector corresponding to copy number, as well as the raw copy number for the samples with largest and smallest absolute value in the canonical variable for the CGH data.

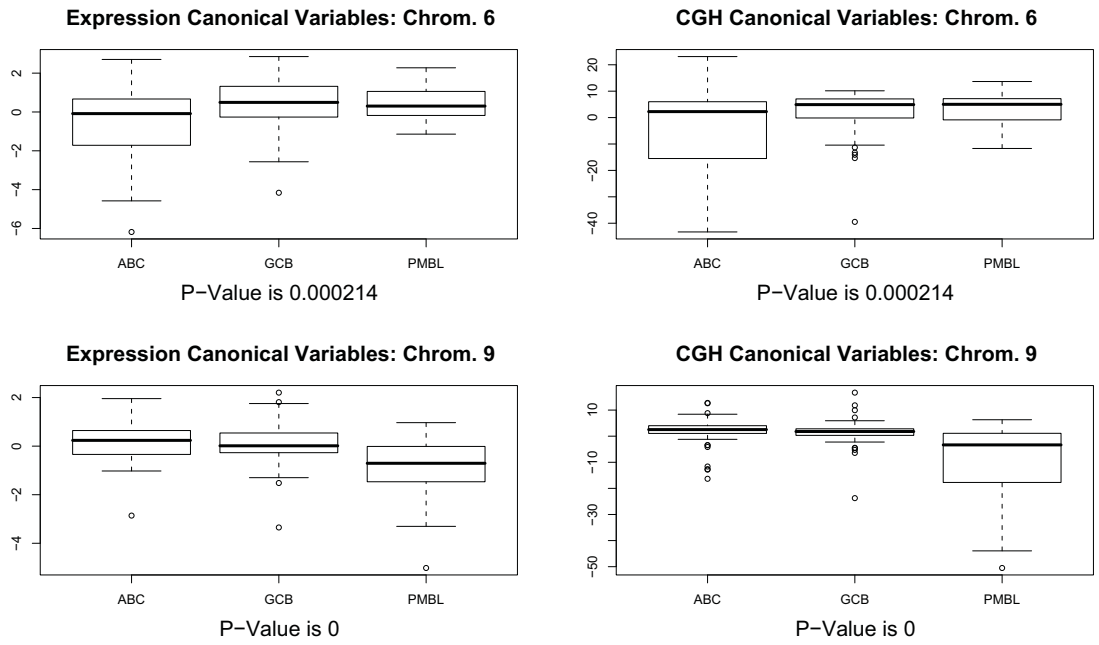

Figure 4.1: Sparse CCA was performed using CGH data on a single chromosome and all gene expression measurements. For chromosomes 6 and 9, the gene expression and CGH canonical variables, stratified by cancer subtype, are shown. P-values reported are replicated from Table 4.1; they reflect the extent to which the canonical variables predict cancer subtype in a multinomial logistic regression model.

We also compare the sparse CCA canonical variables obtained on the DLBCL data to the first principal components obtained if PCA is performed separately on the expression data and on the CGH data. PCA and sparse CCA were performed using all of the gene

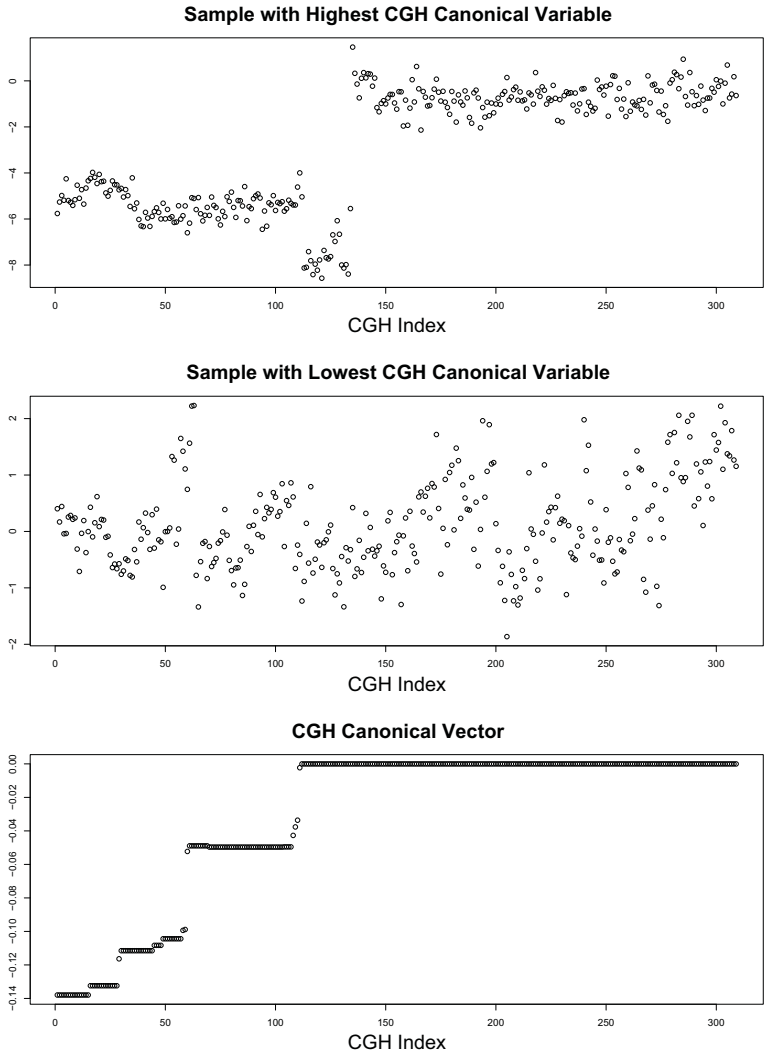

Figure 4.2: Sparse CCA was performed using CGH data on chromosome 9, and all gene expression measurements. The samples with the highest and lowest absolute values in the CGH canonical variable are shown, along with the canonical vector corresponding to the CGH data.

expression data, and the CGH data on chromosome 3. Figure 4.3 shows the resulting canonical variables and principal components. Sparse CCA results in CGH and expression canonical variables that are highly correlated with each other, due to the form of the sparse CCA criterion (4.2). PCA results in principal components that are far less correlated with each other, and consequently may yield better separation between the three subtypes. But PCA does not allow for an integrated interpretation of the expression and CGH data together.

In this section, we assessed the association between the canonical variables found using sparse CCA and the clinical outcomes in order to determine if the results of sparse CCA have biological significance. However, in general, if measurements are available for a clinical outcome of interest, then the sparse sCCA approach of Chapter 4.4 may be appropriate.

4.2.4 Connections with other sparse CCA proposals

In the literature, a number of proposals have been made for performing sparse CCA on high-dimensional data. We briefly review some of those methods here.

Waaijenborg et al. (2008) recast classical CCA as an iterative regression procedure, and then apply an elastic net penalty to the canonical vectors. An approximation of the iterative elastic net procedure results in an algorithm that is similar to our sparse CCA(L_1,L_1) algorithm. However, Waaijenborg et al. (2008) do not appear to be exactly optimizing a criterion.

Parkhomenko et al. (2009) develop an iterative algorithm for estimating the singular vectors of $\mathbf{X}_1^T\mathbf{X}_2$. At each step, they regularize the estimates of the singular vectors by soft-thresholding. Though they do not explicity state a criterion, it appears that they are approximately optimizing a criterion that is related to (4.2) with L_1 penalties. However, they use the Lagrange form, rather than the bound form, of the constraints on $\mathbf{w}_1$ and $\mathbf{w}_2$. Their algorithm is closely related to our sparse CCA algorithm, though they perform extra normalization steps due to computational problems with the Lagrange form of the

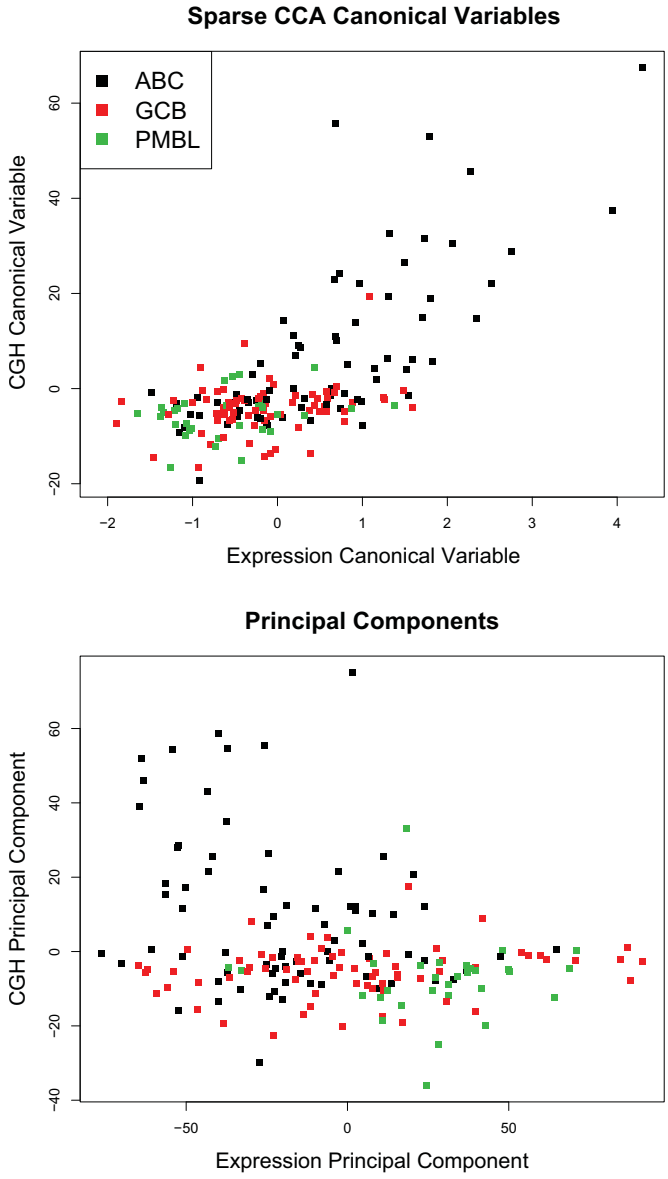

Figure 4.3: Sparse CCA and PCA were performed using CGH data on chromosome 3, and all gene expression measurements.

constraints.

The proposal of Le Cao et al. (2008) and Le Cao et al. (2009) is also closely related to our sparse CCA algorithm, though they use the Lagrange form rather than the bound form of the L_1 penalties on the canonical vectors, and the criterion that they are optimizing is somewhat less natural than (4.2).

Our sparse CCA proposal has the advantage that it results from a natural criterion that can be efficiently optimized. Moreover, it allows for the application of general convex penalties to the canonical vectors.

4.2.5 Connection with nearest shrunken centroids

Consider now a new setting in which we have n observations on p features, and each observation belongs to one of two classes. Let $\mathbf{X}_1$ denote the $n \times p$ matrix of observations by features, and let $\mathbf{X}_2$ be a binary $n \times 1$ matrix indicating class membership of each observation of $\mathbf{X}_1$. In this section, we will show that sparse CCA applied to $\mathbf{X}_1$ and $\mathbf{X}_2$ yields a canonical vector $\mathbf{w}_1$ that is closely related to the nearest shrunken centroids solution (NSC; Tibshirani et al. 2002, Tibshirani et al. 2003).

Assume that each column of $\mathbf{X}_1$ has been standardized to have mean zero and pooled within-class standard deviation equal to one. NSC is a high-dimensional classification method that involves defining "shrunken" class centroids based on only a subset of the features. We first explain the NSC method, applied to data $\mathbf{X}_1$. For class k, the *shrunken centroid* is a vector in $\mathbb{R}^p$ defined as

$$\overline{\mathbf{X}}'_{1k} = m_k S\left(\frac{\overline{\mathbf{X}}_{1k}}{m_k}, \delta\right). \tag{4.10}$$

Here, $\overline{\mathbf{X}}_{1k} \in \mathbb{R}^p$ is the mean vector for the observations in class k, Δ is a nonnegative tuning parameter, and $m_k = \sqrt{\frac{1}{n_k} - \frac{1}{n}}$ where n_k is the number of observations in class k. The NSC classification rule then assigns a new observation to the class whose shrunken centroid

(4.10) is nearest, in terms of Euclidean distance.

Now, rescale the elements of $\mathbf{X}_2$ so that the class 1 values are $\frac{1}{n_1}$ and the class 2 values are $-\frac{1}{n_2}$, and consider the effect of applying sparse CCA with L_1 penalties to matrices $\mathbf{X}_1$ and $\mathbf{X}_2$, where $\mathbf{X}_2$ is considered to be a $n \times 1$ matrix. Examining the criterion

$$\begin{aligned} &\underset{\mathbf{w}_1,\mathbf{w}_2}{\text{maximize}}\{\mathbf{w}_1^T\mathbf{X}_1^T\mathbf{X}_2\mathbf{w}_2\} \\ &\text{subject to } ||\mathbf{w}_1||^2 \leq 1, ||\mathbf{w}_2||^2 \leq 1, ||\mathbf{w}_1||_1 \leq c_1, ||\mathbf{w}_2||_1 \leq c_2, \end{aligned} \tag{4.11}$$

it is clear that since $\mathbf{w}_2 \in \mathbb{R}^1$, (4.11) reduces to

$$\underset{\mathbf{w}_1}{\text{maximize}}\{\mathbf{w}_1^T\mathbf{X}_1^T\mathbf{X}_2\} \text{ subject to } ||\mathbf{w}_1||^2 \leq 1, ||\mathbf{w}_1||_1 \leq c_1, \tag{4.12}$$

which can be rewritten as

$$\underset{\mathbf{w}_1}{\text{maximize}}\{(\overline{\mathbf{X}}_{11} - \overline{\mathbf{X}}_{12})^T\mathbf{w}_1\} \text{ subject to } ||\mathbf{w}_1||^2 \leq 1, ||\mathbf{w}_1||_1 \leq c_1. \tag{4.13}$$

By Proposition 2.3.1, the solution to (4.13) is

$$\mathbf{w}_1 = \frac{S(\overline{\mathbf{X}}_{11} - \overline{\mathbf{X}}_{12}, \Delta)}{||S(\overline{\mathbf{X}}_{11} - \overline{\mathbf{X}}_{12}, \Delta)||_2} = \frac{S((1 + \frac{n_1}{n_2})\overline{\mathbf{X}}_{11}, \Delta)}{||S((1 + \frac{n_1}{n_2})\overline{\mathbf{X}}_{11}, \Delta)||_2} \tag{4.14}$$

where $\Delta = 0$ if that results in $||\mathbf{w}_1||_1 \leq c_1$; otherwise, $\Delta > 0$ is chosen so that $||\mathbf{w}_1||_1 = c_1$. So sparse CCA yields a canonical vector that is proportional to the shrunken centroid $\overline{\mathbf{X}}'_{11}$ (4.10) when the tuning parameters for NSC and sparse CCA are chosen appropriately.

4.3 Sparse multiple CCA

4.3.1 The sparse multiple CCA method

CCA and sparse CCA can be used to perform an integrative analysis of two data sets with features on a single set of samples. But what if more than two such data sets are available? If K data sets are available, then one might wish to identify K linear combinations of variables - one for each data set - such that each pair among the K linear combinations has a high level of correlation. A number of approaches for generalizing CCA to more than two data sets have been proposed in the literature, and some of these extensions are summarized in Gifi (1990). We will focus on one of these proposals for multiple-set CCA.

Let the K data sets be denoted $\mathbf{X}_1, \ldots, \mathbf{X}_K$; data set k is of dimension $n \times p_k$, and each variable has mean zero and standard deviation one. Then, the single-factor multiple-set CCA criterion is

$$\underset{\mathbf{w}_1,\ldots,\mathbf{w}_K}{\text{maximize}}\{\sum_{i<j} \mathbf{w}_i^T \mathbf{X}_i^T \mathbf{X}_j \mathbf{w}_j\} \text{ subject to } \mathbf{w}_k^T \mathbf{X}_k^T \mathbf{X}_k \mathbf{w}_k = 1 \; \forall k, \tag{4.15}$$

where $\mathbf{w}_k \in \mathbb{R}^{p_k}$. It is easy to see that when $K = 2$, then multiple-set CCA simplifies to ordinary CCA. We can develop a method for sparse multiple CCA by imposing sparsity constraints on this natural formulation for multiple-set CCA. We also assume that the features are independent within each data set: that is, $\mathbf{X}_k^T \mathbf{X}_k = \mathbf{I}$ for each k. Then, *sparse multiple CCA* (sparse mCCA) solves the following problem:

$$\underset{\mathbf{w}_1,\ldots,\mathbf{w}_K}{\text{maximize}}\{\sum_{i<j} \mathbf{w}_i^T \mathbf{X}_i^T \mathbf{X}_j \mathbf{w}_j\} \text{ subject to } ||\mathbf{w}_k||^2 \leq 1, P_k(\mathbf{w}_k) \leq c_k \; \forall k, \tag{4.16}$$

where P_k are convex penalty functions. Then, $\mathbf{w}_k$ is the canonical vector associated with $\mathbf{X}_k$. If P_k is an L_1 or fused lasso penalty and c_k is chosen appropriately, then $\mathbf{w}_k$ will be sparse.

It is not hard to see that just as (4.2) is biconvex in $\mathbf{w}_1$ and $\mathbf{w}_2$, (4.16) is *multiconvex* in $\mathbf{w}_1, \ldots, \mathbf{w}_K$. That is, with $\mathbf{w}_j$ held fixed for all $j \neq k$, (4.16) is convex in $\mathbf{w}_k$. This suggests an iterative algorithm that increases the objective function of (4.16) at each iteration.

Algorithm 4.3: Computation of first sparse mCCA component

1. For each k, fix an initial value of $\mathbf{w}_k \in \mathbb{R}^{p_k}$ such that $||\mathbf{w}_k||^2 = 1$.

2. Iterate: For each k, let $\mathbf{w}_k$ be the solution to

$$\underset{\mathbf{w}_k}{\text{maximize}}\{\sum_{i \neq k} \mathbf{w}_i^T \mathbf{X}_i^T \mathbf{X}_k \mathbf{w}_k\} \text{ subject to } ||\mathbf{w}_k||^2 \leq 1, P_k(\mathbf{w}_k) \leq c_k. \tag{4.17}$$

For instance, if P_k is an L_1 penalty, then by Proposition 2.3.1, the update takes the form

$$\mathbf{w}_k = \frac{S(\mathbf{X}_k^T(\sum_{i \neq k} \mathbf{X}_i \mathbf{w}_i), \Delta_k)}{||S(\mathbf{X}_k^T(\sum_{i \neq k} \mathbf{X}_i \mathbf{w}_i), \Delta_k)||_2}, \tag{4.18}$$

where $\Delta_k = 0$ if this results in $||\mathbf{w}_k||_1 \leq c_k$; otherwise, $\Delta_k > 0$ is chosen such that $||\mathbf{w}_k||_1 = c_k$.

We demonstrate the performance of sparse mCCA on a simple simulated example. Data were generated according to the model

$$\mathbf{X}_k = \mathbf{u}\mathbf{w}_k^T + \boldsymbol{\epsilon}_k, 1 \leq k \leq 3 \tag{4.19}$$

where $\mathbf{u} \in \mathbb{R}^{50}$, $\mathbf{w}_1 \in \mathbb{R}^{100}$, $\mathbf{w}_2 \in \mathbb{R}^{200}$, $\mathbf{w}_3 \in \mathbb{R}^{300}$. Only the first 20, 40, and 60 elements of $\mathbf{w}_1$, $\mathbf{w}_2$, and $\mathbf{w}_3$ were nonzero, respectively. Sparse mCCA(L_1,L_1) was run on this data, and the resulting estimates of $\mathbf{w}_1$, $\mathbf{w}_2$, and $\mathbf{w}_3$ are shown in Figure 4.4.

A permutation algorithm for selecting tuning parameter values and assessing significance of sparse mCCA can be found in Chapter 4.5. In addition, an algorithm for obtaining multiple sparse mCCA factors is given in Chapter 4.6.

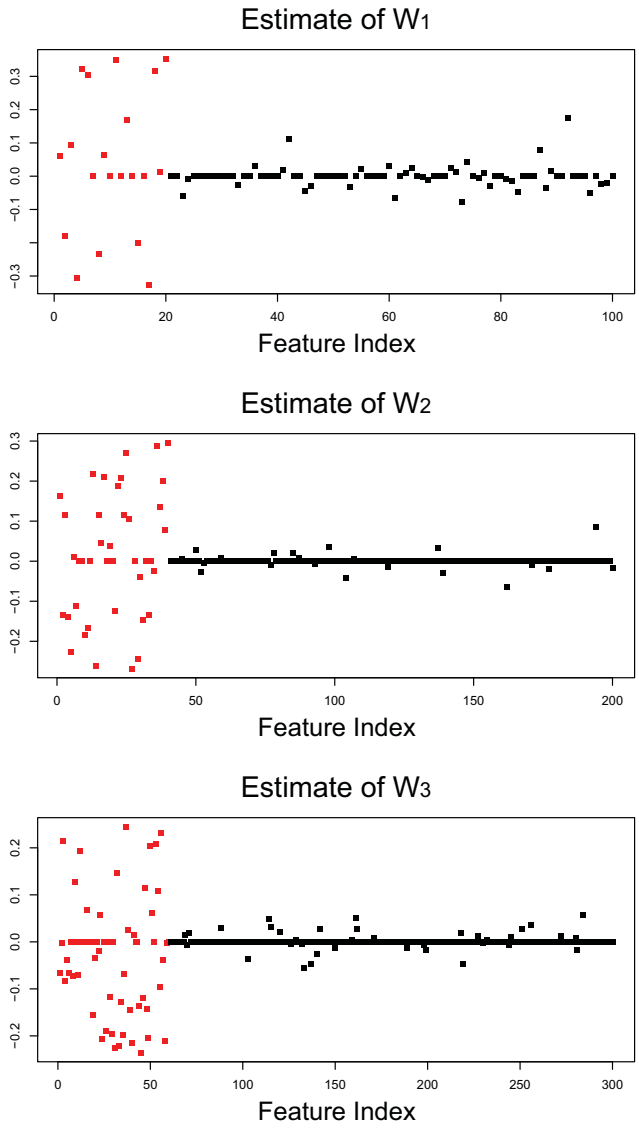

Figure 4.4: Three data sets $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{X}_3$ were generated under a simple model, and sparse mCCA was performed. The resulting estimates of $\mathbf{w}_1$, $\mathbf{w}_2$, and $\mathbf{w}_3$ are fairly accurate at distinguishing between the elements of $\mathbf{w}_i$ that are truly nonzero (red) and those that are not (black).

4.3.2 Example: Sparse mCCA applied to DLBCL CGH data

If CGH measurements are available on a set of patient samples, then one may wonder whether copy number changes in genomic regions on separate chromosomes are correlated with each other. For instance, certain genomic regions may tend to be coamplified or codeleted.

To answer this question for a single pair of chromosomes, we can perform sparse CCA(FL,FL) with two data sets, $\mathbf{X}_1$ and $\mathbf{X}_2$, where $\mathbf{X}_1$ contains the CGH measurements on the first chromosome of interest and $\mathbf{X}_2$ contains the CGH measurements on the second chromosome of interest. If copy number change on the first chromosome is independent of copy number change on the second chromosome, then we expect the corresponding p-value obtained using the method of Chapter 4.5 not to be small. A small p-value indicates that copy number changes on the two chromosomes are more correlated with each other than one would expect due to chance. However, in general, if there are K chromosomes, then there are $\binom{K}{2}$ pairs of chromosomes that can be tested for correlated patterns of amplification and deletion; this leads to a multiple testing problem and excessive computation. Instead, we take a different approach, using sparse mCCA. We apply sparse mCCA to data sets $\mathbf{X}_1, \ldots, \mathbf{X}_K$, where $\mathbf{X}_k$ contains the CGH measurements on chromosome k. A fused lasso penalty is used on each data set. The goal is to identify correlated regions of gain and loss across the entire genome.

This method is applied to the DLBCL data set mentioned previously. We first denoise the samples by applying the fused lasso to each sample individually, as in Tibshirani & Wang (2008). We then perform sparse mCCA on the resulting smoothed CGH data. The canonical vectors that result are shown in Figure 4.5. From the figure, one can conclude that complex patterns of gain and loss tend to co-occur. It is unlikely that a single sample would display the entire pattern found; however, samples with large values in the canonical variables most likely contain some of the patterns shown in the figure.

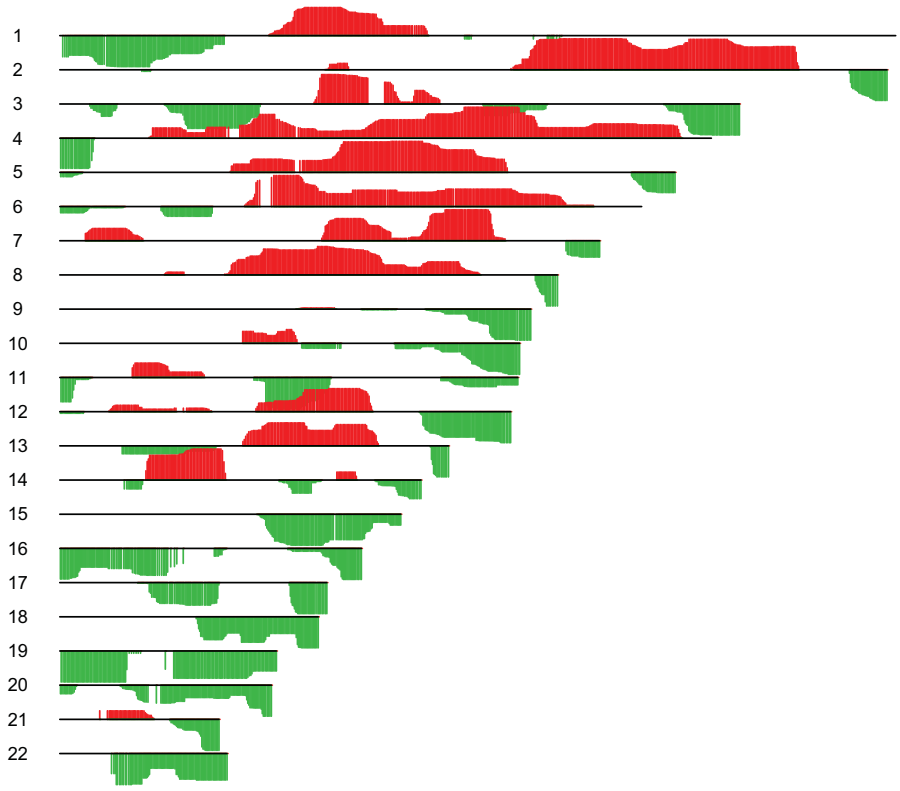

Figure 4.5: Sparse mCCA was performed on the DLBCL CGH data, treating each chromosome as a separate "data set", in order to identify genomic regions that are coamplified and/or codeleted. The canonical vectors are shown, with components ordered by chromosomal location. Positive values of the canonical vectors are shown in red, and negative values are in green.

4.4 Sparse supervised CCA

4.4.1 Supervised PCA

In Chapter 4.2.3, we determined that on the DLBCL data, many of the canonical variables obtained using sparse CCA are highly associated with tumor subtype, and some of the canonical variables are also associated with survival time. Though outcome measurements were available, we took an unsupervised approach in performing sparse CCA. We will now develop an approach to directly make use of an outcome in sparse CCA. Our proposal for *sparse supervised CCA* (sparse sCCA) is closely related to the *supervised principal components analysis* (supervised PCA) proposal of Bair & Tibshirani (2004) and Bair et al. (2006), and so we begin with an overview of supervised PCA.

Principal components regression (PCR; see e.g. Massy 1965) is a method for predicting an outcome $\mathbf{y} \in \mathbb{R}^n$ from a data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$. Assume that the columns of $\mathbf{X}$ have been standardized to have mean zero and standard deviation one. Then, PCR involves regressing $\mathbf{y}$ onto the first few columns of $\mathbf{XV}$, where $\mathbf{X} = \mathbf{UDV}^T$ is the SVD of $\mathbf{X}$. However, since $\mathbf{V}$ is estimated in an unsupervised manner, it is not guaranteed that the first few columns of $\mathbf{XV}$ will predict $\mathbf{y}$ well, even if some of the features in $\mathbf{X}$ are correlated with $\mathbf{y}$.

To remedy this problem, Bair & Tibshirani (2004) and Bair et al. (2006) propose the use of supervised PCA. Their method can be described simply:

1. On training data, the features that are most associated with the outcome $\mathbf{y}$ are identified.

2. PCR is performed using only the features identified in the previous step.

Theoretical results regarding the performance of this method under a latent variable model are presented in Bair et al. (2006).

4.4.2 The sparse supervised CCA method

We return to the notation of Chapter 4.2.1. Suppose that some outcome measurement is available for each observation; that is, we have an n-vector $\mathbf{y}$ in addition to $\mathbf{X}_1$ and $\mathbf{X}_2$. Then we might seek linear combinations of the variables in $\mathbf{X}_1$ and $\mathbf{X}_2$ that are highly correlated with each other and associated with the outcome.

We define *supervised CCA* (sCCA) as the solution to the problem

$$\begin{aligned} &\underset{\mathbf{w}_1,\mathbf{w}_2}{\text{maximize}}\{\mathbf{w}_1^T\mathbf{X}_1^T\mathbf{X}_2\mathbf{w}_2\} \\ &\text{subject to } ||\mathbf{w}_1||^2 \leq 1, ||\mathbf{w}_2||^2 \leq 1, w_{1j} = 0\ \forall j \notin Q_1, w_{2j} = 0\ \forall j \notin Q_2, \end{aligned} \tag{4.20}$$

where Q_1 is the set of features in $\mathbf{X}_1$ that are most associated with $\mathbf{y}$, and Q_2 is the set of features in $\mathbf{X}_2$ that are most associated with $\mathbf{y}$. The number of features in Q_1 and Q_2, or alternatively the association threshold for features to enter Q_1 and Q_2, can be treated as a tuning parameter or can simply be fixed. If $\mathbf{X}_1 = \mathbf{X}_2$, then the criterion (4.20) simplifies to supervised PCA. That is, the canonical vectors that solve (4.20) are equal to each other and to the first principal component of the subset of the data containing only the features that are most associated with the outcome.

Following the approach of previous chapters, sCCA can be easily extended to give *sparse sCCA*, defined as the solution to the problem

$$\begin{aligned} &\underset{\mathbf{w}_1,\mathbf{w}_2}{\text{maximize}}\{\mathbf{w}_1^T\mathbf{X}_1^T\mathbf{X}_2\mathbf{w}_2\} \\ &\text{subject to } ||\mathbf{w}_k||^2 \leq 1, P_k(\mathbf{w}_k) \leq c_k, w_{kj} = 0\ \forall j \notin Q_k, k = 1, 2, \end{aligned} \tag{4.21}$$

where as usual, P_1 and P_2 are convex penalty functions.

We have said that Q_1 and Q_2 contain the features in $\mathbf{X}_1$ and $\mathbf{X}_2$ that are most associated with the outcome $\mathbf{y}$. Naturally, the measure of association used depends on the outcome

type. Let $\mathbf{X}_{ij} \in \mathbb{R}^n$ denote feature j in data set $\mathbf{X}_i$, $i \in \{1, 2\}$ and $j \in \{1, 2, \ldots, p_i\}$. Here are a few possible outcome types and corresponding measures of association:

- If $\mathbf{y}$ is a quantitative outcome (e.g. tumor diameter) then the association of $\mathbf{y}$ with $\mathbf{X}_{ij}$ could be the standardized coefficient for the linear regression of $\mathbf{X}_{ij}$ onto $\mathbf{y}$.
- If $\mathbf{y}$ is a time to event (e.g. a possibly censored survival time), then the association of $\mathbf{y}$ with $\mathbf{X}_{ij}$ could be the score statistic for the univariate Cox proportional hazards model that uses $\mathbf{X}_{ij}$ to predict $\mathbf{y}$.
- If $\mathbf{y}$ contains binary class labels (e.g. each observation is in Class 1 or Class 2) then the association of $\mathbf{y}$ with $\mathbf{X}_{ij}$ could be a two-sample t-statistic for the extent to which $\mathbf{X}_{ij}$ differs between the two classes.
- If $\mathbf{y}$ is a multiple class outcome (e.g. each observation is in some Class k, for some k between 1 and K), then the association of $\mathbf{y}$ with $\mathbf{X}_{ij}$ could be the F-statistic for a one-way ANOVA for the extent to which $\mathbf{X}_{ij}$ differs between the K classes.

The algorithm for sparse sCCA can be written as follows:

Algorithm 4.4: Computation of first sparse sCCA component

1. Let $\tilde{\mathbf{X}}_1$ and $\tilde{\mathbf{X}}_2$ denote the submatrices of $\mathbf{X}_1$ and $\mathbf{X}_2$ consisting of the features in Q_1 and Q_2. Q_1 and Q_2 are calculated as follows:
   (a) In the case of an L_1 penalty on $\mathbf{w}_i$, Q_i is the set of indices of the features in $\mathbf{X}_i$ that have highest association with the outcome.
   (b) In the case of a fused lasso penalty on $\mathbf{w}_i$, the vector of associations between the features in $\mathbf{X}_i$ and the outcome is smoothed using the fused lasso. The resulting smoothed vector is thresholded to obtain the desired number of nonzero cofficients. Q_i contains the indices of the coefficients that are nonzero after thresholding.

2. Perform sparse CCA by applying Algorithm 4.1 in Chapter 4.2.1 to data matrices $\tilde{\mathbf{X}}_1$ and $\tilde{\mathbf{X}}_2$.

Note that the fused lasso case is treated specially because one wishes for the features included in $\tilde{\mathbf{X}}_i$ to be contiguous, so that smoothness in the resulting $\mathbf{w}_i$ weights will translate to smoothness in the weights of the original variable set. Chapter 4.5 contains algorithms for tuning parameter selection and assessment of significance, and Chapter 4.6 contains a method for obtaining multiple canonical vectors.

We explore the performance of sparse sCCA with a quantitative outcome on a toy example. Data are generated according to the model

$$\mathbf{X}_1 = \mathbf{u}\mathbf{w}_1^T + \boldsymbol{\epsilon}_1, \quad \mathbf{X}_2 = \mathbf{u}\mathbf{w}_2^T + \boldsymbol{\epsilon}_2, \quad \mathbf{y} = \mathbf{u}, \tag{4.22}$$

with $\mathbf{u} \in \mathbb{R}^{50}$, $\mathbf{w}_1 \in \mathbb{R}^{500}$, $\mathbf{w}_2 \in \mathbb{R}^{1000}$, $\boldsymbol{\epsilon}_1 \in \mathbb{R}^{50\times 500}$, $\boldsymbol{\epsilon}_2 \in \mathbb{R}^{50\times 1000}$. 50 elements of $\mathbf{w}_1$ and 100 elements of $\mathbf{w}_2$ are nonzero. The first canonical vectors of sparse CCA(L_1,L_1) and sparse sCCA(L_1,L_1) were computed for a range of values of c_1 and c_2. In Figure 4.6, the resulting number of true positives (features that are nonzero in $\mathbf{w}_1$ and $\mathbf{w}_2$ and also in the estimated canonical vectors) are shown on the y-axis, as a function of the number of nonzero elements of the canonical vectors. It is clear that sparse sCCA results in more true positives than does sparse CCA. In Figure 4.7, the canonical variables obtained using sparse CCA and sparse sCCA are plotted against the outcome. The canonical variables obtained using sparse sCCA are highly correlated with the outcome, and those obtained using sparse CCA are less so. Note that under the model (4.22), in the absence of noise, the canonical variables are proportional to the outcome vector $\mathbf{u}$.

In theory, one could choose Q_1 and Q_2 in Step 1 of Algorithm 4.4 to contain fewer than n features; then, ordinary CCA could be performed instead of sparse CCA in Step 2. However, to avoid eliminating important features by excessive screening in Step 1, we recommend using a less stringent cutoff for Q_1 and Q_2 in Step 1, and instead performing

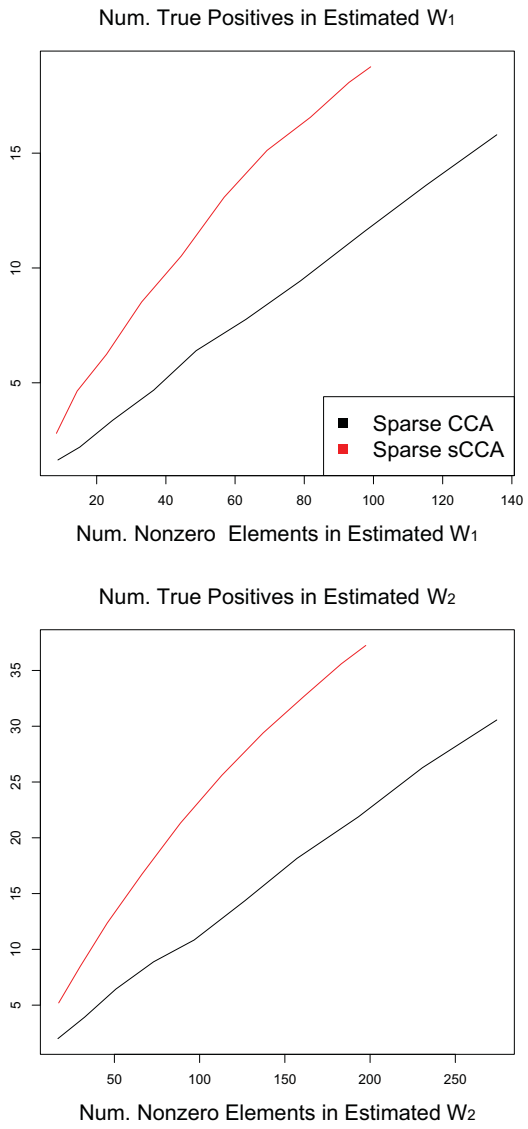

Figure 4.6: Sparse CCA(L_1,L_1) and sparse sCCA(L_1,L_1) were performed on a toy example, for a range of values of the tuning parameters in the sparse CCA criterion. The number of true positives in the estimated canonical vectors is shown as a function of the number of nonzero elements.

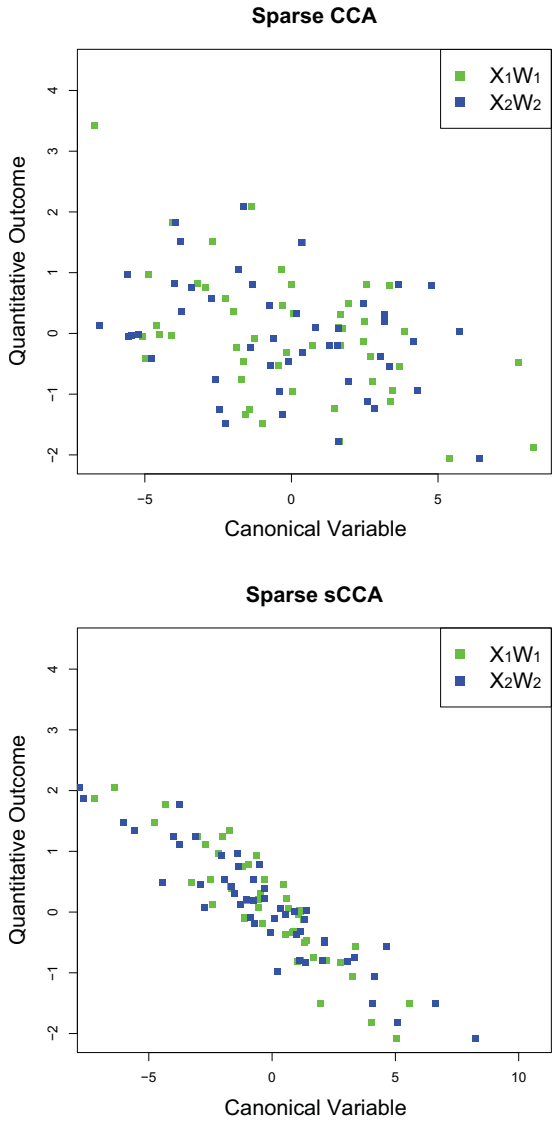

Figure 4.7: Sparse CCA(L_1,L_1) and sparse sCCA(L_1,L_1) were performed on a toy example. The canonical variables obtained using sparse sCCA are highly correlated with the outcome; those obtained using sparse CCA are not.

further feature selection in Step 2 via sparse CCA.

4.4.3 Connection with sparse mCCA

Given $\mathbf{X}_1$, $\mathbf{X}_2$, and a two-class outcome $\mathbf{y}$, one could perform sparse mCCA by treating $\mathbf{y}$ as a third data set. This would yield a different but related method for performing sparse sCCA in the case of a two-class outcome.

Note that the outcome $\mathbf{y}$ is a matrix in $\mathbb{R}^{n\times 1}$. We code the two classes (of n_1 and n_2 observations, respectively) as $\frac{\lambda}{n_1}$ and $-\frac{\lambda}{n_2}$. Assume that the columns of $\mathbf{X}_1$ and $\mathbf{X}_2$ have mean zero and pooled within-class standard deviation equal to one. Consider the sparse mCCA criterion with L_1 penalties, applied to data sets $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{y}$:

$$\begin{aligned}&\underset{\mathbf{w}_1,\mathbf{w}_2,\mathbf{w}_3}{\text{maximize}}\{\mathbf{w}_1^T\mathbf{X}_1^T\mathbf{X}_2\mathbf{w}_2 + \mathbf{w}_1^T\mathbf{X}_1^T\mathbf{y}\mathbf{w}_3 + \mathbf{w}_2^T\mathbf{X}_2^T\mathbf{y}\mathbf{w}_3\}\\ &\text{subject to } ||\mathbf{w}_i||^2 \le 1, ||\mathbf{w}_i||_1 \le c_i\ \forall i.\end{aligned} \tag{4.23}$$

Note that since $\mathbf{w}_3 \in \mathbb{R}^1$, (4.23) is equivalent to

$$\begin{aligned}&\underset{\mathbf{w}_1,\mathbf{w}_2}{\text{maximize}}\{\mathbf{w}_1^T\mathbf{X}_1^T\mathbf{X}_2\mathbf{w}_2 + \mathbf{w}_1^T\mathbf{X}_1^T\mathbf{y} + \mathbf{w}_2^T\mathbf{X}_2^T\mathbf{y}\}\\ &\text{subject to } ||\mathbf{w}_1||^2 \le 1, ||\mathbf{w}_2||^2 \le 1, ||\mathbf{w}_1||_1 \le c_1, ||\mathbf{w}_2||_1 \le c_2.\end{aligned} \tag{4.24}$$

Now, this criterion is biconvex and leads naturally to an iterative algorithm. However, this is not the approach that we take with our sparse sCCA method. Instead, notice that

$$\mathbf{w}_1^T\mathbf{X}_1^T\mathbf{y} = \lambda(\overline{\mathbf{X}}_{11} - \overline{\mathbf{X}}_{12})^T\mathbf{w}_1 = \lambda\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}\mathbf{t}_1^T\mathbf{w}_1, \tag{4.25}$$

where $\overline{\mathbf{X}}_{1k} \in \mathbb{R}^{p_1}$ is the mean vector of the observations in $\mathbf{X}_1$ that belong to class k, and where $\mathbf{t}_1 \in \mathbb{R}^{p_1}$ is the vector of two-sample t-statistics testing whether the classes defined

by $\mathbf{y}$ have equal means within each feature of $\mathbf{X}_1$. Similarly,

$$\mathbf{w}_2^T \mathbf{X}_2^T \mathbf{y} = \lambda \sqrt{\frac{1}{n_1} + \frac{1}{n_2}} \mathbf{t}_2^T \mathbf{w}_2 \tag{4.26}$$

for $\mathbf{t}_2$ defined analogously. So we can rewrite (4.24) as

$$\begin{aligned} &\underset{\mathbf{w}_1, \mathbf{w}_2}{\text{maximize}} \{ \mathbf{w}_1^T \mathbf{X}_1^T \mathbf{X}_2 \mathbf{w}_2 + \lambda \sqrt{\frac{1}{n_1} + \frac{1}{n_2}} (\mathbf{t}_1^T \mathbf{w}_1 + \mathbf{t}_2^T \mathbf{w}_2) \} \\ &\text{subject to } ||\mathbf{w}_1||^2 \leq 1, ||\mathbf{w}_2||^2 \leq 1, ||\mathbf{w}_1||_1 \leq c_1, ||\mathbf{w}_2||_1 \leq c_2. \end{aligned} \tag{4.27}$$

As λ increases, the elements of $\mathbf{w}_1$ and $\mathbf{w}_2$ that correspond to large $|\mathbf{t}_1|$ and $|\mathbf{t}_2|$ values tend to increase in absolute value relative to those that correspond to smaller $|\mathbf{t}_1|$ and $|\mathbf{t}_2|$ values.

Rather than adopting the criterion (4.27) for sparse sCCA, our sparse sCCA criterion results from assigning nonzero weights only to the elements of $\mathbf{w}_1$ and $\mathbf{w}_2$ corresponding to large $|\mathbf{t}_1|$ and $|\mathbf{t}_2|$. We prefer our proposed sparse sCCA algorithm because it is simple, generalizes to the supervised PCA method when $\mathbf{X}_1 = \mathbf{X}_2$, and extends easily to nonbinary outcomes.

4.4.4 Example: Sparse sCCA applied to DLBCL data

We evaluate the performance of sparse sCCA on the DLBCL data set, in terms of the association of the resulting canonical variables with the survival and subtype outcomes. We repeatedly split the observations into training and test sets (75% / 25%). Let $(\mathbf{X}_1^{train}, \mathbf{X}_2^{train}, \mathbf{y}^{train})$ denote the training data, and let $(\mathbf{X}_1^{test}, \mathbf{X}_2^{test}, \mathbf{y}^{test})$ denote the test data. Here, $\mathbf{y}$ can denote either the survival time or the cancer subtype. We perform sparse sCCA on the training data. As in Chapter 4.2.3, for each chromosome, sparse sCCA is run using CGH measurements on that chromosome, and all available gene expression measurements. An L_1 penalty is applied to the expression data, and a fused lasso penalty is applied to the CGH data. Let $\mathbf{w}_1^{train}, \mathbf{w}_2^{train}$ denote the canonical vectors obtained. We then use $\mathbf{X}_1^{test} \mathbf{w}_1^{train}$

and $\mathbf{X}_2^{test}\mathbf{w}_2^{train}$ as features in a Cox proportional hazards model or a multinomial logistic regression model to predict $\mathbf{y}^{test}$. The resulting p-values are shown in Figure 4.8 for both the survival and subtype outcomes; these are compared to the results obtained if the analysis is repeated using unsupervised sparse CCA on the training data. On the whole, for the subtype outcome, the p-values obtained using sparse sCCA are much smaller than those obtained using sparse CCA. The canonical variables obtained using sparse CCA and sparse sCCA with the survival outcome are not significantly associated with survival. In this example, sparse CCA was performed so that 20% of the features in $\mathbf{X}_1$ and $\mathbf{X}_2$ were contained in Q_1 and Q_2 in the sparse sCCA algorithm.

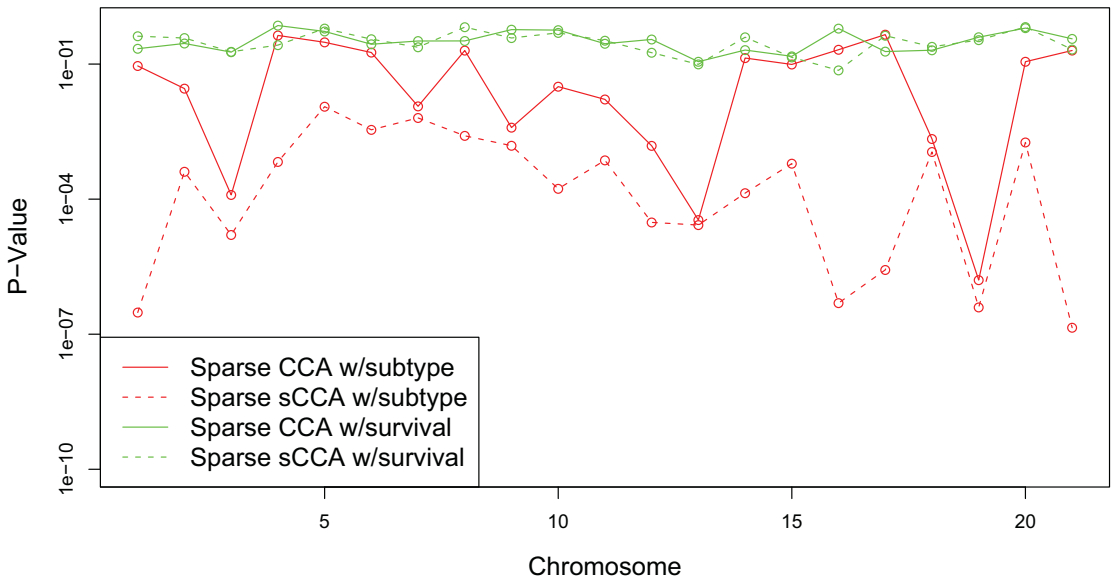

Figure 4.8: On a training set, sparse CCA and sparse sCCA were performed using CGH measurements on a single chromosome, and all available gene expression measurements. The resulting canonical vectors were used to predict survival time and DLBCL subtype on the test set. Median p-values (over training set / test set splits) are shown.

4.5 Tuning parameter selection and calculation of p-values

We now consider the problem of tuning parameter selection for sparse CCA. A number of methods have been proposed in the literature for this problem (see e.g. Waaijenborg et al. 2008, Parkhomenko et al. 2009). The method proposed here has the advantage that it does not require splitting a possibly small number of samples into a training set and a test set. Algorithm 2.5 in Chapter 2.4 can be used for tuning parameter selection for the PMD, and therefore could be used to select tuning parameters for sparse CCA, which is simply an extension of the PMD. However, as that approach requires leaving out scattered elements of the matrix $\mathbf{X}_1^T\mathbf{X}_2$, it is somewhat unnatural in this setting. Here, we take a more natural approach that also provides a measure of significance for the canonical vectors found using sparse CCA.

Algorithm 4.5: Tuning parameter selection and significance assessment for sparse CCA

1. For each tuning parameter value (generally this will be a two-dimensional vector) T_j being considered:
   (a) Apply Algorithm 4.1 in Chapter 4.2.1 to data $\mathbf{X}_1$ and $\mathbf{X}_2$ and tuning parameter T_j in order to obtain canonical vectors $\mathbf{w}_1$ and $\mathbf{w}_2$.
   (b) Compute $c_j = \text{Cor}(\mathbf{X}_1\mathbf{w}_1, \mathbf{X}_2\mathbf{w}_2)$.
   (c) For $b = 1, \ldots, B$, where B is some large number:
      i. Permute the ordering of the n rows of $\mathbf{X}_1$ to obtain the matrix $\mathbf{X}_1^b$.
      ii. Apply Algorithm 4.1 in Chapter 4.2.1 to data $\mathbf{X}_1^b$ and $\mathbf{X}_2$ and tuning parameter T_j in order to obtain canonical vectors $\mathbf{w}_1^b$ and $\mathbf{w}_2^b$.
      iii. Compute $c_j^b = \text{Cor}(\mathbf{X}_1^b\mathbf{w}_1^b, \mathbf{X}_2\mathbf{w}_2^b)$.
   (d) Calculate the p-value $p_j = \frac{1}{B}\sum_{b=1}^{B} 1_{c_j^b \geq c_j}$.

2. Choose the tuning parameter T_j corresponding to the smallest p_j. Alternatively, one can choose the tuning parameter T_j for which $(c_j - \frac{1}{B}\sum_b c_j^b)/\text{sd}(c_j^b)$ is largest, where $\text{sd}(c_j^b)$ indicates the standard deviation of $c_j^1, \ldots, c_j^B$. The resulting p-value is p_j.

Since multiple tuning parameters T_j are considered in the above algorithm, a strict cutoff for the p-value p_j should be used in assessing significance of the canonical vectors obtained, in order to avoid problems associated with multiple testing. Note that the permutations performed in Step 1(c)i do not disrupt the correlations among the features within each data set, but do disrupt the correlations among the features between the two data sets.

We can use the following permutation-based algorithm to assess the significance of the canonical vectors obtained using sparse mCCA:

Algorithm 4.6: Tuning parameter selection and significance assessment for sparse mCCA

1. For each tuning parameter (generally this will be a K-dimensional vector) T_j being considered:

   (a) Apply Algorithm 4.3 in Chapter 4.3 to data $\mathbf{X}_1, \ldots, \mathbf{X}_K$ and tuning parameter T_j in order to obtain canonical vectors $\mathbf{w}_1, \ldots, \mathbf{w}_K$.

   (b) Compute $c_j = \sum_{s<t} \text{Cor}(\mathbf{X}_s\mathbf{w}_s, \mathbf{X}_t\mathbf{w}_t)$.

   (c) For $b = 1, \ldots, B$, where B is some large number:

      i. Permute the orderings of the n rows of $\mathbf{X}_1, \ldots, \mathbf{X}_K$ separately to obtain the matrices $\mathbf{X}_1^b, \ldots, \mathbf{X}_K^b$.

      ii. Apply Algorithm 4.3 in Chapter 4.3 to data $\mathbf{X}_1^b, \ldots, \mathbf{X}_K^b$ and tuning parameter T_j in order to obtain canonical vectors $\mathbf{w}_1^b, \ldots, \mathbf{w}_K^b$.

      iii. Compute $c_j^b = \sum_{s<t} \text{Cor}(\mathbf{X}_s^b\mathbf{w}_s^b, \mathbf{X}_t^b\mathbf{w}_t^b)$.

   (d) Calculate the p-value $p_j = \frac{1}{B}\sum_{b=1}^B 1_{c_j^b \geq c_j}$.

2. Choose the tuning parameter T_j corresponding to the smallest p_j. Alternatively, one can choose the tuning parameter T_j for which $(c_j - \frac{1}{B}\sum_b c_j^b)/\text{sd}(c_j^b)$ is largest, where $\text{sd}(c_j^b)$ indicates the standard deviation of $c_j^1, \ldots, c_j^B$. The resulting p-value is p_j.

Given the above algorithms, the analogous method for selecting tuning parameters and determining significance for sparse sCCA is straightforward. For simplicity, we assume that the number of features in Q_1 and Q_2 in the sparse sCCA algorithm is fixed.

Algorithm 4.7: Tuning parameter selection and significance assessment for sparse sCCA

1. For each tuning parameter (generally this will be a two-dimensional vector) T_j being considered:
   (a) Apply Algorithm 4.4 in Chapter 4.4.2 to data $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{y}$ and tuning parameter T_j in order to obtain supervised canonical vectors $\mathbf{w}_1$ and $\mathbf{w}_2$.
   (b) Compute $c_j = \text{Cor}(\mathbf{X}_1\mathbf{w}_1, \mathbf{X}_2\mathbf{w}_2)$.
   (c) For $b = 1, \ldots, B$, where B is some large number:
      i. Permute the orderings of the n rows of $\mathbf{X}_1$ and $\mathbf{X}_2$ separately to obtain the matrices $\mathbf{X}_1^b$ and $\mathbf{X}_2^b$.
      ii. Apply Algorithm 4.4 in Chapter 4.4.2 to data $\mathbf{X}_1^b$, $\mathbf{X}_2^b$, $\mathbf{y}$, and tuning parameter T_j in order to obtain supervised canonical vectors $\mathbf{w}_1^b$ and $\mathbf{w}_2^b$.
      iii. Compute $c_j^b = \text{Cor}(\mathbf{X}_1^b\mathbf{w}_1^b, \mathbf{X}_2^b\mathbf{w}_2^b)$.
   (d) Calculate the p-value $p_j = \frac{1}{B}\sum_{b=1}^B 1_{c_j^b \geq c_j}$.
2. Choose the tuning parameter T_j corresponding to the smallest p_j. Alternatively, one can choose the tuning parameter T_j for which $(c_j - \frac{1}{B}\sum_b c_j^b)/\text{sd}(c_j^b)$ is largest, where $\text{sd}(c_j^b)$ indicates the standard deviation of $c_j^1, \ldots, c_j^B$. The resulting p-value is p_j.

Note that in the permutation step, we permute the rows of $\mathbf{X}_1$ and $\mathbf{X}_2$ without permuting $\mathbf{y}$; this means that under the permutation null distribution, $\mathbf{y}$ is not correlated with the columns of $\mathbf{X}_1$ and $\mathbf{X}_2$.

4.6 Computation of multiple canonical vectors

To compute multiple canonical vectors for sparse CCA, one can simply apply Algorithm 2.2 from Chapter 2.2, using $\mathbf{X}_1^T\mathbf{X}_2$ as the data matrix. In greater detail, the algorithm is as follows:

Algorithm 4.8: Computation of J sparse CCA canonical vectors

1. Let $\mathbf{Y}^1 = \mathbf{X}_1^T\mathbf{X}_2$.
2. For $j = 1, \ldots, J$:
   (a) Compute $\mathbf{w}_1^j$ and $\mathbf{w}_2^j$ by applying Algorithm 4.1 to data $\mathbf{Y}^j$.
   (b) Let $\mathbf{Y}^{j+1} = \mathbf{Y}^j - (\mathbf{w}_1^{jT}\mathbf{Y}^j\mathbf{w}_2^j)\mathbf{w}_1^j\mathbf{w}_2^{jT}$.

Then, $\mathbf{w}_1^j$ and $\mathbf{w}_2^j$ are the jth canonical vectors. In performing Step 2(a), note that Algorithm 4.1 simply makes use of the crossproduct matrix $\mathbf{X}_1^T\mathbf{X}_2$ and does not require knowledge of $\mathbf{X}_1$ and $\mathbf{X}_2$ individually.

To obtain J sparse sCCA factors, submatrices $\tilde{\mathbf{X}}_1$ and $\tilde{\mathbf{X}}_2$ are formed from the features most associated with the outcome. Algorithm 4.8 is then applied to this new data.

To obtain J sparse mCCA factors, note that Algorithm 4.3 in Chapter 4.3 requires knowledge only of the $\binom{K}{2}$ crossproduct matrices of the form $\mathbf{X}_s^T\mathbf{X}_t$ with $s < t$, rather than the raw data $\mathbf{X}_s$ and $\mathbf{X}_t$.

Algorithm 4.9: Computation of J sparse mCCA canonical vectors

1. For each $1 \leq s < t \leq K$, let $\mathbf{Y}_{st}^1 = \mathbf{X}_s^T \mathbf{X}_t$.

2. For $j = 1, \ldots, J$:

   (a) Compute $\mathbf{w}_1^j, \ldots, \mathbf{w}_K^j$ by applying Algorithm 4.3 in Chapter 4.3 to data $\{\mathbf{Y}_{st}^j\}_{s<t}$.

   (b) Let $\mathbf{Y}_{st}^{j+1} = \mathbf{Y}_{st}^j - (\mathbf{w}_s^{j\,T} \mathbf{Y}_{st}^j \mathbf{w}_t^j) \mathbf{w}_s^j \mathbf{w}_t^{j\,T}$.

3. $\mathbf{w}_1^j, \ldots, \mathbf{w}_K^j$ are the jth canonical vectors.

Chapter 5

Feature selection in clustering

In this chapter, we propose a framework for performing feature selection in clustering. This work will appear in Witten & Tibshirani (2010), and is reprinted with permission from the *Journal of the American Statistical Association*. Copyright 2010 by the American Statistical Association. All rights reserved.

5.1 An overview of feature selection in clustering

5.1.1 Motivation

Let $\mathbf{X}$ denote an $n \times p$ data matrix, with n observations and p features. Suppose that we wish to cluster the observations, and we suspect that the true underlying clusters differ only with respect to some of the features. In this chapter, we propose a method for *sparse clustering*, which allows us to group the observations using only an adaptively chosen subset of the features. This method is most useful for the high-dimensional setting where $p \gg n$, but can also be used when $p < n$. Sparse clustering has two main advantages:

1. If the underlying groups differ only in terms of some of the features, then it might result in more accurate identification of these groups than standard clustering.

2. It yields interpretable results, since one can determine precisely which features are responsible for the observed differences between the groups or clusters.

Though the framework that we propose in this chapter is quite general, we also consider the specific problems of how to perform feature selection for K-means and for hierarchical clustering. It turns out that our proposal for sparse hierarchical clustering is a special case of the PMD.

As a motivating example, we generated 500 independent observations from a bivariate normal distribution. A mean shift on the first feature defines the two classes. The resulting data, as well as the clusters obtained using standard 2-means clustering and our sparse 2-means clustering proposal, can be seen in Figure 5.1. Unlike standard 2-means clustering, our proposal for sparse 2-means clustering automatically identifies a subset of the features to use in clustering the observations. Here it uses only the first feature, and consequently agrees quite well with the true class labels. In this example, one could use an elliptical metric in order to identify the two classes without using feature selection. However, this will not work in general.

Clustering methods require some concept of the *dissimilarity* between pairs of observations. Let $d(\mathbf{x}_i, \mathbf{x}_{i'})$ denote some measure of dissimilarity between observations $\mathbf{x}_i$ and $\mathbf{x}_{i'}$, which are rows i and i' of the data matrix $\mathbf{X}$. Throughout this chapter, we will assume that d is additive in the features. That is, $d(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_{j=1}^{p} d_{i,i',j}$, where $d_{i,i',j}$ indicates the dissimilarity between observations i and i' along feature j. All of the data examples in this chapter take d to be squared Euclidean distance, $d_{i,i',j} = (X_{ij} - X_{i'j})^2$. However, other dissimilarity measures are possible, such as the absolute difference $d_{i,i',j} = |X_{ij} - X_{i'j}|$.

5.1.2 Past work on sparse clustering

A number of authors have noted the necessity of specialized clustering techniques for the high-dimensional setting. Here, we briefly review previous proposals for feature selection

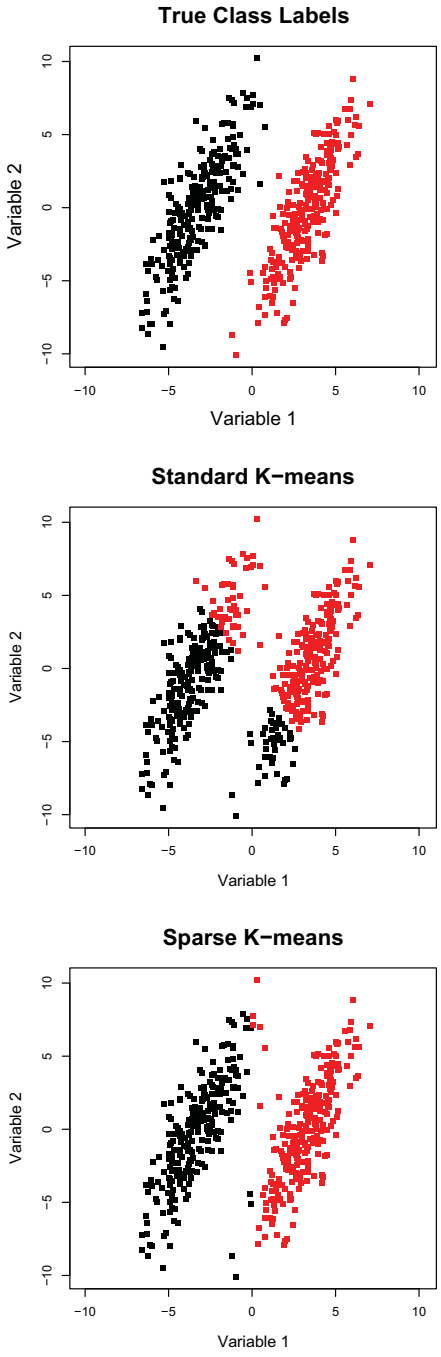

Figure 5.1: In a two-dimensional example, two classes differ only with respect to the first feature. Sparse 2-means clustering selects only the first feature, and therefore yields a superior result.

and dimensionality reduction in clustering.

One way to reduce the dimensionality of the data before clustering is by performing a matrix decomposition. One can approximate the $n \times p$ data matrix $\mathbf{X}$ as $\mathbf{X} \approx \mathbf{AB}$ where $\mathbf{A}$ is a $n \times q$ matrix and $\mathbf{B}$ is a $q \times p$ matrix, $q \ll p$. Then, one can cluster the observations using $\mathbf{A}$ as the data matrix, rather than $\mathbf{X}$. For instance, Ghosh & Chinnaiyan (2002) and Liu et al. (2003) propose performing principal components analysis (PCA) in order to obtain a matrix $\mathbf{A}$ of reduced dimensionality; then, the n rows of $\mathbf{A}$ can be clustered. Similarly, Tamayo et al. (2007) suggest decomposing $\mathbf{X}$ using the nonnegative matrix factorization (Lee & Seung 1999, Lee & Seung 2001), followed by clustering the rows of $\mathbf{A}$. However, these approaches have a number of drawbacks. First of all, the resulting clustering is not sparse in the features, since each of the columns of $\mathbf{A}$ is a function of the full set of p features. Moreover, there is no guarantee that $\mathbf{A}$ contains the signal that one is interested in detecting via clustering. In fact, Chang (1983) studies the effect of performing PCA to reduce the data dimension before clustering, and finds that this procedure is not justified since the principal components with largest eigenvalues do not necessarily provide the best separation between subgroups.

The model-based clustering framework has been studied extensively in recent years, and many of the proposals for feature selection and dimensionality reduction for clustering fall in this setting. An overview of model-based clustering can be found in McLachlan & Peel (2000) and Fraley & Raftery (2002). The basic idea is as follows. One can model the rows of $\mathbf{X}$ as independent multivariate observations drawn from a mixture model with K components; usually a mixture of Gaussians is used. That is, given the data, the log likelihood is

$$\sum_{i=1}^{n} \log[\sum_{k=1}^{K} \pi_k f_k(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)] \tag{5.1}$$

where f_k is a Gaussian density parametrized by its mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$. The EM algorithm (Dempster et al. 1977) can be used to fit this model.

However, when $p \approx n$ or $p \gg n$ a problem arises because the $p \times p$ covariance matrix $\mathbf{\Sigma}_k$ cannot be estimated from only n observations. Proposals for overcoming this problem include the factor analyzer approach of McLachlan et al. (2002) and McLachlan et al. (2003), which assumes that the observations lie in a low-dimensional latent factor space. This leads to dimensionality reduction but not sparsity.

It turns out that model-based clustering lends itself easily to feature selection. Rather than seeking $\boldsymbol{\mu}_k$ and $\mathbf{\Sigma}_k$ that maximize the log likelihood (5.1), one can instead maximize the log likelihood subject to a penalty that is chosen to yield sparsity in the features. This approach is taken in a number of papers, including Pan & Shen (2007), Wang & Zhu (2008), and Xie et al. (2008). For instance, if we assume that the features of $\mathbf{X}$ are centered to have mean zero, then Pan & Shen (2007) propose maximizing the penalized log likelihood

$$\sum_{i=1}^{n} \log[\sum_{k=1}^{K} \pi_k f_k(\mathbf{x}_i; \boldsymbol{\mu}_k, \mathbf{\Sigma}_k)] - \lambda \sum_{k=1}^{K} \sum_{j=1}^{p} |\mu_{kj}| \tag{5.2}$$

where $\mathbf{\Sigma}_1 = \ldots = \mathbf{\Sigma}_K$ is taken to be a diagonal matrix. That is, an L_1 penalty is applied to the elements of $\boldsymbol{\mu}_k$. When the nonnegative tuning parameter λ is large, then some of the elements of $\boldsymbol{\mu}_k$ will be exactly equal to zero. If, for some variable j, $\mu_{kj} = 0$ for all $k = 1, \ldots, K$, then the resulting clustering will not involve feature j. Hence, this yields a clustering that is sparse in the features.

Raftery & Dean (2006) also present a method for feature selection in the model-based clustering setting, using an entirely different approach. They recast the variable selection problem as a model selection problem: models containing nested subsets of variables are compared. The nested models are sparse in the features, and so this yields a method for sparse clustering. A related proposal is made in Maugis et al. (2009).

Friedman & Meulman (2004) propose *clustering objects on subsets of attributes (COSA)*. Let C_k denote the indices of the observations in the kth of K clusters. Then, the COSA

criterion is

$$\underset{C_1,\ldots,C_K,\mathbf{w}}{\text{minimize}}\{\sum_{k=1}^{K} a_k \sum_{i,i'\in C_k}\sum_{j=1}^{p}(w_j d_{i,i',j} + \lambda w_j \log w_j)\}$$
$$\text{subject to } \sum_{j=1}^{p} w_j = 1, w_j \geq 0 \; \forall j. \tag{5.3}$$

Actually, this is a simplified version of the COSA proposal, which allows for different feature weights within each cluster. Here, a_k is some function of the number of elements in cluster k, $\mathbf{w} \in \mathbb{R}^p$ is a vector of feature weights, and $\lambda \geq 0$ is a tuning parameter. It can be seen that this criterion is related to a weighted version of K-means clustering. Unfortunately, this proposal does not truly result in a sparse clustering, since all variables have nonzero weights for $\lambda > 0$. An extension of (5.3) is proposed in order to generalize the method to other types of clustering, such as hierarchical clustering. The proposed optimization algorithm is quite complex, and involves multiple tuning parameters.

Our proposal can be thought of as a much simpler version of (5.3). It is a general framework that can be applied in order to obtain sparse versions of a number of clustering methods. The resulting algorithms are efficient even when p is quite large.

5.1.3 The proposed sparse clustering framework

Suppose that we wish to cluster n observations on p dimensions; recall that $\mathbf{X}$ is of dimension $n \times p$. In this chapter, we take a general approach to the problem of sparse clustering. Let $\mathbf{X}_j \in \mathbb{R}^n$ denote feature j. Many clustering methods can be expressed as an optimization problem of the form

$$\underset{\mathbf{\Theta}\in D}{\text{maximize}}\{\sum_{j=1}^{p} f_j(\mathbf{X}_j, \mathbf{\Theta})\} \tag{5.4}$$

where $f_j(\mathbf{X}_j, \mathbf{\Theta})$ is some function that involves only the jth feature of the data, and $\mathbf{\Theta}$ is a parameter restricted to lie in a set D. K-means and hierarchical clustering are two such

examples, as we show in the next few sections. With K-means, for example, f_j turns out to be the between cluster sum of squares for feature j, and $\mathbf{\Theta}$ is a partition of the observations into K disjoint sets. We define *sparse clustering* as the solution to the problem

$$\underset{\mathbf{w};\mathbf{\Theta}\in D}{\text{maximize}}\{\sum_{j=1}^{p} w_j f_j(\mathbf{X}_j, \mathbf{\Theta})\} \text{ subject to } ||\mathbf{w}||^2 \leq 1, ||\mathbf{w}||_1 \leq s, w_j \geq 0 \; \forall j, \tag{5.5}$$

where w_j is a weight corresponding to feature j and s is a tuning parameter, $1 \leq s \leq \sqrt{p}$. We make a few observations about (5.5):

1. If $w_1 = \ldots = w_p$ in (5.5), then the criterion reduces to (5.4).

2. The L_1 penalty on $\mathbf{w}$ results in sparsity for small values of the tuning parameter s: that is, some of the w_j's will equal zero. The L_2 penalty also serves an important role, since without it, at most one element of $\mathbf{w}$ would be nonzero in general.

3. The value of w_j can be interpreted as the contribution of feature j to the resulting sparse clustering: a large value of w_j indicates a feature that contributes greatly, and $w_j = 0$ means that feature j is not involved in the clustering.

4. In general, for the formulation (5.5) to result in a nontrivial sparse clustering, it is necessary that $f_j(\mathbf{X}_j, \mathbf{\Theta}) > 0$ for some or all j. That is, if $f_j(\mathbf{X}_j, \mathbf{\Theta}) \leq 0$, then $w_j = 0$. If $f_j(\mathbf{X}_j, \mathbf{\Theta}) > 0$, then the nonnegativity constraint on w_j has no effect.

We optimize (5.5) using an iterative algorithm: holding $\mathbf{w}$ fixed, we optimize (5.5) with respect to $\mathbf{\Theta}$, and holding $\mathbf{\Theta}$ fixed, we optimize (5.5) with respect to $\mathbf{w}$. In general, we do not achieve a global optimum of (5.5) using this iterative approach; however, we are guaranteed that each iteration increases the objective function. The first optimization typically involves application of a standard clustering procedure to a weighted version of the data. To optimize (5.5) with respect to $\mathbf{w}$ with $\mathbf{\Theta}$ held fixed, we note that the problem

can be rewritten as

$$\underset{\mathbf{w}}{\text{maximize}}\{\mathbf{w}^T\mathbf{a}\} \text{ subject to } ||\mathbf{w}||^2 \leq 1, ||\mathbf{w}||_1 \leq s, w_j \geq 0 \; \forall j \tag{5.6}$$

where $a_j = f_j(\mathbf{X}_j, \mathbf{\Theta})$. Using Proposition 2.3.1 and arguments from Chapter 4.2.2, the solution to the convex problem (5.6) is $\mathbf{w} = \frac{S(\mathbf{a}_+,\Delta)}{||S(\mathbf{a}_+,\Delta)||_2}$, where $x_+ = \max(x, 0)$ and where $\Delta = 0$ if that results in $||\mathbf{w}||_1 \leq s$; otherwise, $\Delta > 0$ is chosen to yield $||\mathbf{w}||_1 = s$. Here, S is the soft-thresholding operator, given in (1.8).

In the next two sections we show that K-means clustering and hierarchical clustering can be described by criteria of the form (5.4). We then propose sparse versions of K-means clustering and hierarchical clustering using (5.5). The resulting criteria for sparse clustering take on simple forms, are easily optimized, and involve a single tuning parameter s that controls the number of features involved in the clustering. Note that our proposal is a general framework that can be applied to any clustering procedure for which a criterion of the form (5.4) is available.

5.2 Sparse K-means clustering

5.2.1 The sparse K-means method

K-means clustering minimizes the *within-cluster sum of squares* (WCSS). That is, it seeks to partition the n observations into K sets, or clusters, such that the WCSS

$$\sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} \sum_{j=1}^{p} d_{i,i',j} \tag{5.7}$$

is minimal, where n_k is the number of observations in cluster k and C_k contains the indices of the observations in cluster k. In general, $d_{i,i',j}$ can denote any dissimilarity measure between observations i and i' along feature j. However, in this chapter we will take $d_{i,i',j} =$

$(X_{ij} - X_{i'j})^2$; for this reason, we refer to (5.7) as the within-cluster sum of squares. Note that if we define the *between-cluster sum of squares* (BCSS) as

$$\sum_{j=1}^{p}(\frac{1}{n}\sum_{i=1}^{n}\sum_{i'=1}^{n}d_{i,i',j} - \sum_{k=1}^{K}\frac{1}{n_k}\sum_{i,i'\in C_k}d_{i,i',j}), \tag{5.8}$$

then minimizing the WCSS is equivalent to maximizing the BCSS.

One could try to develop a method for sparse K-means clustering by optimizing a weighted WCSS, subject to constraints on the weights: that is,

$$\begin{aligned}&\underset{C_1,\ldots,C_K,\mathbf{w}}{\text{maximize}}\{\sum_{j=1}^{p}w_j(-\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i,i'\in C_k}d_{i,i',j})\}\\ &\text{subject to } ||\mathbf{w}||^2 \leq 1, ||\mathbf{w}||_1 \leq s, w_j \geq 0\ \forall j.\end{aligned} \tag{5.9}$$

Here, s is a tuning parameter. Since each element of the weighted sum is negative, the maximum occurs when all weights are zero, regardless of the value of s. This is not an interesting solution. We instead maximize a weighted BCSS, subject to constraints on the weights. Our *sparse K-means clustering criterion* is as follows:

$$\begin{aligned}&\underset{C_1,\ldots,C_K,\mathbf{w}}{\text{maximize}}\{\sum_{j=1}^{p}w_j(\frac{1}{n}\sum_{i=1}^{n}\sum_{i'=1}^{n}d_{i,i',j} - \sum_{k=1}^{K}\frac{1}{n_k}\sum_{i,i'\in C_k}d_{i,i',j})\}\\ &\text{subject to } ||\mathbf{w}||^2 \leq 1, ||\mathbf{w}||_1 \leq s, w_j \geq 0\ \forall j.\end{aligned} \tag{5.10}$$

The weights will be sparse for an appropriate choice of the tuning parameter s, which should satisfy $1 \leq s \leq \sqrt{p}$. Note that if $w_1 = \ldots = w_p$, then (5.10) simply reduces to the standard K-means clustering criterion. We observe that (5.8) and (5.10) are special cases of (5.4) and (5.5) where $\mathbf{\Theta} = (C_1, \ldots, C_K)$, $f_j(\mathbf{X}_j, \mathbf{\Theta}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{i'=1}^{n}d_{i,i',j} - \sum_{k=1}^{K}\frac{1}{n_k}\sum_{i,i'\in C_k}d_{i,i',j}$, and D denotes the set of all possible partitions of the observations into K clusters.

The criterion (5.10) assigns a weight to each feature, based on the increase in BCSS that

the feature can contribute. First, consider the criterion with the weights $w_1, \ldots, w_p$ fixed. It reduces to a clustering problem, using a weighted dissimilarity measure. Second, consider the criterion with the clusters $C_1, \ldots, C_K$ fixed. Then a weight will be assigned to each feature based on the BCSS of that feature; features with larger BCSS will be given larger weights. We present an iterative algorithm for solving (5.10). Again, we do not expect to obtain the global optimum using this iterative approach.

Algorithm 5.1: Sparse K-means clustering

1. Initialize $\mathbf{w}$ as $w_1 = \ldots = w_p = \frac{1}{\sqrt{p}}$.

2. Iterate:

   (a) Holding $\mathbf{w}$ fixed, optimize (5.10) with respect to $C_1, \ldots, C_K$. That is,

   $$\underset{C_1,\ldots,C_K}{\text{minimize}}\{\sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} \sum_{j=1}^{p} w_j d_{i,i',j}\} \tag{5.11}$$

   by applying the standard K-means algorithm to the $n \times n$ dissimilarity matrix with (i, i') element $\sum_{j=1}^{p} w_j d_{i,i',j}$.

   (b) Holding $C_1, \ldots, C_K$ fixed, optimize (5.10) with respect to $\mathbf{w}$, as follows: $\mathbf{w} = \frac{S(\mathbf{a}_+, \Delta)}{||S(\mathbf{a}_+, \Delta)||_2}$ where

   $$a_j = (\frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} d_{i,i',j} - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} d_{i,i',j}) \tag{5.12}$$

   and $\Delta = 0$ if that results in $||\mathbf{w}||_1 < s$; otherwise, $\Delta > 0$ is chosen so that $||\mathbf{w}||_1 = s$.

3. The clusters are given by $C_1, \ldots, C_K$, and the feature weights corresponding to this clustering are given by $w_1, \ldots, w_p$.

When d is squared Euclidean distance, Step 2(a) amounts to performing K-means on the data after scaling each feature j by $\sqrt{w_j}$. In our implementation of sparse K-means, we iterate Step 2 until the stopping criterion

$$\frac{\sum_{j=1}^{p} |w_j^r - w_j^{r-1}|}{\sum_{j=1}^{p} |w_j^{r-1}|} < \epsilon \tag{5.13}$$

is satisfied; we take $\epsilon = 10^{-4}$. Here, $\mathbf{w}^r$ indicates the set of weights obtained at iteration r. In the examples that we have examined, this criterion tends to be satisfied within no more than 5 to 10 iterations. However, we note that Algorithm 5.1 generally will not converge to the global optimum of the criterion (5.10), since the criterion is not convex and uses in Step 2(a) the algorithm for K-means clustering, which is not guaranteed to find a global optimum (see e.g. MacQueen 1967).

Note the similarity between the COSA criterion (5.3) and (5.10): when $a_k = \frac{1}{n_k}$ in (5.3), then both criteria involve minimizing a weighted function of the WCSS, where the feature weights reflect the importance of each feature in the clustering. However, (5.3) does not result in weights that are exactly equal to zero unless $\lambda = 0$, in which case only one weight is nonzero. The combination of L_1 and L_2 constraints in (5.10) yields the desired effect.

Chapter 5.6.1 contains an additional remark about our criterion for sparse K-means clustering.

5.2.2 Selection of tuning parameter for sparse K-means

Algorithm 5.1 has one tuning parameter, s, which is the L_1 bound on $\mathbf{w}$ in (5.10). We assume that K, the number of clusters, is fixed. The problem of selecting K is outside of the scope of this work, and has been discussed extensively in the literature for standard K-means clustering; we refer the interested reader to Milligan & Cooper (1985), Kaufman & Rousseeuw (1990), Tibshirani et al. (2001), Sugar & James (2003), and Tibshirani & Walther (2005).

A method for choosing the value of s is required. Note that one cannot simply select s to maximize the objective function in (5.10), since as s is increased, the objective will increase as well. Instead, we apply a permutation approach that is closely related to the *gap statistic* of Tibshirani et al. (2001) for selecting the number of clusters K in standard K-means clustering.

Algorithm 5.2: Gap statistic for sparse K-means tuning parameter selection

1. Obtain permuted data sets $\mathbf{X}_1, \ldots, \mathbf{X}_B$ by independently permuting the observations within each feature.

2. For each candidate tuning parameter value s:

   (a) Compute $O(s) = \sum_j w_j (\frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} d_{i,i',j} - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} d_{i,i',j})$, the objective obtained by performing sparse K-means with tuning parameter value s on the data $\mathbf{X}$.

   (b) For $b = 1, 2, \ldots, B$, compute $O_b(s)$, the objective obtained by performing sparse K-means with tuning parameter value s on the data $\mathbf{X}_b$.

   (c) Calculate $\text{Gap}(s) = \log(O(s)) - \frac{1}{B} \sum_{b=1}^{B} \log(O_b(s))$.

3. Choose s^* corresponding to the largest value of $\text{Gap}(s)$. Alternatively, one can choose s^* to equal the smallest value for which $\text{Gap}(s^*)$ is within a standard deviation of $\log(O_b(s^*))$ of the largest value of $\text{Gap}(s)$.

Note that while there may be strong correlations between the features in the original data $\mathbf{X}$, the features in the permuted data sets $\mathbf{X}_1, \ldots, \mathbf{X}_B$ are uncorrelated with each other. The gap statistic measures the strength of the clustering obtained on the real data relative to the clustering obtained on null data that does not contain subgroups. The optimal tuning parameter value occurs when this quantity is greatest.

In Figure 5.2, we apply this method to a simple example with 6 equally sized classes, where $n = 120$, $p = 2000$, and 200 features differ between classes. In the figure we have used the *classification error rate* (CER) for two partitions of a set of n observations. This is defined as follows. Let P and Q denote the two partitions; P might be the true class labels, and Q might be a partition obtained by clustering. Let $1_{P(i,i')}$ be an indicator for whether partition P places observations i and i' in the same group, and define $1_{Q(i,i')}$ analogously. Then, the CER (used for example in Chipman & Tibshirani 2005) is defined as

$$\frac{\sum_{i>i'} |1_{P(i,i')} - 1_{Q(i,i')}|}{\binom{n}{2}}. \tag{5.14}$$

The CER equals 0 if the partitions P and Q agree perfectly; a high value indicates disagreement. Note that CER is one minus the Rand index (Rand 1971).

5.2.3 A simulation study of sparse K-means

Simulation 1: A comparison of sparse and standard K-means

We compare the performances of standard and sparse K-means in a simulation study where $q = 50$ features differ between $K = 3$ classes. $X_{ij} \sim N(\mu_{ij}, 1)$ independent; $\mu_{ij} = \mu(1_{i\in C_1, j\leq q} - 1_{i\in C_2, j\leq q})$. Data sets were generated with various values of μ and p, with 20 observations per class. The results can be seen in Tables 5.1, 5.2, and 5.3. In this example, when $p > q$, sparse 3-means tends to outperform standard 3-means, since it exploits the sparsity of the signal. On the other hand, when $p = q$, then standard 3-means is at an advantage, since it gives equal weights to all features. The value of the tuning parameter s for sparse 3-means was selected to maximize the gap statistic. As seen in Table 5.3, this generally resulted in more than 50 features with nonzero weights. This reflects the fact that the tuning parameter selection method tends not to be very accurate. Fewer features with nonzero weights would result from selecting the tuning parameter at the smallest value that is within one standard deviation of the maximal gap statistic, as described in

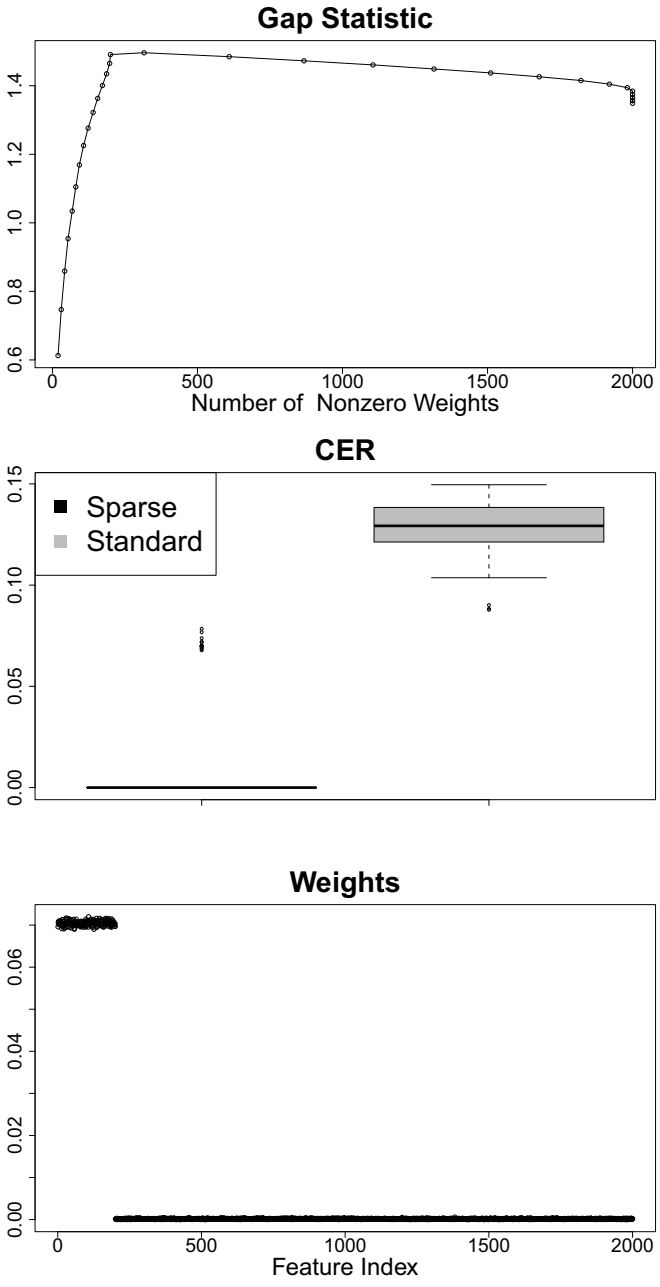

Figure 5.2: Sparse and standard 6-means clustering applied to a simulated 6-class example. **Left:** Gap statistics averaged over 10 simulated data sets. **Center:** CERs obtained using sparse and standard 6-means clustering on 100 simulated data sets. **Right:** Weights obtained using sparse 6-means clustering, averaged over 100 simulated data sets. First 200 features differ between classes.

	$p = 50$	$p = 200$	$p = 500$	$p = 1000$
$\mu = 0.6$	**0.07(0.01)**	0.184(0.015)	0.22(0.009)	0.272(0.006)
$\mu = 0.7$	**0.023(0.005)**	0.077(0.009)	0.16(0.012)	0.232(0.01)
$\mu = 0.8$	**0.013(0.004)**	0.038(0.007)	0.08(0.005)	0.198(0.01)
$\mu = 0.9$	**0.001(0.001)**	0.013(0.005)	0.048(0.008)	0.102(0.013)
$\mu = 1$	0.002(0.002)	0.004(0.002)	0.013(0.004)	0.05(0.006)

Table 5.1: Standard 3-means results for Simulation 1. The reported values are the mean (and standard error) of the CER over 20 simulations. The μ/p combinations for which the CER of standard 3-means is significantly less than that of sparse 3-means (at level $\alpha = 0.05$) are shown in bold.

	$p = 50$	$p = 200$	$p = 500$	$p = 1000$
$\mu = 0.6$	0.146(0.014)	**0.157(0.016)**	**0.183(0.015)**	0.241(0.017)
$\mu = 0.7$	0.081(0.011)	**0.049(0.008)**	**0.078(0.013)**	**0.098(0.013)**
$\mu = 0.8$	0.043(0.008)	**0.031(0.007)**	**0.031(0.005)**	**0.037(0.006)**
$\mu = 0.9$	0.015(0.006)	**0.005(0.003)**	**0.014(0.004)**	**0.014(0.004)**
$\mu = 1$	0.009(0.004)	0.004(0.002)	**0.001(0.001)**	**0.002(0.002)**

Table 5.2: Sparse 3-means results for Simulation 1. The reported values are the mean (and standard error) of the CER over 20 simulations. The μ/p combinations for which the CER of sparse 3-means is significantly less than that of standard 3-means (at level $\alpha = 0.05$) are shown in bold.

Chapter 5.2.2.

Simulation 2: A comparison with other approaches

We compare the performance of sparse K-means to a number of competitors:

1. **The COSA proposal of Friedman & Meulman (2004).** COSA was run using the `R` code available from the website `http://www-stat.stanford.edu/~jhf/COSA.html`, in order to obtain a reweighted dissimilarity matrix. Then, two methods were used to obtain a clustering:

    - 3-medoids clustering (using the *partitioning around medoids* algorithm described in Kaufman & Rousseeuw 1990) was performed on the reweighted dissimilarity matrix.

	$p = 50$	$p = 200$	$p = 500$	$p = 1000$
$\mu = 0.6$	41.35(0.895)	167.4(7.147)	243.1(31.726)	119.45(41.259)
$\mu = 0.7$	40.85(0.642)	195.65(2.514)	208.85(19.995)	130.15(17.007)
$\mu = 0.8$	38.2(0.651)	198.85(0.654)	156.35(13.491)	106.7(10.988)
$\mu = 0.9$	38.7(0.719)	200(0)	204.75(19.96)	83.7(9.271)
$\mu = 1$	36.95(0.478)	200(0)	222.85(20.247)	91.65(14.573)

Table 5.3: Sparse 3-means results for Simulation 1. The mean number of nonzero feature weights resulting from Algorithm 5.2 is shown; standard errors are given in parentheses. Note that 50 features differ between the three classes.

- Hierarchical clustering with average linkage was performed on the reweighted dissimilarity matrix, and the dendrogram was cut so that 3 groups were obtained.

2. **The model-based clustering approach of Raftery & Dean (2006).** It was run using the `R` package `clustvarsel`, available from `http://cran.r-project.org/`.

3. **The penalized log likelihood approach of Pan & Shen (2007).** `R` code implementing this method was provided by the authors.

4. **PCA followed by 3-means clustering.** Only the first principal component was used, since in the simulations considered the first principal component contained the signal. This is similar to several proposals in the literature (see e.g. Ghosh & Chinnaiyan 2002, Liu et al. 2003, Tamayo et al. 2007).

The setup is similar to that of Chapter 5.2.3, in that there are $K = 3$ classes and $X_{ij} \sim N(\mu_{ij}, 1)$ independent; $\mu_{ij} = \mu(1_{i \in C_1, j \leq q} - 1_{i \in C_2, j \leq q})$. Two simulations were run: a small simulation with $p = 25, q = 5$, and 10 observations per class, and a larger simulation with $p = 500, q = 50$, and 20 observations per class. The results are shown in Table 5.4. The quantitities reported are the mean and standard error (given in parentheses) of the CER and the number of nonzero coefficients, over 25 simulated data sets. Note that the method of Raftery & Dean (2006) was run only on the smaller simulation for computational reasons.

Simulation	Method	CER	Num. Nonzero
Small Simulation: $p = 25, q = 5$, 10 obs. per class	Sparse K-means	0.112(0.019)	8.2(0.733)
	K-means	0.263(0.011)	25(0)
	Pan and Shen	0.126(0.017)	6.72(0.334)
	COSA w/Hier. Clust.	0.381(0.016)	25(0)
	COSA w/K-medoids	0.369(0.012)	25(0)
	Raftery and Dean	0.514(0.031)	22(0.86)
	PCA w/K-means	0.16(0.012)	25(0)
Large Simulation: $p = 500, q = 50$, 20 obs. per class	Sparse K-means	0.106(0.019)	141.92(9.561)
	K-means	0.214(0.011)	500(0)
	Pan and Shen	0.134(0.013)	76(3.821)
	COSA w/Hier. Clust.	0.458(0.011)	500(0)
	COSA w/K-medoids	0.427(0.004)	500(0)
	PCA w/K-means	0.058(0.006)	500(0)

Table 5.4: Results for Simulation 2. The quantities reported are the mean and standard error (given in parentheses) of the CER, and of the number of nonzero coefficients, over 25 simulated data sets.

We make a few comments about Table 5.4. First of all, neither variant of COSA performed well in this example, in terms of CER. This is somewhat surprising. However, COSA allows the features to take on a different set of weights with respect to each cluster. In the simulation, each cluster is defined on the same set of features, and COSA may have lost power by allowing different weights for each cluster. The method of Raftery & Dean (2006) also did quite poorly in this example, although its performance seems to improve somewhat as the signal to noise ratio in the simulation is increased (results not shown). The penalized model-based clustering method of Pan & Shen (2007) resulted in low CER as well as sparsity in both simulations. In addition, the simple method of PCA followed by 3-means clustering yielded quite low CER. However, since the principal components are linear combinations of all of the features, the resulting clustering is not sparse in the features and thus does not achieve the stated goal in this chapter of performing feature selection.

In both simulations, sparse K-means performed quite well, in that it resulted in a low CER and sparsity. The tuning parameter was chosen to maximize the gap statistic; however, greater sparsity could have been achieved by choosing the smallest tuning parameter value

within one standard deviation of the maximal gap statistic, as described in Algorithm 5.2. Our proposal also has the advantage of generalizing to other types of clustering, as described next.

5.3 Sparse hierarchical clustering

5.3.1 The sparse hierarchical clustering method

Hierarchical clustering produces a dendrogram that represents a nested set of clusters: depending on where the dendrogram is cut, between 1 and n clusters can result. One could develop a method for sparse hierarchical clustering by cutting the dendrogram at some height and maximizing a weighted version of the resulting BCSS, as in Chapter 5.2. However, it is not clear where the dendrogram should be cut, nor whether multiple cuts should be made and somehow combined. Instead, we pursue a simpler and more natural approach to sparse hierarchical clustering.

Note that hierarchical clustering takes as input a $n \times n$ dissimilarity matrix $\mathbf{U}$. The clustering can use any type of linkage - complete, average, or single. If $\mathbf{U}$ is the *overall dissimilarity matrix* $\{\sum_{j=1}^{p} d_{i,i',j}\}_{i,i'}$, then *standard hierarchical clustering* results. In this section, we cast the overall dissimilarity matrix $\{\sum_j d_{i,i',j}\}_{i,i'}$ in the form (5.4), and then propose a criterion of the form (5.5) that leads to a reweighted dissimilarity matrix that is sparse in the features. When hierarchical clustering is performed on this reweighted dissimilarity matrix, then *sparse hierarchical clustering* results.

Since scaling the dissimilarity matrix by a factor does not affect the shape of the resulting dendrogram, we ignore proportionality constants in the following discussion. Consider the criterion

$$\underset{\mathbf{U}}{\text{maximize}}\{\sum_{j=1}^{p}\sum_{i,i'} d_{i,i',j}U_{i,i'}\} \text{ subject to } \sum_{i,i'} U_{i,i'}^2 \leq 1. \tag{5.15}$$

Let $\mathbf{U}^*$ solve (5.15). It is not hard to show that $U^*_{i,i'} \propto \sum_{j=1}^{p} d_{i,i',j}$, and so performing

hierarchical clustering on $\mathbf{U}^*$ results in standard hierarchical clustering. So we can think of standard hierarchical clustering as resulting from the criterion (5.15). To obtain sparsity in the features, we modify (5.15) by multiplying each element of the summation over j by a weight w_j, subject to constraints on the weights:

$$\begin{aligned} &\underset{\mathbf{w},\mathbf{U}}{\text{maximize}}\{\sum_{j=1}^{p} w_j \sum_{i,i'} d_{i,i',j} U_{i,i'}\} \\ &\text{subject to } \sum_{i,i'} U_{i,i'}^2 \leq 1, ||\mathbf{w}||^2 \leq 1, ||\mathbf{w}||_1 \leq s, w_j \geq 0 \; \forall j. \end{aligned} \tag{5.16}$$

The $\mathbf{U}^{**}$ that solves (5.16) is proportional to $\{\sum_{j=1}^{p} d_{i,i',j} w_j\}_{i,i'}$. Since $\mathbf{w}$ is sparse for small values of the tuning parameter s, $\mathbf{U}^{**}$ involves only a subset of the features, and so performing hierarchical clustering on $\mathbf{U}^{**}$ results in sparse hierarchical clustering. We refer to (5.16) as the *sparse hierarchical clustering criterion*. Observe that (5.15) takes the form (5.4) with $\mathbf{\Theta} = \mathbf{U}$, $f_j(\mathbf{X}_j, \mathbf{\Theta}) = \sum_{i,i'} d_{i,i',j} U_{i,i'}$ and $\mathbf{\Theta} \in D$ corresponding to $\sum_{i,i'} U_{i,i'}^2 \leq 1$. It follows directly that (5.16) takes the form (5.5), and so sparse hierarchical clustering fits into the framework of Chapter 5.1.3.

By inspection, (5.16) is biconvex in $\mathbf{U}$ and $\mathbf{w}$, and so can be optimized using a simple iterative algorithm. However, before we present this algorithm, we introduce some additional notation that will prove useful. Let $\mathbf{D} \in \mathbb{R}^{n^2 \times p}$ be the matrix in which column j consists of the elements $\{d_{i,i',j}\}_{i,i'}$, strung out into a vector. Then, $\sum_{j=1}^{p} w_j \sum_{i,i'} d_{i,i',j} U_{i,i'} = \mathbf{u}^T \mathbf{D} \mathbf{w}$ where $\mathbf{u} \in \mathbb{R}^{n^2}$ is obtained by stringing out $\mathbf{U}$ into a vector. It follows that the criterion (5.16) is equivalent to

$$\underset{\mathbf{w},\mathbf{u}}{\text{maximize}}\{\mathbf{u}^T \mathbf{D} \mathbf{w}\} \text{ subject to } ||\mathbf{u}||^2 \leq 1, ||\mathbf{w}||^2 \leq 1, ||\mathbf{w}||_1 \leq s, w_j \geq 0 \; \forall j. \tag{5.17}$$

We now present our algorithm for sparse hierarchical clustering.

Algorithm 5.3: Sparse hierarchical clustering

1. Initialize $\mathbf{w}$ as $w_1 = \ldots = w_p = \frac{1}{\sqrt{p}}$.

2. Iterate until convergence:

   (a) Update $\mathbf{u} = \frac{\mathbf{Dw}}{||\mathbf{Dw}||_2}$.

   (b) Update $\mathbf{w} = \frac{S(\mathbf{a}_+,\Delta)}{||S(\mathbf{a}_+,\Delta)||_2}$ where $\mathbf{a} = \mathbf{D}^T\mathbf{u}$ and $\Delta = 0$ if this results in $||\mathbf{w}||_1 \leq s$; otherwise, $\Delta > 0$ is chosen such that $||\mathbf{w}||_1 = s$.

3. Rewrite $\mathbf{u}$ as a $n \times n$ matrix, $\mathbf{U}$.

4. Perform hierarchical clustering on the $n \times n$ dissimilarity matrix $\mathbf{U}$.

Observe that (5.17) is the SPC criterion (3.3), with an additional nonnegativity constraint on $\mathbf{w}$. If $d_{i,i',j} \geq 0$, as is usually the case, then the nonnegativity constraint can be dropped.

When viewed in this way, our method for sparse hierarchical clustering is quite simple. The first SPC of the $n^2 \times p$ matrix $\mathbf{D}$ is denoted $\mathbf{w}$. Then $\mathbf{u} \propto \mathbf{Dw}$ can be rewritten as a $n \times n$ matrix $\mathbf{U}$, which is a weighted linear combination of the feature-wise dissimilarity matrices. When s is small, then some elements of $\mathbf{w}$ will equal zero, and so $\mathbf{U}$ will depend on only a subset of the features. We then perform hierarchical clustering on $\mathbf{U}$ in order to obtain a dendrogram that is based only on an adaptively chosen subset of the features.

In our implementation of Algorithm 5.3, we use (5.13) as a stopping criterion in Step 2. In our experience, the stopping criterion generally is satisfied within 10 iterations. As mentioned earlier, the criterion (5.17) is biconvex in $\mathbf{u}$ and $\mathbf{w}$, and we are not guaranteed convergence to a global optimum using this iterative approach.

5.3.2 A simple model for sparse hierarchical clustering

We study the behaviors of sparse and standard hierarchical clustering under a simple model. Suppose that the n observations fall into two classes, C_1 and C_2, which differ only with respect to the first q features. The elements X_{ij} are independent and normally distributed

with a mean shift between the two classes in the first q features:

$$X_{ij} \sim \begin{cases} N(\mu_j + c, \sigma^2) & \text{if } j \leq q, i \in C_1, \\ N(\mu_j, \sigma^2) & \text{otherwise.} \end{cases} \tag{5.18}$$

Note that for $i \neq i'$,

$$X_{ij} - X_{i'j} \sim \begin{cases} N(\pm c, 2\sigma^2) & \text{if } j \leq q \text{ and } i, i' \text{ in different classes,} \\ N(0, 2\sigma^2) & \text{otherwise.} \end{cases} \tag{5.19}$$

Let $d_{i,i',j} = (X_{ij} - X_{i'j})^2$; that is, the dissimilarity measure is squared Euclidean distance. Then, for $i \neq i'$,

$$d_{i,i',j} \sim \begin{cases} 2\sigma^2 \chi_1^2(\frac{c^2}{2\sigma^2}) & \text{if } j \leq q \text{ and } i, i' \text{ in different classes,} \\ 2\sigma^2 \chi_1^2 & \text{otherwise,} \end{cases} \tag{5.20}$$

where $\chi_1^2(\lambda)$ denotes the noncentral χ_1^2 distribution with noncentrality parameter λ. This means that the overall dissimilarity matrix used by standard hierarchical clustering has off-diagonal elements

$$\sum_j d_{i,i',j} \sim \begin{cases} 2\sigma^2 \chi_p^2(\frac{qc^2}{2\sigma^2}) & \text{if } i, i' \text{ in different classes,} \\ 2\sigma^2 \chi_p^2 & \text{otherwise,} \end{cases} \tag{5.21}$$

and so for $i \neq i'$,

$$\mathrm{E}(d_{i,i',j}) = \begin{cases} 2\sigma^2 + c^2 & \text{if } j \leq q \text{ and } i, i' \text{ in different classes,} \\ 2\sigma^2 & \text{otherwise,} \end{cases} \tag{5.22}$$

and

$$\mathrm{E}(\sum_j d_{i,i',j}) = \begin{cases} 2p\sigma^2 + qc^2 & \text{if } i, i' \text{ in different classes,} \\ 2p\sigma^2 & \text{otherwise.} \end{cases} \tag{5.23}$$

We now consider the behavior of sparse hierarchical clustering. Suppose that $w_j \propto 1_{j \leq q}$; this corresponds to the ideal situation in which the important features have equal nonzero weights, and the unimportant features have weights that equal zero. Then the dissimilarity matrix used for sparse hierarchical clustering has elements

$$\sum_j w_j d_{i,i',j} \propto \begin{cases} 2\sigma^2 \chi^2_q(\frac{qc^2}{2\sigma^2}) & \text{if } i, i' \text{ in different classes,} \\ 2\sigma^2 \chi^2_q & \text{otherwise.} \end{cases} \tag{5.24}$$

So in this ideal setting, the dissimilarity matrix used for sparse hierarchical clustering (5.24) is a denoised version of the dissimilarity matrix used for standard hierarchical clustering (5.21). Of course, in practice, w_j is not proportional to $1_{j \leq q}$.

We now allow $\mathbf{w}$ to take a more general form. Recall that $\mathbf{w}$ is the first SPC of $\mathbf{D}$, obtained by writing $\{d_{i,i',j}\}_{i,i',j}$ as a $n^2 \times p$ matrix. To simplify the discussion, suppose instead that $\mathbf{w}$ is the first SPC of $\mathrm{E}(\mathbf{D})$. Then, $\mathbf{w}$ is not random, and

$$w_1 = \ldots = w_q > w_{q+1} = \ldots = w_p \tag{5.25}$$

from (5.22). To see this latter point, note that by the Algorithm 5.3, $\mathbf{w}$ is obtained by repeating the operation

$$\mathbf{w} = \frac{S(\mathrm{E}(\mathbf{D})^T \mathrm{E}(\mathbf{D})\mathbf{w}, \Delta)}{||S(\mathrm{E}(\mathbf{D})^T \mathrm{E}(\mathbf{D})\mathbf{w}, \Delta)||_2} \tag{5.26}$$

until convergence, for $\Delta \geq 0$. Initially, $w_1 = \ldots = w_p$. By inspection, in each subsequent iteration, (5.25) holds true.

From (5.22), the expectations of the off-diagonal elements of the dissimilarity matrix

used for sparse hierarchical clustering are therefore

$$\mathrm{E}(\sum_j w_j d_{i,i',j}) = \sum_j w_j \mathrm{E}(d_{i,i',j}) = \begin{cases} 2\sigma^2 \sum_j w_j + c^2 \sum_{j \le q} w_j & \text{if } i, i' \text{ in different classes,} \\ 2\sigma^2 \sum_j w_j & \text{otherwise.} \end{cases} \tag{5.27}$$

By comparing (5.23) to (5.27), and using (5.25), we see that the expected dissimilarity between observations in different classes relative to observations in the same class is greater for sparse hierarchical clustering than for standard hierarchical clustering. Note that we have taken the weight vector $\mathbf{w}$ to be the first SPC of $\mathrm{E}(\mathbf{D})$, rather than the first SPC of $\mathbf{D}$.

5.3.3 Selection of tuning parameter for sparse hierarchical clustering

We now consider the problem of selecting a value for s, the L_1 bound for $\mathbf{w}$ in the sparse hierarchical clustering criterion. We essentially apply Algorithm 5.2, in this case letting $O(s) = \sum_j w_j \sum_{i,i'} d_{i,i',j} U_{i,i'}$.

We demonstrate the performance of this tuning parameter selection method on the simulated 6-class data set used for Figure 5.2. We performed standard, COSA, and sparse hierarchical clustering with complete linkage. The results can be seen in Figure 5.3. Sparse hierarchical clustering results in better separation between the subgroups. Moreover, the correct features are given nonzero weights.

In this example and throughout this chapter, we used $d_{i,i',j} = (X_{ij} - X_{i'j})^2$; that is, the dissimilarity measure used was squared Euclidean distance. However, in many simulated examples, we found that better performance results from using absolute value dissimilarity, $d_{i,i',j} = |X_{ij} - X_{i'j}|$.

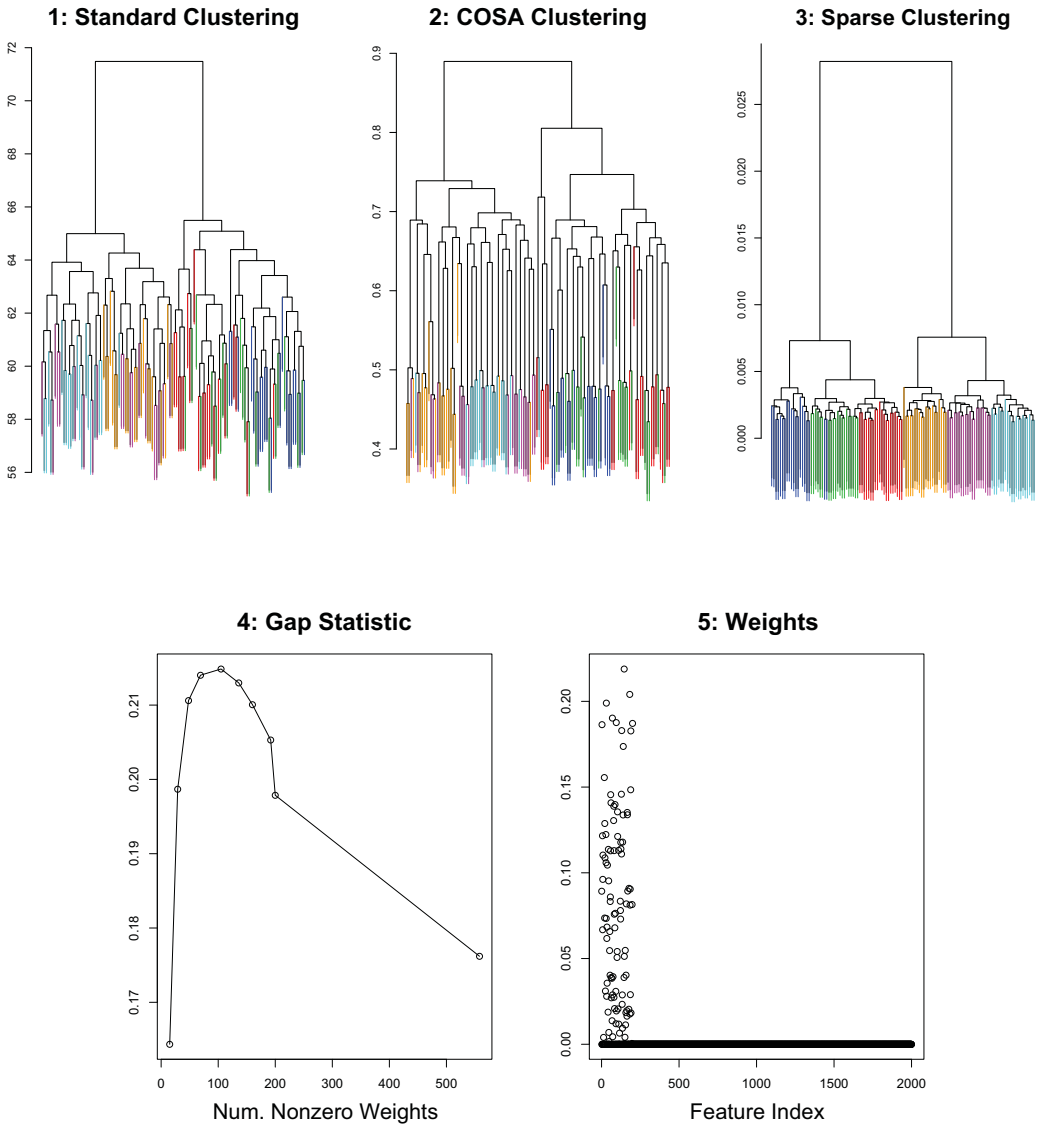

Figure 5.3: Standard hierarchical clustering, COSA, and sparse hierarchical clustering with complete linkage were performed on simulated 6-class data. **1**, **2**, **3**: The color of each leaf indicates its class identity. CERs were computed by cutting each dendrogram at the height that results in 6 clusters: standard, COSA, and sparse clustering yielded CERs of 0.169, 0.160, and 0.0254. **4**: The gap statistics obtained for sparse hierarchical clustering, as a function of the number of features included for each value of the tuning parameter. **5**: The $\mathbf{w}$ obtained using sparse hierarchical clustering; note that the six classes differ with respect to the first 200 features.

5.3.4 Complementary sparse clustering

Standard hierarchical clustering is often dominated by a single group of features that have high variance and are highly correlated with each other. The same is true of sparse hierarchical clustering. Nowak & Tibshirani (2008) propose *complementary clustering*, a method that allows for the discovery of a secondary clustering after removing the signal found in the standard hierarchical clustering. Here we provide a method for *complementary sparse clustering*, an analogous approach for the sparse clustering framework. This simple method follows directly from our sparse hierarchical clustering proposal.

As in Chapter 5.3.1, we let $\mathbf{D}$ denote the $n^2 \times p$ matrix of which column j consists of $\{d_{i,i',j}\}_{i,i'}$ in vector form. Let $\mathbf{u}_1, \mathbf{w}_1$ solve (5.17); that is, $\mathbf{U}_1$ (obtained by writing $\mathbf{u}_1$ in matrix form) is a weighted linear combination of the feature-wise dissimilarity matrices, and $\mathbf{w}_1$ denotes the corresponding feature weights. Then, the criterion

$$\begin{aligned} &\underset{\mathbf{u}_2,\mathbf{w}_2}{\text{maximize}}\{\mathbf{u}_2^T\mathbf{D}\mathbf{w}_2\} \\ &\text{subject to } ||\mathbf{u}_2||^2 \leq 1, ||\mathbf{w}_2||^2 \leq 1, ||\mathbf{w}_2||_1 \leq s, \mathbf{u}_1^T\mathbf{u}_2 = 0, w_j \geq 0 \; \forall j \end{aligned} \tag{5.28}$$

results in a dissimilarity matrix $\mathbf{U}_2$, obtained by writing $\mathbf{u}_2$ as a $n \times n$ matrix, that yields a complementary sparse clustering. The feature weights for this secondary clustering are given by $\mathbf{w}_2$. Note that (5.28) is simply the proposal (3.13) for finding the second SPC of $\mathbf{D}$ subject to orthogonality constraints, with an additional nonnegativity constraint on $\mathbf{w}$.

Observe that $\mathbf{U}_2$ is symmetric with zeroes on the diagonal, and that due to the constraint that $\mathbf{u}_1^T\mathbf{u}_2 = 0$, some elements of $\mathbf{U}_2$ will be negative. However, since only the off-diagonal elements of a dissimilarity matrix are used in hierarchical clustering, and since the shape of the dendrogram is not affected by adding a constant to the off-diagonal elements, in practice this is not a problem. The algorithm for complementary sparse clustering is as follows:

Algorithm 5.4: Complementary sparse hierarchical clustering

1. Apply Algorithm 5.3 to $\mathbf{D}$, and let $\mathbf{u}_1$ denote the resulting linear combination of the p feature-wise dissimilarity matrices, written in vector form.
2. Initialize $\mathbf{w}_2$ as $w_{21} = \ldots = w_{2p} = \frac{1}{\sqrt{p}}$.
3. Iterate until convergence:
   (a) Update $\mathbf{u}_2 = \frac{(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T)\mathbf{D}\mathbf{w}_2}{||(\mathbf{I}-\mathbf{u}_1\mathbf{u}_1^T)\mathbf{D}\mathbf{w}_2||_2}$.
   (b) Update $\mathbf{w}_2 = \frac{S(\mathbf{a}_+,\Delta)}{||S(\mathbf{a}_+,\Delta)||_2}$ where $\mathbf{a} = \mathbf{D}^T\mathbf{u}_2$ and $\Delta = 0$ if this results in $||\mathbf{w}_2||_1 \leq s$; otherwise, $\Delta > 0$ is chosen such that $||\mathbf{w}_2||_1 = s$.
4. Rewrite $\mathbf{u}_2$ as a $n \times n$ matrix, $\mathbf{U}_2$.
5. Perform hierarchical clustering on $\mathbf{U}_2$.

Of course, one could easily extend this procedure in order to obtain further complementary clusterings.

5.4 Example: Reanalysis of a breast cancer data set

In a well known paper, Perou et al. (2000) used gene expression microarrays to profile 65 surgical specimens of human breast tumors. Some of the samples were taken from the same tumor before and after chemotherapy. The data are available at `http://genome-www.stanford.edu/breast_cancer/molecularportraits/download.shtml`. The 65 samples were hierarchically clustered using what we will refer to as "Eisen" linkage; this is a centroid-based linkage that is implemented in Michael Eisen's `Cluster` program (Eisen et al. 1998). Two sets of genes were used for the clustering: the full set of 1753 genes, and an *intrinsic gene set* consisting of 496 genes. The intrinsic genes were defined as having the greatest level of variation in expression between different tumors relative to variation in

expression between paired samples taken from the same tumor before and after chemotherapy. The dendrogram obtained using the intrinsic gene set was used to identify four classes – basal-like, Erb-B2, normal breast-like, and ER+ – to which 62 of the 65 samples belong. It was determined that the remaining three observations did not belong to any of the four classes. These four classes are not visible in the dendrogram obtained using the full set of genes, and the authors concluded that the intrinsic gene set is necessary to observe the classes. In Figure 5.5, two dendrograms obtained by clustering on the intrinsic gene set are shown. The first was obtained by clustering all 65 observations, and the second was obtained by clustering the 62 observations that were assigned to one of the four classes. The former figure is in the original paper, and the latter is not. In particular, note that the four classes are not clearly visible in the dendrogram obtained using only 62 observations.

We wondered whether our proposal for sparse hierarchical clustering could yield a dendrogram that reflects the four classes, without any knowledge of the paired samples or of the intrinsic genes. We performed four versions of hierarchical clustering with Eisen linkage on the 62 observations that were assigned to the four classes:

1. Sparse hierarchical clustering of all 1753 genes, with the tuning parameter chosen to yield 496 nonzero genes.

2. Standard hierarchical clustering using all 1753 genes.

3. Standard hierarchical clustering using the 496 genes with highest marginal variance.

4. COSA hierarchical clustering using all 1753 genes.

The resulting dendrograms are shown in Figure 5.5. Sparse clustering of all 1753 genes with the tuning parameter chosen to yield 496 nonzero genes does best at capturing the four classes; in fact, a comparison with Figure 5.4 reveals that it does quite a bit better than clustering based on the intrinsic genes only! Figure 5.6 displays the result of performing the

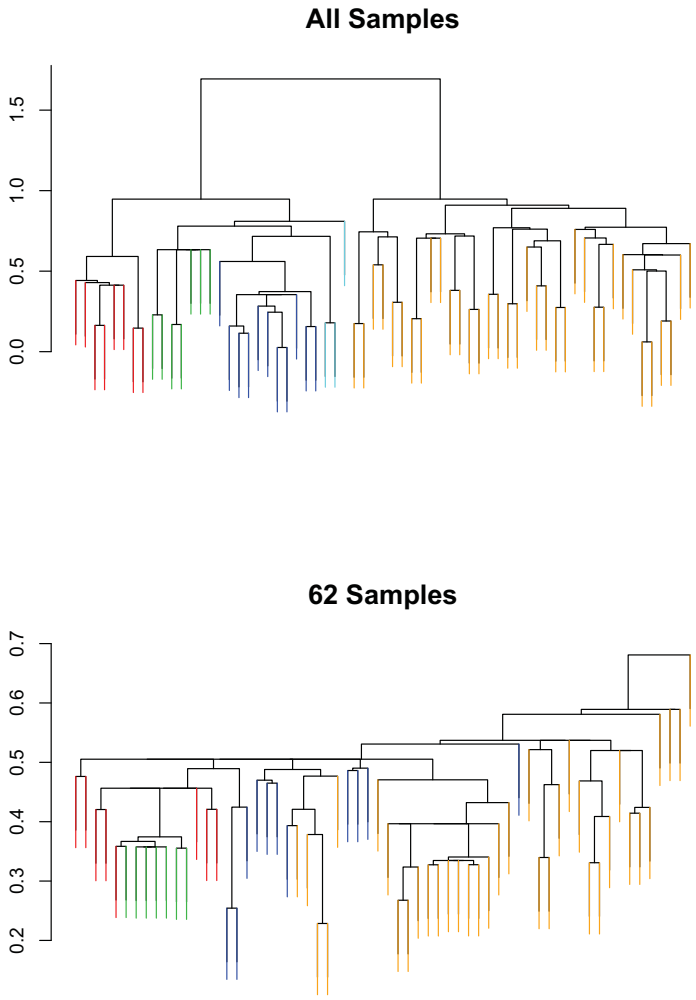

Figure 5.4: Using the intrinsic gene set, hierarchical clustering was performed on all 65 observations (top panel) and on only the 62 observations that were assigned to one of the four classes (bottom panel). Note that the classes identified using all 65 observations are largely lost in the dendrogram obtained using just 62 observations. The four classes are basal-like (red), Erb-B2 (green), normal breast-like (blue), and ER+ (orange). In the top panel, observations that do not belong to any class are shown in light blue.

automated tuning parameter selection method. This resulted in 93 genes having nonzero weights.

Figure 5.7 shows that the gene weights obtained using sparse clustering are highly correlated with the marginal variances of the genes. However, the results obtained from sparse clustering are different from the results obtained by simply clustering on the high variance genes (Figure 5.5). The reason for this lies in the form of the criterion (5.16). Though the nonzero w_j's tend to correspond to genes with high marginal variances, sparse clustering does not simply cluster the genes with highest marginal variances. Rather, it weights each gene-wise dissimilarity matrix by a different amount.

We also performed complementary sparse clustering on the full set of 1753 genes, using the method of Chapter 5.3.4. Tuning parameters for the initial and complementary sparse clusterings were selected to yield 496 genes with nonzero weights. The complementary sparse clustering dendrogram is shown in Figure 5.8, along with a plot of $\mathbf{w}_1$ and $\mathbf{w}_2$ (the feature weights for the initial and complementary clusterings). The dendrogram obtained using complementary sparse clustering suggests a previously unknown pattern in the data. Recall that the dendrogram for the initial sparse clustering can be found in Figure 5.5.

5.5 Example: HapMap Data

We wondered whether one could use sparse clustering in order to identify distinct populations in single nucleotide polymorphism (SNP) data, and also to identify the SNPs that differ between the populations. A SNP is a nucleotide position in a DNA sequence at which genetic variability exists in the population. We used the publicly available Haplotype Map ("HapMap") data of the International HapMap Consortium (International HapMap Consortium 2005, International HapMap Consortium 2007). We used the Phase III SNP data for chromosome 22, and restricted the analysis to three populations: African ancestry in southwest USA, Utah residents with European ancestry, and Han Chinese from Beijing.

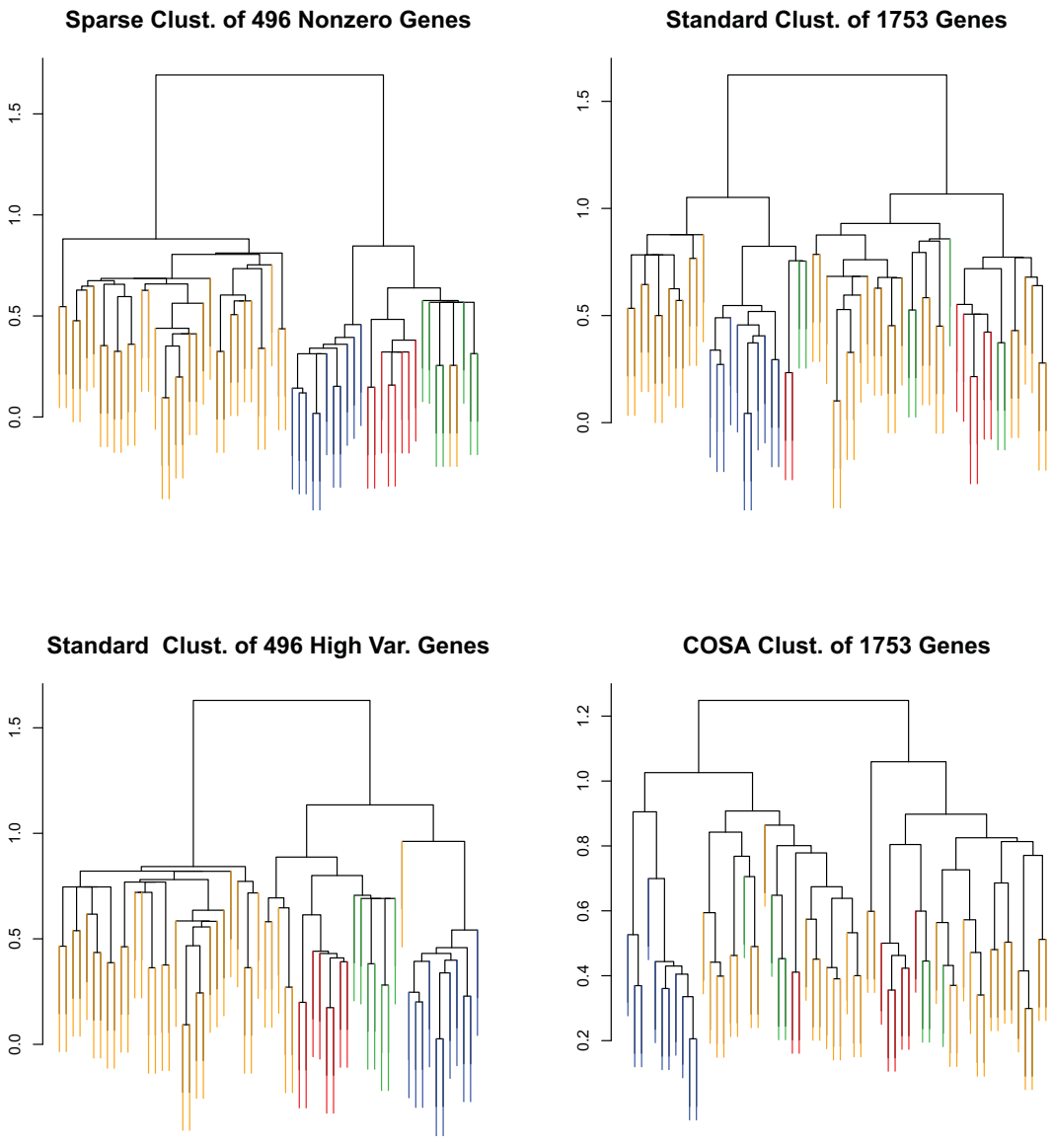

Figure 5.5: Four hierarchical clustering methods were used to cluster the 62 observations that were assigned to one of four classes in Perou et al. (2000). Sparse clustering results in the best separation between the four classes. The color coding is as in Figure 5.4.

Figure 5.6: The gap statistic was used to determine the optimal value of the tuning parameter for sparse hierarchical clustering. **Left:** The largest value of the gap statistic corresponds to 93 genes with nonzero weights. **Right:** The dendrogram corresponding to 93 nonzero weights. The color coding is as in Figure 5.4.

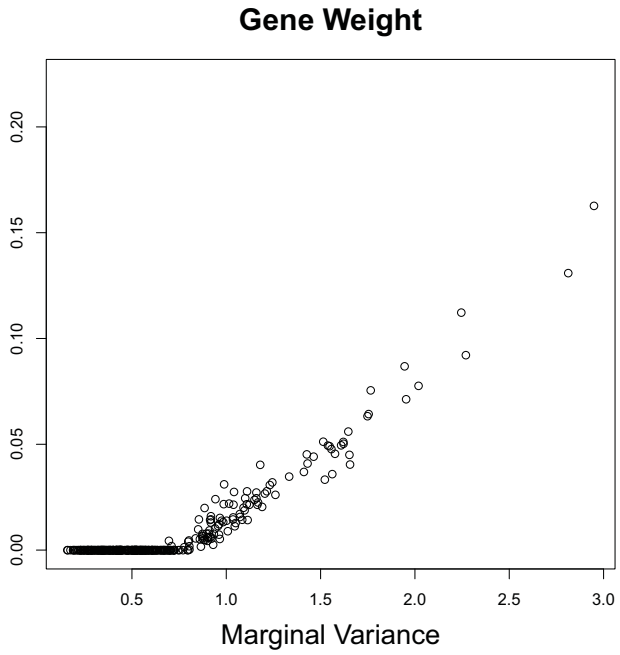

Figure 5.7: For each gene, the sparse clustering weight is plotted against the marginal variance.

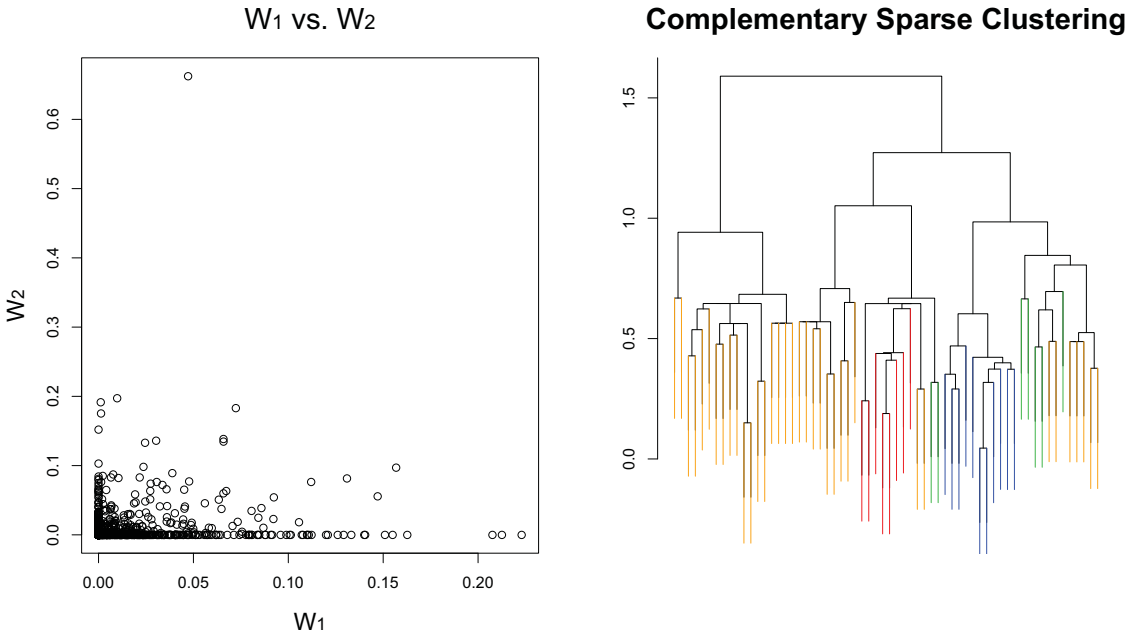

Figure 5.8: Complementary sparse clustering was performed. Tuning parameters for the initial and complementary clusterings were selected to yield 496 genes with nonzero weights. **Left:** A plot of $\mathbf{w}_1$ against $\mathbf{w}_2$. **Right:** The dendrogram for complementary sparse clustering. The color coding is as in Figure 5.4.

We used the SNPs for which measurements are available in all three populations. The resulting data have dimension 315×17026. We coded `AA` as 2, `Aa` as 1, and `aa` as 0. Missing values were imputed using 5-nearest neighbors (Troyanskaya et al. 2001). Sparse and standard 3-means clustering were performed on the data. The CERs obtained using standard 3-means and sparse 3-means are shown in Figure 5.9; CER was computed by comparing the clustering class labels to the true population identity for each sample. When the tuning parameter in sparse clustering was chosen to yield between 198 and 2809 SNPs with nonzero weights, sparse clustering resulted in slightly lower CER than standard 3-means clustering. The main advantage of sparse clustering over standard clustering is in interpretability, since the nonzero elements of $\mathbf{w}$ determine the SNPs involved in the sparse clustering. We can use the weights obtained from sparse clustering to identify SNPs on chromosome 22 that distinguish between the populations (Figure 5.9). SNPs in a few genomic regions appear to be responsible for the clustering obtained.

Based on Figure 5.9, it appears that for this data Algorithm 5.2 does not perform well. Rather than selecting a tuning parameter that yields between 198 and 2809 SNPs with nonzero weights (resulting in the lowest CER), the highest gap statistic is obtained when all SNPs are used. The one standard deviation rule in Algorithm 5.2 results in a tuning parameter that yields 7160 genes with nonzero weights. The fact that the gap statistic seemingly overestimates the number of features with nonzero weights may reflect the need for a more accurate method for tuning parameter selection, or it may suggest the presence of further population substructure beyond the three population labels.

In this example, we applied sparse clustering to SNP data for which the populations were already known. However, the presence of unknown subpopulations in SNP data is often a concern, as population substructure can confound attempts to identify SNPs that are associated with diseases and other outcomes (see e.g. Price et al. 2006). In general, one could use sparse clustering to identify subpopulations in SNP data in an unsupervised way before further analyses are performed.

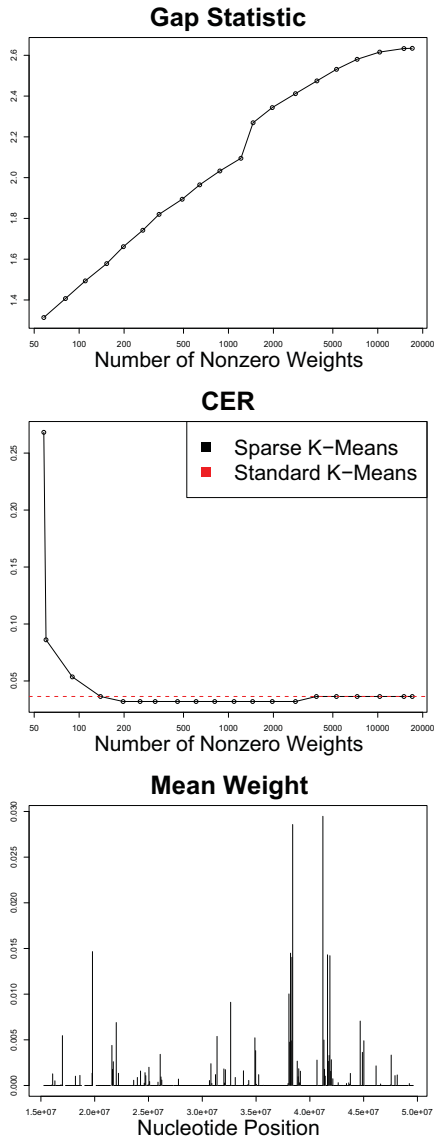

Figure 5.9: **Left:** The gap statistics obtained as a function of the number of SNPs with nonzero weights. **Center:** The CERs obtained using sparse and standard 3-means clustering, for a range of values of the tuning parameter. **Right:** Sparse clustering was performed using the tuning parameter that yields 198 nonzero SNPs. Chromosome 22 was split into 500 segments of equal length. The average weights of the SNPs in each segment are shown, as a function of the nucleotide position of the segments.

5.6 Additional comments

5.6.1 An additional remark on sparse K-means clustering

In the case where d is squared Euclidean distance, the K-means criterion (5.7) is equivalent to

$$\underset{C_1,\ldots,C_K,\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_K}{\text{minimize}}\{\sum_{k=1}^{K}\sum_{i\in C_k} d(\mathbf{x}_i,\boldsymbol{\mu}_k)\} \tag{5.29}$$

where $\boldsymbol{\mu}_k$ is the centroid for cluster k. However, if d is not squared Euclidean distance - for instance, if d is the sum of the absolute differences - then (5.7) and (5.29) are not equivalent. We used the criterion (5.7) to define K-means clustering, and consequently to derive a method for sparse K-means clustering, for simplicity and consistency with the COSA method of Friedman & Meulman (2004). But if (5.29) is used to define K-means clustering and the dissimilarity measure is not squared Euclidean distance (but is still additive in the features), then an analogous criterion and algorithm for sparse K-means clustering can be derived instead. In practice, this is not an important distinction, since K-means clustering is generally performed using squared distance as the dissimilarity measure.

5.6.2 Sparse K-medoids clustering

In Chapter 5.1.3, we mentioned that any clustering method of the form (5.4) could be modified to obtain a sparse clustering method of the form (5.5). (However, for the resulting sparse method to have a nonzero weight for feature j, it is necessary that $f_j(\mathbf{X}_j,\boldsymbol{\Theta}) > 0$.) In addition to K-means and sparse hierarchical clustering, another method that takes the form (5.4) is K-medoids. Let $i_k \in \{1,\ldots,n\}$ denote the index of the observation that serves as the medoid for cluster k, and let C_k denote the indices of the observations in cluster k. The K-medoids criterion is

$$\underset{C_1,\ldots,C_K,i_1,\ldots,i_K}{\text{minimize}}\{\sum_{k=1}^{K}\sum_{i\in C_k}\sum_{j=1}^{p} d_{i,i_k,j}\}, \tag{5.30}$$

or equivalently

$$\underset{C_1,\ldots,C_K,i_1,\ldots,i_K}{\text{maximize}} \{\sum_{j=1}^{p}(\sum_{i=1}^{n} d_{i,i_0,j} - \sum_{k=1}^{K}\sum_{i\in C_k} d_{i,i_k,j})\} \tag{5.31}$$

where $i_0 \in \{1,\ldots,n\}$ is the index of the medoid for the full set of n observations. Since (5.31) is of the form (5.4), the criterion

$$\begin{aligned}&\underset{\mathbf{w},C_1,\ldots,C_K,i_1,\ldots,i_K}{\text{maximize}} \{\sum_{j=1}^{p} w_j(\sum_{i=1}^{n} d_{i,i_0,j} - \sum_{k=1}^{K}\sum_{i\in C_k} d_{i,i_k,j})\}\\ &\text{subject to } ||\mathbf{w}||^2 \leq 1, ||\mathbf{w}||_1 \leq s, w_j \geq 0\ \forall j\end{aligned} \tag{5.32}$$

results in a method for sparse K-medoids clustering, which can be optimized by an iterative approach.

5.6.3 A dissimilarity matrix that is sparse in the features

The proposal of Chapter 5.3 for sparse hierarchical clustering involves computing a dissimilarity matrix that involves only a subset of the features. That dissimilarity matrix is then used as input for a standard hierarchical clustering procedure. In fact, nothing about the proposal for obtaining the reweighted dissimilarity matrix is specific to hierarchical clustering. Using this approach, one could obtain a sparse version of any statistical method that takes as its input a dissimilarity matrix. For instance, a sparse version of least squares multidimensional scaling (see e.g. Borg & Groenen 2005) could be obtained using this approach. Similarly, one could develop a sparse version of spectral clustering (see e.g. von Luxburg 2007).

Chapter 6

Penalized linear discriminant analysis

In this chapter, we develop a proposal for extending linear discriminant analysis to the high-dimensional setting by applying a variant of the PMD to the between-class covariance matrix of the features.

6.1 Linear discriminant analysis in high dimensions

In this chapter, we consider the classification setting. The data consist of a $n \times p$ matrix $\mathbf{X}$ with p features measured on n observations, each of which belongs to one of K classes. Linear discriminant analysis (LDA) is a well-known method for this problem in the classical setting where $n > p$. However, in high dimensions (when the number of features is large relative to the number of observations) LDA faces two problems:

1. The maximum likelihood estimate of the within-class covariance matrix is approximately singular (if p is almost as large as n) or singular (if $p > n$). Even if the estimate is not singular, the resulting classifier can suffer from high variance, resulting in poor performance.

2. When p is large, the resulting classifier is difficult to interpret, since the classification rule involves a linear combination of all p features.

The LDA classifier can be derived in three different ways, which we will refer to as the *maximum likelihood problem*, the *optimal scoring problem*, and *Fisher's discriminant problem* (see e.g. Mardia et al. 1979, Hastie et al. 2009). In recent years, a number of papers have extended LDA to the high-dimensional setting in such a way that the resulting classifier involves a sparse linear combination of the features (see e.g. Tibshirani et al. 2002, Tibshirani et al. 2003, Grosenick et al. 2008, Leng 2008, Clemmensen et al. 2010). These methods involve *regularizing* or *penalizing* the maximum likelihood problem or the optimal scoring problem by applying an L_1 or lasso penalty (Tibshirani 1996).

Here, we take a different approach. We regularize Fisher's discriminant problem, which is in our opinion the most natural of the three problems that result in LDA. The resulting problem is highly nonconvex and difficult to optimize. We overcome this difficulty using a minorization-maximization approach (see e.g. Lange et al. 2000, Hunter & Lange 2004, Lange 2004), which allows us to solve the problem efficiently when convex penalties are applied to the discriminant vectors. This is equivalent to recasting Fisher's discriminant problem as a biconvex problem that can be optimized using a simple iterative algorithm. Our approach has some advantages over competing methods:

1. It results from a natural criterion for which a simple optimization strategy is provided.

2. A reduced rank solution can be obtained.

3. It provides a natural way to enforce a diagonal estimate for the within-class covariance matrix, which has been shown to yield good results in the high-dimensional setting (see e.g. Dudoit et al. 2001, Tibshirani et al. 2003, Bickel & Levina 2004).

4. It yields interpretable discriminant vectors, where the concept of interpretability can be chosen based on the problem at hand. Interpretability is achieved via application

of convex penalties to the discriminant vectors. For instance, if L_1 penalties are used, then the resulting discriminant vectors are sparse.

6.2 Fisher's discriminant problem

6.2.1 Fisher's discriminant problem when $n > p$

Let $\mathbf{X}$ be a $n \times p$ matrix with observations on the rows and features on the columns. We assume that the features are centered to have mean zero, and we let $\mathbf{X}_j$ denote feature/column j and $\mathbf{x}_i$ denote observation/row i. $C_k \subset \{1, \ldots, n\}$ contains the indices of the observations in class k, and $n_k = |C_k|, \sum_{k=1}^{K} n_k = n$.

Fisher's discriminant problem involves seeking a low-dimensional projection of the observations such that the between-class variance is large relative to the within-class variance. That is, we sequentially solve

$$\underset{\boldsymbol{\beta}_k \in \mathbb{R}^p}{\text{maximize}} \{\boldsymbol{\beta}_k^T \hat{\boldsymbol{\Sigma}}_b \boldsymbol{\beta}_k\} \text{ subject to } \boldsymbol{\beta}_k^T \hat{\boldsymbol{\Sigma}}_w \boldsymbol{\beta}_k \leq 1, \boldsymbol{\beta}_k^T \hat{\boldsymbol{\Sigma}}_w \boldsymbol{\beta}_i = 0 \ \ \forall i < k. \tag{6.1}$$

Note that the problem (6.1) is generally written with the inequality constraint replaced with an equality constraint, but the two are equivalent if $\hat{\boldsymbol{\Sigma}}_w$ has full rank. We will refer to the solution to (6.1) as the *kth discriminant vector*. Here, $\hat{\boldsymbol{\Sigma}}_b = \frac{1}{n}\mathbf{X}^T\mathbf{X} - \hat{\boldsymbol{\Sigma}}_w$ is the *between-class covariance matrix*, with the *within-class covariance matrix* $\hat{\boldsymbol{\Sigma}}_w$ given by

$$\hat{\boldsymbol{\Sigma}}_w = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in C_k} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^T \tag{6.2}$$

and $\hat{\boldsymbol{\mu}}_k$ the mean vector for class k. Later, we will make use of the fact that $\hat{\boldsymbol{\Sigma}}_b = \frac{1}{n}\mathbf{X}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{X}$, where $\mathbf{Y}$ is a $n \times K$ matrix with Y_{ik} an indicator of whether observation i is in class k.

A classification rule is obtained by computing $\mathbf{X}\hat{\boldsymbol{\beta}}_1, \ldots, \mathbf{X}\hat{\boldsymbol{\beta}}_{K-1}$ and assigning each observation to its nearest centroid in this transformed space. Alternatively, one can transform the observations using only the first $k < K-1$ discriminant vectors in order to perform *reduced rank classification*. LDA derives its name from the fact that the classification rule involves a linear combination of the features.

One can solve (6.1) by substituting $\tilde{\boldsymbol{\beta}}_k = \hat{\boldsymbol{\Sigma}}_w^{\frac{1}{2}} \boldsymbol{\beta}_k$, where $\hat{\boldsymbol{\Sigma}}_w^{\frac{1}{2}}$ is the symmetric matrix square root of $\hat{\boldsymbol{\Sigma}}_w$. Then, Fisher's discriminant problem is reduced to a standard eigenproblem.

6.2.2 Past proposals for extending Fisher's discriminant problem to $p > n$

In high dimensions, there are two reasons that problem (6.1) does not lead to a suitable classifier:

1. $\hat{\boldsymbol{\Sigma}}_w$ is singular. A discriminant vector that is in the null space of $\hat{\boldsymbol{\Sigma}}_w$ but not in the null space of $\hat{\boldsymbol{\Sigma}}_b$ can result in an arbitrarily large between-class variance.

2. The resulting classifer is not interpretable when p is very large, because the discriminant vectors contain p elements that have no particular structure.

A number of modifications to Fisher's discriminant problem have been proposed to address the singularity problem. Krzanowski et al. (1995) and Tebbens & Schlesinger (2007) consider modifying (6.1) by instead seeking a unit vector $\boldsymbol{\beta}$ that maximizes $\boldsymbol{\beta}^T \hat{\boldsymbol{\Sigma}}_b \boldsymbol{\beta}$ subject to $\boldsymbol{\beta}^T \hat{\boldsymbol{\Sigma}}_w \boldsymbol{\beta} = 0$. Others have proposed modifying (6.1) by using a positive definite estimate of $\boldsymbol{\Sigma}_w$. For instance, Friedman (1989), Dudoit et al. (2001), and Bickel & Levina (2004) consider the use of the diagonal estimate

$$\text{diag}(\hat{\sigma}_1^2, \ldots, \hat{\sigma}_p^2), \tag{6.3}$$

where $\hat{\sigma}_j^2$ is the jth diagonal element of $\hat{\boldsymbol{\Sigma}}_w$ (6.2). Other positive definite estimates for $\boldsymbol{\Sigma}_w$ are suggested in Krzanowski et al. (1995) and Xu et al. (2009). The resulting criterion is

$$\underset{\boldsymbol{\beta}_k \in \mathbb{R}^p}{\text{maximize}}\{\boldsymbol{\beta}_k^T \hat{\boldsymbol{\Sigma}}_b \boldsymbol{\beta}_k\} \text{ subject to } \boldsymbol{\beta}_k^T \tilde{\boldsymbol{\Sigma}}_w \boldsymbol{\beta}_k \leq 1, \boldsymbol{\beta}_k^T \tilde{\boldsymbol{\Sigma}}_w \boldsymbol{\beta}_i = 0 \ \ \forall i < k, \tag{6.4}$$

where $\tilde{\boldsymbol{\Sigma}}_w$ is a positive definite estimate for $\boldsymbol{\Sigma}_w$. The criterion (6.4) addresses the singularity issue, but not the interpretability issue. Here, we extend (6.4) so that the resulting discriminant vectors are interpretable.

Note that in this chapter, $\hat{\boldsymbol{\Sigma}}_w$ will always refer to the standard estimate of $\boldsymbol{\Sigma}_w$ (6.2), whereas $\tilde{\boldsymbol{\Sigma}}_w$ will refer to some positive definite estimate of $\boldsymbol{\Sigma}_w$ for which the specific form will depend on the context.

6.3 The penalized LDA proposal

6.3.1 First penalized LDA discriminant vector

We would like to modify the problem (6.4) by imposing penalty functions on the discriminant vectors. We define the *first penalized discriminant vector* to be the solution to the problem

$$\underset{\boldsymbol{\beta}_1}{\text{maximize}}\{\boldsymbol{\beta}_1^T \hat{\boldsymbol{\Sigma}}_b \boldsymbol{\beta}_1 - P(\boldsymbol{\beta}_1)\} \text{ subject to } \boldsymbol{\beta}_1^T \tilde{\boldsymbol{\Sigma}}_w \boldsymbol{\beta}_1 \leq 1, \tag{6.5}$$

where $\tilde{\boldsymbol{\Sigma}}_w$ is a positive definite estimate for $\boldsymbol{\Sigma}_w$ and where P is a convex penalty function.

To obtain multiple discriminant vectors, we must extend (6.5). Rather then doing so by direct analogy to (6.1), which would involve requiring that $\boldsymbol{\beta}_k^T \hat{\boldsymbol{\Sigma}}_w \boldsymbol{\beta}_i = 0$ for all $i < k$, we instead notice that the kth unpenalized discriminant vector maximizes a modified between-class variance, obtained by projecting onto the subspace that is orthogonal to the previous discriminant vectors. This is explained in the following proposition.

Proposition 6.3.1. *The kth unpenalized discriminant vector is the solution to the problem*

$$\underset{\boldsymbol{\beta}_k}{\text{maximize}}\{\boldsymbol{\beta}_k^T \hat{\boldsymbol{\Sigma}}_b^k \boldsymbol{\beta}_k\} \text{ subject to } \boldsymbol{\beta}_k^T \tilde{\boldsymbol{\Sigma}}_w \boldsymbol{\beta}_k \leq 1 \tag{6.6}$$

where

$$\hat{\boldsymbol{\Sigma}}_b^k = \frac{1}{n}\mathbf{X}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\mathbf{P}_k^{\perp}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{X}, \tag{6.7}$$

and where $\mathbf{P}_k^{\perp}$ *is an orthogonal projection matrix into the space that is orthogonal to* $\boldsymbol{\beta}_i^T \tilde{\boldsymbol{\Sigma}}_w^{-\frac{1}{2}} \mathbf{X}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}$ *for all* $i < k$.

So we define the *kth penalized discriminant vector* to be the solution to

$$\underset{\boldsymbol{\beta}_k}{\text{maximize}}\{\boldsymbol{\beta}_k^T \hat{\boldsymbol{\Sigma}}_b^k \boldsymbol{\beta}_k - P_k(\boldsymbol{\beta}_k)\} \text{ subject to } \boldsymbol{\beta}_k^T \tilde{\boldsymbol{\Sigma}}_w \boldsymbol{\beta}_k \leq 1, \tag{6.8}$$

where $\hat{\boldsymbol{\Sigma}}_b^k$ is given by (6.7) with $\mathbf{P}_k^{\perp}$ an orthogonal projection matrix into the space that is orthogonal to $\boldsymbol{\beta}_i^T \tilde{\boldsymbol{\Sigma}}_w^{-\frac{1}{2}} \mathbf{X}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}$ for all $i < k$. Note that (6.5) is a special case of (6.8). Here P_k is a convex penalty function. When $P_1 = \ldots = P_k = 0$ then $\hat{\boldsymbol{\beta}}_k{}^T \tilde{\boldsymbol{\Sigma}}_w \hat{\boldsymbol{\beta}}_i = 0 \; \forall i < k$, where $\hat{\boldsymbol{\beta}}_k$ is the solution to (6.8). But for general P_k we do not expect orthogonality of the subsequent penalized discriminant vectors.

In general, the problem (6.8) is not convex, because the objective is not concave. We apply a minorization algorithm to solve it. For any positive semidefinite matrix $\mathbf{A}$, $f(\boldsymbol{\beta}) = \boldsymbol{\beta}^T\mathbf{A}\boldsymbol{\beta}$ is convex in $\boldsymbol{\beta}$. Thus, for a fixed value of $\boldsymbol{\beta}^{(m)}$,

$$f(\boldsymbol{\beta}) \geq f(\boldsymbol{\beta}^{(m)}) + (\boldsymbol{\beta} - \boldsymbol{\beta}^{(m)})^T \nabla f(\boldsymbol{\beta}^{(m)}) = 2\boldsymbol{\beta}^T\mathbf{A}\boldsymbol{\beta}^{(m)} - \boldsymbol{\beta}^{(m)T}\mathbf{A}\boldsymbol{\beta}^{(m)} \tag{6.9}$$

for any $\boldsymbol{\beta}$, and equality holds when $\boldsymbol{\beta} = \boldsymbol{\beta}^{(m)}$. Therefore,

$$g(\boldsymbol{\beta}_k, \boldsymbol{\beta}^{(m)}) = 2\boldsymbol{\beta}_k^T \hat{\boldsymbol{\Sigma}}_b^k \boldsymbol{\beta}^{(m)} - \boldsymbol{\beta}^{(m)T} \hat{\boldsymbol{\Sigma}}_b^k \boldsymbol{\beta}^{(m)} - P_k(\boldsymbol{\beta}_k) \tag{6.10}$$

minorizes the objective of (6.8) at $\boldsymbol{\beta}^{(m)}$. Moreover, since P_k is a convex function, g is concave in $\boldsymbol{\beta}_k$ and hence can be maximized using convex optimization tools.

Algorithm 6.1: Obtaining the kth penalized discriminant vector

1. If $k > 1$, define an orthogonal projection matrix $\mathbf{P}_k^{\perp}$ that projects onto the space that is orthogonal to $\hat{\boldsymbol{\beta}}_i^T \tilde{\boldsymbol{\Sigma}}_w^{-\frac{1}{2}} \mathbf{X}^T \mathbf{Y} (\mathbf{Y}^T \mathbf{Y})^{-\frac{1}{2}}$ for all $i < k$. Let $\mathbf{P}_1^{\perp} = \mathbf{I}$.
2. Let $\hat{\boldsymbol{\Sigma}}_b^k = \frac{1}{n} \mathbf{X}^T \mathbf{Y} (\mathbf{Y}^T \mathbf{Y})^{-\frac{1}{2}} \mathbf{P}_k^{\perp} (\mathbf{Y}^T \mathbf{Y})^{-\frac{1}{2}} \mathbf{Y}^T \mathbf{X}$. Note that $\hat{\boldsymbol{\Sigma}}_b^1 = \hat{\boldsymbol{\Sigma}}_b$.
3. Let $\boldsymbol{\beta}_k^{(0)}$ be the kth eigenvector of $\tilde{\boldsymbol{\Sigma}}_w^{-1} \hat{\boldsymbol{\Sigma}}_b$; this is simply the kth unpenalized discriminant vector.
4. For $m = 1, 2, \ldots$: Let $\boldsymbol{\beta}_k^{(m)}$ be the solution to

$$\underset{\boldsymbol{\beta}_k}{\text{maximize}} \{ 2 \boldsymbol{\beta}_k^T \hat{\boldsymbol{\Sigma}}_b^k \boldsymbol{\beta}_k^{(m-1)} - P_k(\boldsymbol{\beta}_k) \} \text{ subject to } \boldsymbol{\beta}_k^T \tilde{\boldsymbol{\Sigma}}_w \boldsymbol{\beta}_k \leq 1. \tag{6.11}$$

Of course, the solution to (6.11) will depend on the form of the convex function P_k.

6.3.2 Penalized LDA-L_1

We define *penalized LDA-L_1* to be the solution to (6.8) with an L_1 penalty,

$$\underset{\boldsymbol{\beta}_k}{\text{maximize}} \{ \boldsymbol{\beta}_k^T \hat{\boldsymbol{\Sigma}}_b^k \boldsymbol{\beta}_k - \lambda_k \sum_{j=1}^{p} |\hat{\sigma}_j \beta_{kj}|) \} \text{ subject to } \boldsymbol{\beta}_k^T \tilde{\boldsymbol{\Sigma}}_w^k \boldsymbol{\beta}_k \leq 1. \tag{6.12}$$

When the tuning parameter λ_k is large, some elements of the solution $\hat{\boldsymbol{\beta}}_k$ will be exactly equal to zero. The inclusion of $\hat{\sigma}_j$ in the penalty has the effect that features that vary more within each class undergo greater penalization. Penalized LDA-L_1 is appropriate if we want to obtain a sparse classifier.

To solve (6.12), we use the minorization approach outlined in Algorithm 6.1. Step 4 can be written as

$$\underset{\boldsymbol{\beta}_k}{\text{maximize}} \{ 2 \boldsymbol{\beta}_k^T \hat{\boldsymbol{\Sigma}}_b^k \boldsymbol{\beta}_k^{(m-1)} - \lambda_k \sum_{j=1}^{p} |\hat{\sigma}_j \beta_{kj}|) \} \text{ subject to } \boldsymbol{\beta}_k^T \tilde{\boldsymbol{\Sigma}}_w \boldsymbol{\beta}_k \leq 1. \tag{6.13}$$

Now (6.13) is a convex problem, and so the Karush-Kuhn-Tucker (KKT) conditions (see e.g. Boyd & Vandenberghe 2004) imply that necessary and sufficient conditions for a solution are as follows:

$$2\hat{\boldsymbol{\Sigma}}_b^k \boldsymbol{\beta}_k^{(m-1)} - \lambda_k \Gamma - 2\delta \tilde{\boldsymbol{\Sigma}}_w \boldsymbol{\beta}_k = 0, \;\; \delta \geq 0, \;\; \delta(\boldsymbol{\beta}_k^T \tilde{\boldsymbol{\Sigma}}_w \boldsymbol{\beta}_k - 1) = 0, \;\; \boldsymbol{\beta}_k^T \tilde{\boldsymbol{\Sigma}}_w \boldsymbol{\beta}_k \leq 1, \quad (6.14)$$

where Γ is a subgradient of $\sum_{j=1}^p |\hat{\sigma}_j \beta_{kj}|$. When $\tilde{\boldsymbol{\Sigma}}_w$ is the diagonal estimate (6.3), then the solution to (6.13), and therefore Step 4 of Algorithm 6.1, is particularly simple:

Algorithm 6.2: Solving the minorization step for penalized LDA-L_1

1. Compute $\mathbf{a} = 2\hat{\boldsymbol{\Sigma}}_b^k \boldsymbol{\beta}_k^{(m-1)}$.
2. For $j = 1, \ldots, p$, let $d_j = \frac{1}{\hat{\sigma}_j^2} S(a_j, \lambda_k \hat{\sigma}_j)$ where S is the soft-thresholding operator (1.8).
3. The solution $\hat{\boldsymbol{\beta}}_k$ to (6.13) is as follows:

$$\hat{\boldsymbol{\beta}}_k = \begin{cases} 0 & \text{if } \mathbf{d}=0. \\ \frac{\mathbf{d}}{\sqrt{\sum_{j=1}^p \hat{\sigma}_j^2 d_j^2}} & \text{otherwise.} \end{cases}$$

We now consider the problem of selecting the tuning parameter λ_k. The simplest approach would be to take $\lambda_k = \lambda \; \forall k$. However, this results in effectively penalizing each component more than the previous components, since the largest eigenvalue of $\hat{\boldsymbol{\Sigma}}_b^k$ is nonincreasing in k. Therefore, we instead take $\lambda_k = \lambda ||\hat{\boldsymbol{\Sigma}}_b^k||$ where $||\cdot||$ is the largest eigenvalue. The value of λ can be chosen by cross-validation.

We use the fact that $\hat{\boldsymbol{\Sigma}}_b^k$ has low rank in order to quickly perform Step 1 of Algorithm 6.2 and calculate the largest eigenvalue of $\hat{\boldsymbol{\Sigma}}_b^k$.

6.3.3 Penalized LDA-FL

We define *penalized LDA-FL* to be the solution to the problem (6.8) with a fused lasso penalty (Tibshirani et al. 2005):

$$\begin{aligned} &\underset{\boldsymbol{\beta}_k}{\text{maximize}}\{\boldsymbol{\beta}_k^T\hat{\boldsymbol{\Sigma}}_b^k\boldsymbol{\beta}_k - \lambda_k\sum_{j=1}^{p}|\hat{\sigma}_j\beta_{kj}| - \gamma_k\sum_{j=2}^{p}|\hat{\sigma}_j\beta_{kj} - \hat{\sigma}_{j-1}\beta_{k,j-1}|\} \\ &\text{subject to } \boldsymbol{\beta}_k^T\tilde{\boldsymbol{\Sigma}}_w\boldsymbol{\beta}_k \leq 1. \end{aligned} \tag{6.15}$$

When the nonnegative tuning parameter λ_k is large then the resulting discriminant vector will be sparse in the features, and when the nonnegative tuning parameter γ_k is large then the discriminant vector will be piecewise constant. This classifier is appropriate if the features are ordered on a line, and one believes that the true underlying signal is sparse and piecewise constant.

To solve (6.8), we again apply Algorithm 6.1. Step 4 can be written as

$$\begin{aligned} &\underset{\boldsymbol{\beta}_k}{\text{maximize}}\{2\boldsymbol{\beta}_k^T\hat{\boldsymbol{\Sigma}}_b^k\boldsymbol{\beta}_k^{(m-1)} - \lambda_k\sum_{j=1}^{p}|\hat{\sigma}_j\beta_{kj}| - \gamma_k\sum_{j=2}^{p}|\hat{\sigma}_j\beta_{kj} - \hat{\sigma}_{j-1}\beta_{k,j-1}|\} \\ &\text{subject to } \boldsymbol{\beta}_k^T\tilde{\boldsymbol{\Sigma}}_w\boldsymbol{\beta}_k \leq 1. \end{aligned} \tag{6.16}$$

By the KKT conditions, the following steps yield the solution to (6.16) and therefore Step 4 of Algorithm 6.1 for penalized LDA-FL with $\tilde{\boldsymbol{\Sigma}}_w$ the diagonal estimate (6.3):

Algorithm 6.3: Solving the minorization step for penalized LDA-FL

1. Compute $\mathbf{a} = 2\hat{\boldsymbol{\Sigma}}_b^k\boldsymbol{\beta}_k^{(m-1)}$.

2. Let $\hat{\mathbf{d}}$ denote the solution to the problem

$$\underset{\mathbf{d}\in\mathbb{R}^p}{\text{minimize}}\{\sum_{j=1}^{p}d_j^2\hat{\sigma}_j^2 - \mathbf{d}^T\mathbf{a} + \lambda_k\sum_{j=1}^{p}|\hat{\sigma}_j d_j| + \gamma_k\sum_{j=2}^{p}|\hat{\sigma}_j d_j - \hat{\sigma}_{j-1}d_{j-1}|\}.$$

3. The solution $\hat{\boldsymbol{\beta}}_k$ to (6.16) is as follows:

$$\hat{\boldsymbol{\beta}}_k = \begin{cases} 0 & \text{if } \hat{\mathbf{d}} = 0. \\ \frac{\hat{\mathbf{d}}}{\sqrt{\sum_{j=1}^{p} \hat{\sigma}_j^2 \hat{d}_j^2}} & \text{otherwise.} \end{cases}$$

We choose λ_k and γ_k by fixing nonnegative constants λ and γ. Then, we take $\lambda_k = \lambda||\hat{\boldsymbol{\Sigma}}_b^k||$ and $\gamma_k = \gamma||\hat{\boldsymbol{\Sigma}}_b^k||$ where $||\cdot||$ indicates the largest eigenvalue. Note that fast software exists to perform Step 2 (see e.g. Hoefling 2009*a*).

6.3.4 Recasting penalized LDA as a biconvex problem

It turns out that one could instead solve the nonconvex problem (6.5) by recasting it as a biconvex problem. Consider the problem

$$\underset{\boldsymbol{\beta},\mathbf{u}}{\text{maximize}}\{\frac{2}{\sqrt{n}}\boldsymbol{\beta}^T\mathbf{X}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\mathbf{u} - P(\boldsymbol{\beta}) - \mathbf{u}^T\mathbf{u}\} \text{ subject to } \boldsymbol{\beta}^T\tilde{\boldsymbol{\Sigma}}_w\boldsymbol{\beta} \leq 1. \tag{6.17}$$

Partially optimizing (6.17) with respect to $\mathbf{u}$ reveals that the $\boldsymbol{\beta}$ that solves it also solves (6.5). Moreover, (6.17) is a biconvex problem. This suggests a simple iterative approach for solving it. We repeatedly hold $\boldsymbol{\beta}$ fixed and solve with respect to $\mathbf{u}$, and then hold $\mathbf{u}$ fixed and solve with respect to $\boldsymbol{\beta}$.

Algorithm 6.4: A biconvex formulation for penalized LDA

1. Let $\boldsymbol{\beta}^{(0)}$ be the first eigenvector of $\tilde{\boldsymbol{\Sigma}}_w^{-1}\hat{\boldsymbol{\Sigma}}_b$.

2. For $m = 1, 2, \ldots$:

   (a) Let $\mathbf{u}^{(m)}$ solve

$$\underset{\mathbf{u}}{\text{maximize}}\{\frac{2}{\sqrt{n}}{\boldsymbol{\beta}^{(m-1)}}^T\mathbf{X}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\mathbf{u} - \mathbf{u}^T\mathbf{u}\}. \tag{6.18}$$

(b) Let $\boldsymbol{\beta}^{(m)}$ solve

$$\underset{\boldsymbol{\beta}}{\text{maximize}}\{\frac{2}{\sqrt{n}}\boldsymbol{\beta}^T\mathbf{X}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\mathbf{u}^{(m)} - P(\boldsymbol{\beta})\} \text{ subject to } \boldsymbol{\beta}^T\tilde{\boldsymbol{\Sigma}}_w\boldsymbol{\beta} \leq 1. \quad (6.19)$$

Combining steps 2(a) and 2(b), we see that $\boldsymbol{\beta}^{(m)}$ solves

$$\underset{\boldsymbol{\beta}}{\text{maximize}}\{2\boldsymbol{\beta}^T\hat{\boldsymbol{\Sigma}}_b\boldsymbol{\beta}^{(m-1)} - P(\boldsymbol{\beta})\} \text{ subject to } \boldsymbol{\beta}^T\tilde{\boldsymbol{\Sigma}}_w\boldsymbol{\beta} \leq 1. \quad (6.20)$$

Comparing (6.20) to (6.11), we see that the biconvex formulation (6.17) results in the same algorithm as the minorization approach outlined in Algorithm 6.1 for finding the first penalized discriminant vector.

6.3.5 Connection with the PMD

Our proposal for penalized LDA (6.5) is quite similar to what one would obtain by applying the PMD to the matrix $\hat{\boldsymbol{\Sigma}}_b$ with an arbitrary convex penalty. There are a few main differences:

1. In (6.5), rather than a constraint of the form $||\boldsymbol{\beta}_k||^2 \leq 1$, there is a constraint of the form $\boldsymbol{\beta}_k^T\tilde{\boldsymbol{\Sigma}}_w\boldsymbol{\beta}_k \leq 1$.

2. Rather than a bound form for the penalty on $\boldsymbol{\beta}_k$, the Lagrange form is used in (6.5) in order to obtain a computationally faster algorithm.

The close connection between the PMD and penalized LDA stems from the fact that Fisher's discriminant problem is simply a generalized eigenproblem.

6.4 Examples

6.4.1 A simulation study

We compare penalized LDA to nearest shrunken centroids (NSC) and sparse discriminant analysis (SDA) in a simulation study. NSC and SDA are described in Chapter 6.5.3. Briefly, NSC results from using a diagonal estimate of $\mathbf{\Sigma}_w$ and imposing L_1 penalties on the class mean vectors in the maximum likelihood problem, and SDA results from applying an elastic net penalty to the discriminant vectors in the optimal scoring problem. Three simulations were considered. In each simulation, there are 1200 observations, equally split between the classes. Of these 1200 observations, 100 belong to the training set, 100 belong to the test set, and 1000 are in the validation set. Each simulation consists of measurements on 1000 features, of which 200 differ between classes.

Simulation 1. Mean shift with independent features. There are four classes. If observation i is in class k, then $\mathbf{x}_i \sim N(\boldsymbol{\mu}_k, \mathbf{I})$, where $\mu_{1j} = 0.7 \times 1_{(1 \leq j \leq 50)}, \mu_{2j} = 0.7 \times 1_{(51 \leq j \leq 100)}, \mu_{3j} = 0.7 \times 1_{(101 \leq j \leq 150)}, \mu_{4j} = 0.7 \times 1_{(151 \leq j \leq 200)}$.

Simulation 2. Mean shift with dependent features. There are two classes. For $i \in C_1$, $\mathbf{x}_i \sim N(0, \mathbf{\Sigma})$ and for $i \in C_2$, $\mathbf{x}_i \sim N(\boldsymbol{\mu}, \mathbf{\Sigma})$, $\mu_j = 0.4 \times 1_{(j \leq 200)}$. The covariance structure is block diagonal, with 10 blocks each of dimension 100×100. The blocks have (j, j') element $\rho^{|j-j'|}$ where $\rho = 0.6$. This covariance structure is intended to mimic gene expression data, in which sets of genes are positively correlated with each other and different pathways are independent of each other.

Simulation 3. One-dimensional mean shift with independent features. There are four classes, and the features are independent. For $i \in C_k$, $X_{ij} \sim N(\frac{k-1}{3}, 1)$ if $j \leq 200$, and $X_{ij} \sim N(0, 1)$ otherwise. Note that a one-dimensional projection of the data fully captures the class structure.

Figure 6.1 displays the class mean vectors for each simulation.

For each method, models were fit on the training set using a range of tuning parameter values. Tuning parameter values were then selected to minimize the test set error. Finally, the training set models with appropriate tuning parameter values were evaluated on the validation set. For penalized LDA-L_1, λ was a tuning parameter. For penalized LDA-FL, we treated $\lambda = \gamma$ as a single tuning parameter in order to avoid performing cross-validation on a two-dimensional grid. NSC has a single tuning parameter, which corresponds to the amount of soft-thresholding performed. To avoid performing cross-validation on a two-dimensional grid, SDA was performed with $\mathbf{\Omega} = \frac{\lambda}{2}\mathbf{I}$ in (6.25). Moreover, all methods but NSC had an additional tuning parameter, the number of discriminant vectors to include in the classifier.

Validation set errors and the numbers of nonzero features used are reported in Table 6.1. In that table, the numbers of discriminant vectors used by all methods except NSC are also reported. Penalized LDA-FL has by far the best performance in all three simulations, since it exploits the fact that the important features have a linear ordering. Of course, in real data applications, penalized LDA-FL can only be applied if such an ordering is present. Note that penalized LDA and SDA tend to use fewer than three components in Simulation 3, in which a one-dimensional projection of the data allows for differentiation between the classes. SDA performs poorly in all simulations, presumably because it uses a full covariance matrix of the features instead of a diagonal estimate.

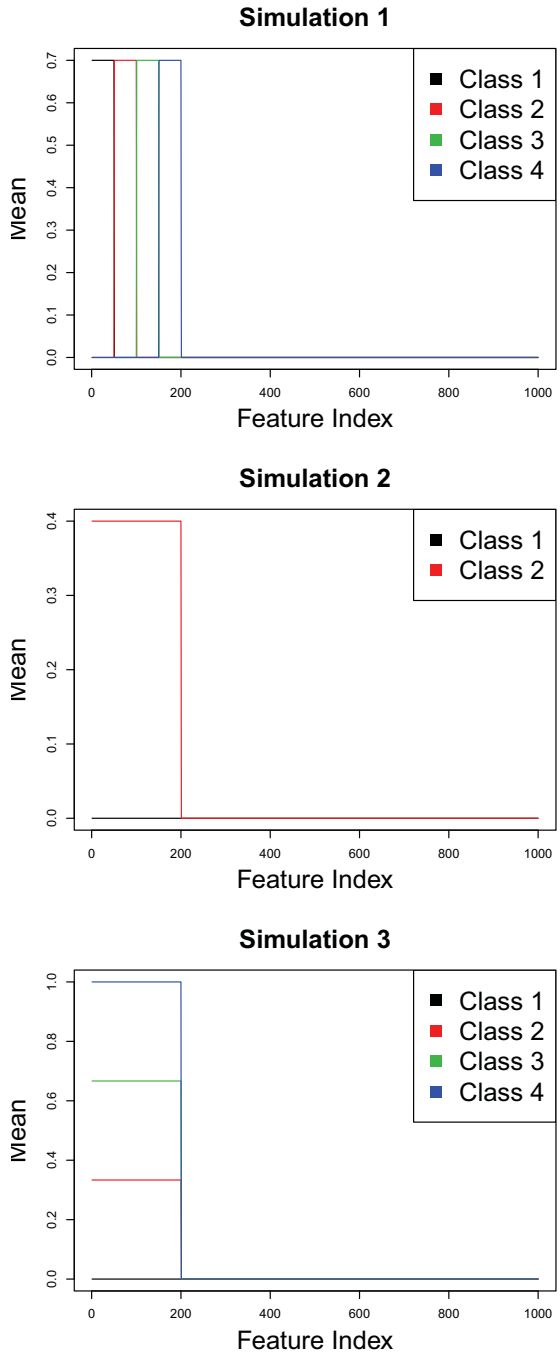

Figure 6.1: Class mean vectors for each simulation.

		Penalized LDA-L_1	Penalized LDA-FL	NSC	SDA
Sim 1	Errors	21.92(0.6)	4.08(1)	20.98(1.2)	69.58(2.4)
	Features	645.34(18.8)	258.94(17.5)	734.02(28.3)	189.5(6.2)
	Components	3(0)	2.98(0)	-	3(0)
Sim 2	Errors	129.44(2)	111.4(2)	131.46(2)	191.72(3.6)
	Features	465.78(23.5)	362.8(18.6)	626.84(35.5)	261.44(20.9)
	Components	1(0)	1(0)	-	1(0)
Sim 3	Errors	71.7(4.6)	18.72(0.7)	192.7(4.3)	377.2(15.4)
	Features	274.28(10.5)	215.2(7.6)	933.36(9.5)	99.62(8.5)
	Components	1.04(0)	1(0)	-	1.22(0.1)

Table 6.1: Results for penalized LDA, NSC, and SDA on Simulations 1, 2, and 3. Mean (and standard errors) of three quantities are shown, computed over 50 repetitions: validation set errors, number of nonzero features, and number of discriminant vectors used.

6.4.2 Application to gene expression data

We compare penalized LDA-L_1, NSC, and SDA on two gene expression data sets:

> **Alon data**. A colon cancer data set consisting of 40 tumor and 22 normal colon tissue samples with measurements on 2000 genes (Alon et al. 1999).
>
> **Khan data**. A small round blue cell tumor data set, consisting of 2308 gene expression measurements for 88 samples of four types of tumors: neuroblastoma, rhabdomyosarcoma, non-Hodgkin lymphoma, and the Ewing family of tumors (Khan et al. 2001).

Each data set was repeatedly split into training, test, and validation sets of equal sizes. For each method, models were fit on the training set for a range of tuning parameter values. Tuning parameter values were then chosen to minimize test set error. Finally, validation set errors were computed. The tuning parameters are as described in Chapter 6.4.1.

Validation set error rates and the number of nonzero coefficients in the final models are reported in Table 6.2. In this example, SDA has the worst performance, presumably because it does not use a diagonal estimate for $\mathbf{\Sigma}_w$.

		NSC	Penalized LDA-L_1	SDA
Alon	Errors	3.44(0.2)	3.56(0.2)	4.2(0.3)
	Features	1026.18(124)	625.46(37.8)	340.88(103.2)
	Components	-	1(0)	1(0)
Khan	Errors	7.42(0.3)	7.88(0.3)	8.18(0.3)
	Features	560.4(101.1)	1504.72(27.2)	74.9(45.6)
	Components	-	2.82(0.1)	2.68(0.1)

Table 6.2: Results obtained on gene expression data over 50 training/test/validation set splits. Quantities reported are the average (and standard error) of validation set errors, nonzero coefficients, and discriminant vectors used.

6.4.3 Application to DNA copy number data

Comparative genomic hybridization (CGH) is a technique for measuring the DNA copy number of a tissue sample at selected locations in the genome (see e.g. Kallioniemi et al. 1992). Each CGH measurement represents the $\log_2$ ratio between the number of copies of a gene in the tissue of interest and the number of copies of that same gene in reference cells; we will assume that these measurements are ordered along the chromosome. In general, there should be two copies of each chromosome in an individual's genome: one per parent. Consequently, CGH data tends to be sparse. Under certain conditions, chromosomal regions spanning multiple genes may be amplified or deleted in a given sample, and so CGH data tends to be piecewise constant. A number of methods have been proposed for identification of regions of copy number gain and loss in a single CGH sample (see e.g. Venkatraman & Olshen 2007, Picard et al. 2005). In particular, the proposal of Tibshirani & Wang (2008) involves using the fused lasso to approximate a CGH sample as a sparse and piecewise constant signal.

In Beck et al. (2010), a number of samples from leiomyosarcoma patients were profiled. Clustering the samples on the basis of gene expression measurements revealed the existence of three previously unknown distinct subgroups of leiomyosarcoma. CGH data were then collected for the samples corresponding to two of these subgroups. It is natural to ask whether one can distinguish between these two subgroups on the basis of the CGH data. Our proposal for penalized LDA-FL can be applied directly to this problem. The fused lasso penalty is appropriate because we expect that chromosomal regions composed of sets of contiguous CGH spots will have different amplification patterns between subgroups. It must be applied with care in order to encourage the discriminant vector to be piecewise constant within each chromosome, but not between chromosomes.

The Beck et al. (2010) data consist of 19 samples and 29910 CGH measurements. The two subgroups contain 12 and 7 samples each. For the sake of comparison, NSC was also

performed. Since the sample size of this data set is quite small, rather than splitting the data into a training set and a test set, we simply performed 5-fold cross-validation on the full data set and report the cross-validation errors. NSC resulted in a minimum of 2/19 cross-validation errors, and penalized LDA-FL resulted in a minimum of 1/19 cross-validation errors. The main advantage of penalized LDA-FL is in the interpretability of the discriminant vector, shown in Figure 6.2. It can be seen from the figure that the penalized LDA-FL classifier makes decisions based on contiguous regions of chromosomal gain or loss. A similar analysis was performed in Beck et al. (2010).

6.5 Maximum likelihood, optimal scoring, and extensions to high dimensions

In this section, we review the maximum likelihood problem and the optimal scoring problem, which lead to the same classification rule as Fisher's discriminant problem (Mardia et al. 1979). We also review past extensions of LDA to the high-dimensional setting.

6.5.1 The maximum likelihood problem

Suppose that the observations are independent and normally distributed with a common within-class covariance matrix $\mathbf{\Sigma}_w \in \mathbb{R}^{p\times p}$ and a class-specific mean vector $\boldsymbol{\mu}_k \in \mathbb{R}^p$. The log likelihood under this model is

$$\sum_{k=1}^{K}\sum_{i\in C_k}\{-\frac{1}{2}\log|\mathbf{\Sigma}_w| - \frac{1}{2}\mathrm{tr}[\mathbf{\Sigma}_w^{-1}(\mathbf{x}_i-\boldsymbol{\mu}_k)(\mathbf{x}_i-\boldsymbol{\mu}_k)^T]\} + c. \tag{6.21}$$

If the classes have equal prior probabilities, then by Bayes' theorem, a new observation $\mathbf{x}$ is classified to the class for which the discriminant function

$$\delta_k(\mathbf{x}) = \mathbf{x}^T\hat{\mathbf{\Sigma}}_w^{-1}\hat{\boldsymbol{\mu}}_k - \frac{1}{2}\hat{\boldsymbol{\mu}}_k^T\hat{\mathbf{\Sigma}}_w^{-1}\hat{\boldsymbol{\mu}}_k \tag{6.22}$$

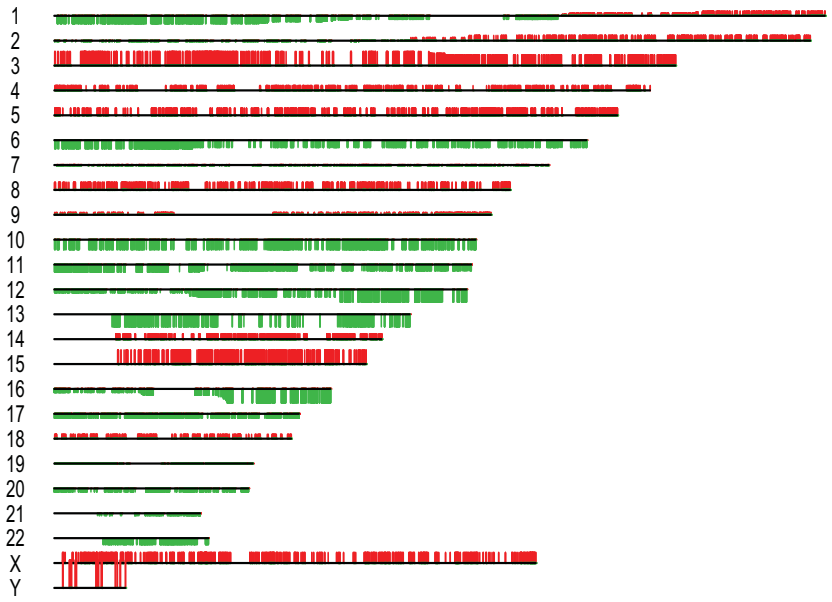

Figure 6.2: For the CGH data example, the discriminant vector obtained using penalized LDA-FL is shown. The discriminant coefficients are shown at the appropriate chromosomal locations. A red line indicates a positive value in the discriminant coefficient at that chromosomal position, and a green line indicates a negative value.

is maximal. One can show that this is the same as the classification rule obtained from Fisher's discriminant problem.

6.5.2 The optimal scoring problem

Let $\mathbf{Y}$ be a $n \times K$ matrix, with $Y_{ik} = 1_{i \in C_k}$. Then, optimal scoring involves sequentially solving

$$\begin{aligned} &\underset{\boldsymbol{\beta}_k \in \mathbb{R}^p, \boldsymbol{\theta}_k \in \mathbb{R}^K}{\text{minimize}} \{\frac{1}{n}||\mathbf{Y}\boldsymbol{\theta}_k - \mathbf{X}\boldsymbol{\beta}_k||^2\} \\ &\text{subject to } \boldsymbol{\theta}_k^T \mathbf{Y}^T \mathbf{Y} \boldsymbol{\theta}_k = 1, \boldsymbol{\theta}_k^T \mathbf{Y}^T \mathbf{Y} \boldsymbol{\theta}_i = 0 \ \ \forall i < k \end{aligned} \tag{6.23}$$

for $k = 1, \ldots, K-1$. The solution $\hat{\boldsymbol{\beta}}_k$ to (6.23) is proportional to the solution to (6.1). Somewhat involved proofs of this fact are given in Breiman & Ihaka (1984) and Hastie et al. (1995). We provide a simpler proof in Chapter 6.7.

6.5.3 LDA in high dimensions

An attractive way to obtain an interpretable classifier in the high-dimensional setting is through a penalization approach. In Chapter 6.3, we proposed penalizing Fisher's discriminant problem. Past proposals have involved penalizing the maximum likelihood and optimal scoring problems.

The *nearest shrunken centroids (NSC)* proposal (Tibshirani et al. 2002, Tibshirani et al. 2003) assigns an observation $\mathbf{x}^*$ to the class that minimizes

$$\sum_{j=1}^{p} \frac{(x_j^* - \bar{\mu}_{kj})^2}{\hat{\sigma}_j^2}, \tag{6.24}$$

where $\bar{\mu}_{kj} = S(\hat{\mu}_{kj}, \lambda \hat{\sigma}_j \sqrt{\frac{1}{n_k} + \frac{1}{n}})$, S is the soft-thresholding operator (1.8), and we have assumed equal prior probabilities for each class. This classification rule approximately follows from applying an L_1 penalty to the mean vectors in the log likelihood (6.21) and

assuming independence of the features (Hastie et al. 2009).

Several authors have proposed penalizing the optimal scoring criterion (6.23) by imposing penalties on $\boldsymbol{\beta}_k$ (see e.g. Grosenick et al. 2008, Leng 2008). For instance, the *sparse discriminant analysis* (SDA) proposal (Clemmensen et al. 2010) involves sequentially solving

$$\begin{aligned} &\underset{\boldsymbol{\beta}_k, \boldsymbol{\theta}_k}{\text{minimize}} \{ \frac{1}{n} ||\mathbf{Y}\boldsymbol{\theta}_k - \mathbf{X}\boldsymbol{\beta}_k||^2 + \boldsymbol{\beta}_k^T \boldsymbol{\Omega} \boldsymbol{\beta}_k + \lambda ||\boldsymbol{\beta}_k||_1 \} \\ &\text{subject to } \boldsymbol{\theta}_k^T \mathbf{Y}^T \mathbf{Y} \boldsymbol{\theta}_k = 1, \boldsymbol{\theta}_k^T \mathbf{Y}^T \mathbf{Y} \boldsymbol{\theta}_i = 0 \ \ \forall i < k. \end{aligned} \tag{6.25}$$

where λ is a nonnegative tuning parameter and $\boldsymbol{\Omega}$ is a penalization matrix. If $\boldsymbol{\Omega} = \gamma \mathbf{I}$ for $\gamma > 0$, then this is an elastic net penalty. The resulting discriminant vectors will be sparse if λ is sufficiently large. If $\lambda = 0$, then this reduces to the *penalized discriminant analysis* proposal of Hastie et al. (1995). The criterion (6.25) can be optimized in a simple iterative fashion. In fact, if any convex penalties are applied to the discriminant vectors in the optimal scoring criterion (6.23), then the resulting problem is easy to solve using an iterative approach. However, the optimal scoring problem is a somewhat unnatural formulation for LDA.

Our penalized LDA proposal is a direct extension of (6.1) that is even simpler to optimize than penalized optimal scoring. Trendafilov & Jolliffe (2007) consider a problem very similar to penalized LDA-L_1, and they provide a suitable algorithm. But they discuss only the $p < n$ case. Their algorithm is more complex than ours, and does not extend to general convex penalty functions.

A summary of proposals that extend LDA to the high-dimensional setting through the use of L_1 penalties is given in Table 6.3. Next, we will explain how our penalized LDA-L_1 proposal relates to the NSC and SDA methods.

Approach	Advantages	Disadvantages	Citation
Max. Lik.	Sparse class means if diagonal estimate of $\mathbf{\Sigma}_w$ is used. Fast computation.	Does not give sparse discriminant vectors.	Tibshirani et al. (2002)
Opt. Scoring	Sparse discriminant vectors.	Difficult to enforce diagonal estimate for $\mathbf{\Sigma}_w$, (useful in $p > n$ setting). Slow computation.	Grosenick et al. (2008) Leng (2008) Clemmensen et al. (2010)
Fisher's Disc.	Sparse discriminant vectors. Simple to enforce diagonal estimate of $\mathbf{\Sigma}_w$. Fast computation if diagonal estimate of $\mathbf{\Sigma}_w$ is used.	Slow computation unless diagonal estimate of $\mathbf{\Sigma}_w$ is used.	This work.

Table 6.3: Summary of approaches for penalizing LDA using L_1.

6.6 Connections with existing methods

6.6.1 Connection with SDA

Consider the SDA criterion (6.25) with $k = 1$. We drop the subscripts on $\boldsymbol{\beta}_1$ and $\boldsymbol{\theta}_1$ for convenience. For any $\boldsymbol{\beta}$ for which $\mathbf{Y}^T\mathbf{X}\boldsymbol{\beta} = 0$, the optimal $\boldsymbol{\theta}$ equals $\frac{(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{X}\boldsymbol{\beta}}{\sqrt{\boldsymbol{\beta}^T\mathbf{X}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{X}\boldsymbol{\beta}}}$. So (6.25) can be rewritten as

$$\underset{\boldsymbol{\beta}}{\text{maximize}}\{\frac{2}{\sqrt{n}}\sqrt{\boldsymbol{\beta}^T\hat{\boldsymbol{\Sigma}}_b\boldsymbol{\beta}} - \boldsymbol{\beta}^T(\hat{\boldsymbol{\Sigma}}_b + \hat{\boldsymbol{\Sigma}}_w + \boldsymbol{\Omega})\boldsymbol{\beta} - \lambda||\boldsymbol{\beta}||_1\}. \tag{6.26}$$

Assume that each feature has been standardized to have within-class standard deviation equal to 1. Take $\tilde{\boldsymbol{\Sigma}}_w = \hat{\boldsymbol{\Sigma}}_w + \boldsymbol{\Omega}$, where $\boldsymbol{\Omega}$ is chosen so that $\tilde{\boldsymbol{\Sigma}}_w$ is positive definite. Then, the following proposition holds.

Proposition 6.6.1. *Consider the penalized LDA-L_1 problem (6.12) where $\lambda_1 > 0$ and $k = 1$. Suppose that at the solution $\boldsymbol{\beta}^*$ to (6.12), the objective is positive. Then, there exists a positive tuning parameter λ_2 and a positive c such that $c\boldsymbol{\beta}^*$ corresponds to a zero of the generalized gradient of the SDA objective (6.26) with $k = 1$.*

A proof is given in Chapter 6.7. Proposition 6.6.1 states that if the same positive definite estimate for $\boldsymbol{\Sigma}_w$ is used for both problems, then the solution of the penalized LDA-L_1 problem corresponds to a point where the generalized gradient of the SDA problem is zero. But since the SDA problem is not convex, this does not imply that there is a correspondence between the solutions of the two problems. Penalized LDA-L_1 has some advantages over SDA. Unlike SDA, penalized LDA-L_1 has a clear relationship with Fisher's discriminant problem. Moreover, unlike SDA, it provides a natural way to enforce a diagonal estimate of $\boldsymbol{\Sigma}_w$.

6.6.2 Connection with NSC

Consider the NSC classification rule when there are $K = 2$ classes with equal class sizes $n_1 = n_2 = \frac{n}{2}$. This classification rule is the same as the one obtained from the problem

$$\underset{\boldsymbol{\beta}}{\text{maximize}}\{\sqrt{\boldsymbol{\beta}^T\hat{\boldsymbol{\Sigma}}_b\boldsymbol{\beta}} - \lambda\sum_{j=1}^{p}|\beta_j\hat{\sigma}_j|\} \text{ subject to } \boldsymbol{\beta}^T\tilde{\boldsymbol{\Sigma}}_w\boldsymbol{\beta} \leq 1 \tag{6.27}$$

where $\tilde{\boldsymbol{\Sigma}}_w$ is the diagonal estimate (6.3). (6.27) is simply a modified version of the penalized LDA-L_1 criterion, in which the between-class variance term has been replaced with its square root.

In this case, NSC assigns an observation $\mathbf{x} \in \mathbb{R}^p$ to the class that maximizes

$$\sum_{j=1}^{p}\frac{x_j S(\overline{\mathbf{X}}_{kj}, \hat{\sigma}_j\lambda)}{\hat{\sigma}_j^2} \tag{6.28}$$

where $\overline{\mathbf{X}}_{kj}$ is the mean of feature j in class k, and the soft-thresholding operator S is given by (1.8). On the other hand, (6.27) assigns $\mathbf{x}$ to the class that minimizes

$$\left|\sum_{j=1}^{p}\frac{\overline{\mathbf{X}}_{kj}S(\overline{\mathbf{X}}_{1j}, \hat{\sigma}_j\lambda)}{\hat{\sigma}_j^2} - \sum_{j=1}^{p}\frac{x_j S(\overline{\mathbf{X}}_{1j}, \hat{\sigma}_j\lambda)}{\hat{\sigma}_j^2}\right|. \tag{6.29}$$

This follows from the fact that (6.27) reduces to

$$\underset{\boldsymbol{\beta}}{\text{maximize}}\{\boldsymbol{\beta}^T\overline{\mathbf{X}}_1 - \lambda\sum_{j=1}^{p}|\beta_j\hat{\sigma}_j|\} \text{ subject to } \sum_{j=1}^{p}\beta_j^2\hat{\sigma}_j^2 \leq 1, \tag{6.30}$$

since $\frac{1}{\sqrt{n}}\mathbf{X}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}} = \overline{\mathbf{X}}_1[\frac{1}{\sqrt{2}} \quad -\frac{1}{\sqrt{2}}]$ and $\hat{\boldsymbol{\Sigma}}_b = \frac{1}{n}\mathbf{X}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{X}$.

Since the first term in (6.29) is positive if $k = 1$ and negative if $k = 2$, (6.27) classifies to class 1 if $\sum_{j=1}^{p}\frac{x_j S(\overline{\mathbf{X}}_{1j},\hat{\sigma}_j\lambda)}{\hat{\sigma}_j^2} > 0$ and classifies to class 2 if $\sum_{j=1}^{p}\frac{x_j S(\overline{\mathbf{X}}_{1j},\hat{\sigma}_j\lambda)}{\hat{\sigma}_j^2} < 0$. Because $\overline{\mathbf{X}}_{1j} = -\overline{\mathbf{X}}_{2j}$, by inspection of (6.28), the two methods result in the same classification rule.

6.7 Proofs

6.7.1 Proof of equivalence of Fisher's LDA and optimal scoring

Proof. Consider the following two problems:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\text{maximize}}\{\boldsymbol{\beta}^T\hat{\boldsymbol{\Sigma}}_b\boldsymbol{\beta}\} \text{ subject to } \boldsymbol{\beta}^T(\hat{\boldsymbol{\Sigma}}_w+\boldsymbol{\Omega})\boldsymbol{\beta}=1 \tag{6.31}$$

and

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p,\boldsymbol{\theta}\in\mathbb{R}^K}{\text{minimize}}\{\frac{1}{n}||\mathbf{Y}\boldsymbol{\theta}-\mathbf{X}\boldsymbol{\beta}||^2+\boldsymbol{\beta}^T\boldsymbol{\Omega}\boldsymbol{\beta}\} \text{ subject to } \boldsymbol{\theta}^T\mathbf{Y}^T\mathbf{Y}\boldsymbol{\theta}=1. \tag{6.32}$$

In Hastie et al. (1995), a somewhat challenging proof is given of the fact that the solutions $\hat{\boldsymbol{\beta}}$ to the two problems are proportional to each other. Here, we present a more direct argument. In (6.31) and (6.32), $\boldsymbol{\Omega}$ is a matrix such that $\hat{\boldsymbol{\Sigma}}_w+\boldsymbol{\Omega}$ is positive definite; if $\boldsymbol{\Omega}=0$ then these two problems reduce to Fisher's LDA and optimal scoring. Optimizing (6.32) with respect to $\boldsymbol{\theta}$, we see that the $\boldsymbol{\beta}$ that solves (6.32) also solves

$$\underset{\boldsymbol{\beta}}{\text{minimize}}\{-\frac{2}{\sqrt{n}}\sqrt{\boldsymbol{\beta}^T\hat{\boldsymbol{\Sigma}}_b\boldsymbol{\beta}}+\boldsymbol{\beta}^T\hat{\boldsymbol{\Sigma}}_b\boldsymbol{\beta}+\boldsymbol{\beta}^T(\hat{\boldsymbol{\Sigma}}_w+\boldsymbol{\Omega})\boldsymbol{\beta}\}. \tag{6.33}$$

For notational convenience, let $\tilde{\boldsymbol{\beta}}=(\hat{\boldsymbol{\Sigma}}_w+\boldsymbol{\Omega})^{\frac{1}{2}}\boldsymbol{\beta}$ and $\tilde{\boldsymbol{\Sigma}}_b=(\hat{\boldsymbol{\Sigma}}_w+\boldsymbol{\Omega})^{-\frac{1}{2}}\hat{\boldsymbol{\Sigma}}_b(\hat{\boldsymbol{\Sigma}}_w+\boldsymbol{\Omega})^{-\frac{1}{2}}$. Then, the problems become

$$\underset{\tilde{\boldsymbol{\beta}}}{\text{maximize}}\{\tilde{\boldsymbol{\beta}}^T\tilde{\boldsymbol{\Sigma}}_b\tilde{\boldsymbol{\beta}}\} \text{ subject to } \tilde{\boldsymbol{\beta}}^T\tilde{\boldsymbol{\beta}}=1 \tag{6.34}$$

and

$$\underset{\tilde{\boldsymbol{\beta}}}{\text{minimize}}\{-\frac{2}{\sqrt{n}}\sqrt{\tilde{\boldsymbol{\beta}}^T\tilde{\boldsymbol{\Sigma}}_b\tilde{\boldsymbol{\beta}}}+\tilde{\boldsymbol{\beta}}^T(\tilde{\boldsymbol{\Sigma}}_b+\mathbf{I})\tilde{\boldsymbol{\beta}}\}. \tag{6.35}$$

It is easy to see that the solution to (6.34) is the first eigenvector of $\tilde{\mathbf{\Sigma}}_b$. Let $\hat{\boldsymbol{\beta}}$ denote the solution to (6.35). Consequently, $\hat{\boldsymbol{\beta}}^T \tilde{\mathbf{\Sigma}}_b \hat{\boldsymbol{\beta}} > 0$. So $\hat{\boldsymbol{\beta}}$ satisfies

$$\tilde{\mathbf{\Sigma}}_b \hat{\boldsymbol{\beta}}(1 - \frac{1}{\sqrt{n\hat{\boldsymbol{\beta}}^T \tilde{\mathbf{\Sigma}}_b \hat{\boldsymbol{\beta}}}}) + \hat{\boldsymbol{\beta}} = 0, \tag{6.36}$$

and therefore $\sqrt{n\hat{\boldsymbol{\beta}}^T \tilde{\mathbf{\Sigma}}_b \hat{\boldsymbol{\beta}}} < 1$. Now (6.36) indicates that $\hat{\boldsymbol{\beta}}$ is an eigenvector of $\tilde{\mathbf{\Sigma}}_b$ with eigenvalue $\lambda = \frac{\sqrt{n\hat{\boldsymbol{\beta}}^T \tilde{\mathbf{\Sigma}}_b \hat{\boldsymbol{\beta}}}}{1-\sqrt{n\hat{\boldsymbol{\beta}}^T \tilde{\mathbf{\Sigma}}_b \hat{\boldsymbol{\beta}}}}$; however, it remains to determine which eigenvector. Notice that if we let $w = \hat{\boldsymbol{\beta}}^T \hat{\boldsymbol{\beta}}$ then $\lambda = \frac{\sqrt{n\lambda w}}{1-\sqrt{n\lambda w}}$, and so $w = \frac{\lambda}{n(1+\lambda)^2}$. Then the objective of (6.35) evaluated at $\hat{\boldsymbol{\beta}}$ equals

$$-\frac{2}{\sqrt{n}}\sqrt{\lambda w} + \lambda w + w = \frac{-2\lambda}{n(1+\lambda)} + \frac{\lambda}{n(1+\lambda)} = -\frac{\lambda}{n(1+\lambda)}. \tag{6.37}$$

The minimum occurs when λ is large. So the solution to (6.35) is the largest eigenvector of $\tilde{\mathbf{\Sigma}}_b$. □

This argument can be extended to show that subsequent solutions to Fisher's discriminant problem and the optimal scoring problem are proportional to each other.

6.7.2 Proof of Proposition 6.3.1

Proof. Letting $\tilde{\boldsymbol{\beta}}_k = \tilde{\mathbf{\Sigma}}_w^{\frac{1}{2}} \boldsymbol{\beta}_k$, (6.6) becomes

$$\underset{\tilde{\boldsymbol{\beta}}_k}{\text{maximize}} \{\tilde{\boldsymbol{\beta}}_k^T \tilde{\mathbf{\Sigma}}_w^{-\frac{1}{2}} \mathbf{X}^T \mathbf{Y} (\mathbf{Y}^T \mathbf{Y})^{-\frac{1}{2}} \mathbf{P}_k^{\perp} (\mathbf{Y}^T \mathbf{Y})^{-\frac{1}{2}} \mathbf{Y}^T \mathbf{X} \tilde{\mathbf{\Sigma}}_w^{-\frac{1}{2}} \tilde{\boldsymbol{\beta}}_k\} \text{ subject to } ||\tilde{\boldsymbol{\beta}}_k||^2 \leq 1, \tag{6.38}$$

which is equivalent to

$$\underset{\tilde{\boldsymbol{\beta}}_k, \mathbf{u}_k}{\text{maximize}} \{\tilde{\boldsymbol{\beta}}_k^T \mathbf{A} \mathbf{P}_k^{\perp} \mathbf{u}_k\} \text{ subject to } ||\tilde{\boldsymbol{\beta}}_k||^2 \leq 1, ||\mathbf{u}_k||^2 \leq 1, \tag{6.39}$$

where $\mathbf{A} = \tilde{\mathbf{\Sigma}}_w^{-\frac{1}{2}} \mathbf{X}^T \mathbf{Y} (\mathbf{Y}^T \mathbf{Y})^{-\frac{1}{2}}$. Equivalence of (6.39) and (6.38) can be seen from partially optimizing (6.39) with respect to $\mathbf{u}_k$.

We claim that $\tilde{\boldsymbol{\beta}}_k$ and $\mathbf{u}_k$ that solve (6.39) are the kth left and right singular vectors of $\mathbf{A}$. By inspection, the claim holds when $k = 1$. Now, suppose that the claim holds for all $i < k$, where $k > 1$. Then, partially optimizing (6.39) with respect to $\boldsymbol{\beta}_k$ yields

$$\underset{\mathbf{u}_k}{\text{maximize}} \{\mathbf{u}_k^T \mathbf{P}_k^{\perp} \mathbf{A}^T \mathbf{A} \mathbf{P}_k^{\perp} \mathbf{u}_k\} \text{ subject to } ||\mathbf{u}_k||^2 \leq 1. \tag{6.40}$$

From the definition of $\mathbf{P}_k^{\perp}$ and the fact that $\boldsymbol{\beta}_i$ and $\mathbf{u}_i$ are the ith singular vectors of $\mathbf{A}$ for all $i < k$, it follows that $\mathbf{P}_k^{\perp} = \mathbf{I} - \sum_{i=1}^{k-1} \mathbf{u}_i \mathbf{u}_i^T$. Therefore, $\mathbf{u}_k$ is the kth right singular vector of $\mathbf{A}$. So $\tilde{\boldsymbol{\beta}}_k$ is the kth left singular vector of $\mathbf{A}$, or equivalently the kth eigenvector of $\tilde{\mathbf{\Sigma}}_w^{-\frac{1}{2}} \hat{\mathbf{\Sigma}}_b \tilde{\mathbf{\Sigma}}_w^{-\frac{1}{2}}$. Therefore, $\boldsymbol{\beta}_k$ that solves (6.6) is the kth unpenalized discriminant vector.

□

6.7.3 Proof of Proposition 6.6.1

Proof. Consider (6.12) with tuning parameter λ_1 and $k = 1$. Then by Theorem 6.1.1 of Clarke (1990), if there is a nonzero solution $\boldsymbol{\beta}^*$, then there exists $\mu \geq 0$ such that

$$0 \in 2\hat{\mathbf{\Sigma}}_b \boldsymbol{\beta}^* - \lambda_1 \Gamma(\boldsymbol{\beta}^*) - 2\mu \tilde{\mathbf{\Sigma}}_w \boldsymbol{\beta}^*, \tag{6.41}$$

where $\Gamma(\boldsymbol{\beta})$ is the subdifferential of $||\boldsymbol{\beta}||_1$. The subdifferential is the set of subderivatives of $||\boldsymbol{\beta}||_1$; the jth element of a subderivative equals $\text{sign}(\beta_j)$ if $\beta_j \neq 0$ and is between -1 and 1 if $\beta_j = 0$. Left-multiplying (6.41) by $\boldsymbol{\beta}^*$ yields $0 = 2\boldsymbol{\beta}^{*T} \hat{\mathbf{\Sigma}}_b \boldsymbol{\beta}^* - \lambda_1 ||\boldsymbol{\beta}^*||_1 - 2\mu \boldsymbol{\beta}^{*T} \tilde{\mathbf{\Sigma}}_w \boldsymbol{\beta}^*$. Since the sum of the first two terms is positive (since $\boldsymbol{\beta}^*$ is a nonzero solution), it follows that $\mu > 0$.

Now, define a new vector that is proportional to $\boldsymbol{\beta}^*$:

$$\hat{\boldsymbol{\beta}} = \frac{\mu}{(1+\mu)a}\boldsymbol{\beta}^* = c\boldsymbol{\beta}^* \tag{6.42}$$

where $a = \sqrt{n\boldsymbol{\beta}^{*T}\hat{\boldsymbol{\Sigma}}_b\boldsymbol{\beta}^*}$. By inspection, $a \neq 0$, since otherwise $\boldsymbol{\beta}^*$ would not be a nonzero solution. Also, let $\lambda_2 = \lambda_1(\frac{1-ca}{a})$. Note that $1 - ca = \frac{1}{1+\mu} > 0$, so $\lambda_2 > 0$.

The generalized gradient of (6.26) with tuning parameter λ_2 evaluated at $\hat{\boldsymbol{\beta}}$ is proportional to

$$2\hat{\boldsymbol{\Sigma}}_b\hat{\boldsymbol{\beta}} - \lambda_2\Gamma(\hat{\boldsymbol{\beta}})(\frac{\sqrt{n\hat{\boldsymbol{\beta}}^T\hat{\boldsymbol{\Sigma}}_b\hat{\boldsymbol{\beta}}}}{1-\sqrt{n\hat{\boldsymbol{\beta}}^T\hat{\boldsymbol{\Sigma}}_b\hat{\boldsymbol{\beta}}}}) - 2\tilde{\boldsymbol{\Sigma}}_w\hat{\boldsymbol{\beta}}(\frac{\sqrt{n\hat{\boldsymbol{\beta}}^T\hat{\boldsymbol{\Sigma}}_b\hat{\boldsymbol{\beta}}}}{1-\sqrt{n\hat{\boldsymbol{\beta}}^T\hat{\boldsymbol{\Sigma}}_b\hat{\boldsymbol{\beta}}}}), \tag{6.43}$$

or equivalently,

$$\begin{aligned} 2c\hat{\boldsymbol{\Sigma}}_b\boldsymbol{\beta}^* - \lambda_2\Gamma(\boldsymbol{\beta}^*)\frac{ac}{1-ac} - 2c\tilde{\boldsymbol{\Sigma}}_w\boldsymbol{\beta}^*\frac{ac}{1-ac} &= 2c\hat{\boldsymbol{\Sigma}}_b\boldsymbol{\beta}^* - \lambda_1 c\Gamma(\boldsymbol{\beta}^*) - 2c\tilde{\boldsymbol{\Sigma}}_w\boldsymbol{\beta}^*\frac{ac}{1-ac} \\ &= 2c\hat{\boldsymbol{\Sigma}}_b\boldsymbol{\beta}^* - \lambda_1 c\Gamma(\boldsymbol{\beta}^*) - 2c\mu\tilde{\boldsymbol{\Sigma}}_w\boldsymbol{\beta}^* \\ &= c(2\hat{\boldsymbol{\Sigma}}_b\boldsymbol{\beta}^* - \lambda_1\Gamma(\boldsymbol{\beta}^*) - 2\mu\tilde{\boldsymbol{\Sigma}}_w\boldsymbol{\beta}^*). \end{aligned} \tag{6.44}$$

Comparing (6.41) to (6.44), we see that 0 is contained in the generalized gradient of the SDA objective evaluated at $\hat{\boldsymbol{\beta}}$.

□

Chapter 7

Discussion

In recent years, massive data sets have become increasingly common across a number of fields. Consequently, there is a growing need for computationally efficient statistical methods that are appropriate for the high-dimensional setting in which the number of features exceeds the number of observations.

In this dissertation, we have proposed a penalized matrix decomposition, an extension of the singular value decomposition that yields interpretable discriminant vectors. We have used this decomposition in order to develop a number of statistical tools for the supervised and unsupervised analysis of high-dimensional data. We have attempted to explain how our proposals fit into the existing statistical literature, and have sought to unify past proposals when possible.

Though many proposals for the analysis of high-dimensional data have been made in the literature, much remains to be done. In particular, as the cost of collecting very large data sets continues to decrease across a variety of fields, we expect that there will be an increased need for statistical tools geared at hypothesis generation rather than hypothesis testing. When hypothesis generation is the goal, one may wish to apply unsupervised methods such as matrix decompositions and clustering in order to discover previously unknown signal in the data. Unsupervised learning in the high-dimensional setting remains a

relatively unexplored research area. It is often difficult to assess the results obtained using unsupervised methods, since unlike in the supervised setting there is no "gold standard". For each of the unsupervised methods proposed in this work, we have suggested validation methods. But improved methods for evaluating unsupervised methods are needed.

In this dissertation, we have attempted to develop statistical tools to solve real problems that domain scientists face in the analysis of their data. As scientific fields change, novel statistical methods will continue to be needed. Therefore, we expect that high-dimensional data analysis will remain an important statistical research area in the coming years.

Bibliography

Alizadeh, A., Eisen, M., Davis, R. E., Ma, C., Lossos, I., Rosenwald, A., Boldrick, J., Sabet, H., Tran, T., Yu, X., Powell, J., Marti, G., Moore, T., Hudson, J., Lu, L., Lewis, D., Tibshirani, R., Sherlock, G., Chan, W., Greiner, T., Weisenburger, D., Armitage, K., Warnke, R., Levy, R., Wilson, W., Grever, M., Byrd, J., Botstein, D., Brown, P. & Staudt, L. (2000), 'Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling', *Nature* **403**, 503–511.

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D. & Levine, A. (1999), 'Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays', *Proc. Nat. Acad. Sciences* **96**, 6745–6750.

Bair, E., Hastie, T., Paul, D. & Tibshirani, R. (2006), 'Prediction by supervised principal components', *J. Amer. Statist. Assoc.* **101**, 119–137.

Bair, E. & Tibshirani, R. (2004), 'Semi-supervised methods to predict patient survival from gene expression data', *PLOS Biology* **2**, 511–522.

Beck, A., Lee, C., Witten, D., Gleason, B., Edris, B., Espinosa, I., Zhu, S., Li, R., Montgomery, K., Marinelli, R., Tibshirani, R., Hastie, T., Jablons, D., Rubin, B., Fletcher, C., West, R. & van de Rijn, M. (2010), 'Discovery of molecular subtypes in leiomyosarcoma through integrative molecular profiling', *Oncogene* **29**, 845–854.

Bickel, P. & Levina, E. (2004), 'Some theory for Fisher's linear discriminant function, 'naive Bayes', and some alternatives when there are many more variables than observations', *Bernoulli* **10(6)**, 989–1010.

Borg, I. & Groenen, P. (2005), *Modern multidimensional scaling*, Springer, New York.

Boyd, S. & Vandenberghe, L. (2004), *Convex Optimization*, Cambridge University Press.

Breiman, L. & Ihaka, R. (1984), Nonlinear discriminant analysis via scaling and ACE, Technical report, Univ. California, Berkeley.

Chang, W.-C. (1983), 'On using principal components before separating a mixture of two multivariate normal distributions', *Journal of the Royal Statistical Society, Series C (Applied Statistics)* **32**, 267–275.

Chin, K., DeVries, S., Fridlyand, J., Spellman, P., Roydasgupta, R., Kuo, W.-L., Lapuk, A., Neve, R., Qian, Z., Ryder, T., Chen, F., Feiler, H., Tokuyasu, T., Kingsley, C., Dairkee, S., Meng, Z., Chew, K., Pinkel, D., Jain, A., Ljung, B., Esserman, L., Albertson, D., Waldman, F. & Gray, J. (2006), 'Genomic and transcriptional aberrations linked to breast cancer pathophysiologies', *Cancer Cell* **10**, 529–541.

Chipman, H. & Tibshirani, R. (2005), 'Hybrid hierarchical clustering with applications to microarray data', *Biostatistics* **7**, 286–301.

Clarke, F. (1990), *Optimization and nonsmooth analysis*, SIAM, Troy, New York.

Clemmensen, L., Hastie, T. & Ersboll, B. (2010), 'Sparse discriminant analysis', *To appear in Technometrics* .

Dempster, A., Laird, N. & Rubin, D. (1977), 'Maximum likelihood from incomplete data via the EM algorithm (with discussion)', *J. R. Statist. Soc. B.* **39**, 1–38.

Dudoit, S., Fridlyand, J. & Speed, T. (2001), 'Comparison of discrimination methods for the classification of tumors using gene expression data', *J. Amer. Statist. Assoc.* **96**, 1151–1160.

Eckart, C. & Young, G. (1936), 'The approximation of one matrix by another of low rank', *Psychometrika* **1**, 211.

Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2004), 'Least angle regression', *Annals of Statistics* **32**(2), 407–499.

Eisen, M., Spellman, P., Brown, P. & Botstein, D. (1998), 'Cluster analysis and display of genome-wide expression patterns', *Proc. Natl. Acad. Sci., USA.* **95**, 14863–14868.

Fan, J. & Li, R. (2001), 'Variable selection via nonconcave penalized likelihood and its oracle properties', *J. Amer. Statist. Assoc.* **96**, 1348–1360.

Fraley, C. & Raftery, A. (2002), 'Model-based clustering, discriminant analysis, and density estimation', *J. Amer. Statist. Assoc.* **97**, 611–631.

Friedman, H., Hastie, T. & Tibshirani, R. (2010), 'Regularization paths for generalized linear models via coordinate descent', *Journal of Statistical Software* **33**.

Friedman, J. (1989), 'Regularized discriminant analysis', *Journal of the American Statistical Association* **84**, 165–175.

Friedman, J., Hastie, T., Hoefling, H. & Tibshirani, R. (2007), 'Pathwise coordinate optimization', *Annals of Applied Statistics* **1**, 302–332.

Friedman, J. & Meulman, J. (2004), 'Clustering objects on subsets of attributes', *J. Roy. Stat. Soc., Ser. B* **66**, 815–849.

Ghosh, D. & Chinnaiyan, A. M. (2002), 'Mixture modelling of gene expression data from microarray experiments', *Bioinformatics* **18**, 275–286.

Gifi, A. (1990), *Nonlinear multivariate analysis*, Wiley, Chichester, England.

Gorski, J., Pfeuffer, F. & Klamroth, K. (2007), 'Biconvex sets and optimization with biconvex functions: a survey and extensions', *Mathematical Methods of Operations Rsearch* **66**, 373–407.

Grosenick, L., Greer, S. & Knutson, B. (2008), 'Interpretable classifiers for fMRI improve prediction of purchases', *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **16(6)**, 539–547.

Guo, Y., Hastie, T. & Tibshirani, R. (2007), 'Regularized linear discriminant analysis and its application in microarrays', *Biostatistics* **8**, 86–100.

Hastie, T., Buja, A. & Tibshirani, R. (1995), 'Penalized discriminant analysis', *Annals of Statistics* **23**, 73–102.

Hastie, T., Tibshirani, R. & Friedman, J. (2009), *The Elements of Statistical Learning; Data Mining, Inference and Prediction*, Springer Verlag, New York.

Hoefling, H. (2009*a*), 'A path algorithm for the fused lasso signal approximator', *arXiv:0910.0526* .

Hoefling, H. (2009*b*), Topics in statistical learning, PhD thesis, Dept. of Statistics, Stanford University.

Hoerl, A. E. & Kennard, R. (1970), 'Ridge regression: Biased estimation for nonorthogonal problems', *Technometrics* **12**, 55–67.

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

Hoyer, P. (2002), 'Non-negative sparse coding', *Proc. IEEE Workshop on Neural Networks for Signal Processing* .

Hoyer, P. (2004), 'Non-negative matrix factorization with sparseness constraints', *Journal of Machine Learning Research* **5**, 1457–1469.

Hunter, D. & Lange, K. (2004), 'A tutorial on MM algorithms', *The American Statistician* **58**, 30–37.

Hyman, E., Kauraniemi, P., Hautaniemi, S., Wolf, M., Mousses, S., Rozenblum, E., Ringner, M., Sauter, G., Monni, O., Elkahloun, A., Kallioniemi, O.-P. & Kallioniemi, A. (2002), 'Impact of DNA amplification on gene expression patterns in breast cancer', *Cancer Research* **62**, 6240–6245.

International HapMap Consortium (2005), 'A haplotype map of the human genome', *Nature* **437**, 1299–1320.

International HapMap Consortium (2007), 'A second generation human haplotype map of over 3.1 million SNPs', *Nature* **449**, 851–861.

Jolliffe, I., Trendafilov, N. & Uddin, M. (2003), 'A modified principal component technique based on the lasso', *Journal of Computational and Graphical Statistics* **12**, 531–547.

Kallioniemi, A., Kallioniemi, O. P., Sudar, D., Rutovitz, D., Gray, J. W., Waldman, F. & Pinkel, D. (1992), 'Comparative genomic hybridization for molecular cytogenetic analysis of solid tumors', *Science* **258**, 818–821.

Kaufman, L. & Rousseeuw, P. (1990), *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley, New York.

Khan, J., Wei, J., Ringner, M., Saal, L., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C., Peterson, C., & Meltzer, P. (2001), 'Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks', *Nature Medicine* **7**, 673–679.

Krzanowski, W., Jonathan, P., McCarthy, W. & Thomas, M. (1995), 'Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data', *Journal of the Royal Statistical Society, Series C* **44**, 101–115.

Lange, K. (2004), *Optimization*, Springer, New York.

Lange, K., Hunter, D. & Yang, I. (2000), 'Optimization transfer using surrogate objective functions', *Journal of Computational and Graphical Statistics* **9**, 1–20.

Lazzeroni, L. & Owen, A. (2002), 'Plaid models for gene expression data', *Statistica Sinica* **12**, 61–86.

Le Cao, K., Pascal, M., Robert-Granie, C. & Philippe, B. (2009), 'Sparse canonical methods for biological data integration: application to a cross-platform study', *BMC Bioinformatics* **10**.

Le Cao, K., Rossouw, D., Robert-Granie, C. & Besse, P. (2008), 'A sparse PLS for variable selection when integrating Omics data', *Statistical applications in genetics and molecular biology* **7**.

Lee, D. D. & Seung, H. S. (1999), 'Learning the parts of objects by non-negative matrix factorization', *Nature* **401**, 788.

Lee, D. D. & Seung, H. S. (2001), Algorithms for non-negative matrix factorization, *in* 'Advances in Neural Information Processing Systems, (NIPS 2001)'.

Leng, C. (2008), 'Sparse optimal scoring for multiclass cancer diagnosis and biomarker detection using microarray data', *Computational Biology and Chemistry* **32**, 417–425.

Lenz, G., Wright, G., Emre, N., Kohlhammer, H., Dave, S., Davis, R., Carty, S., Lam, L., Shaffer, A., Xiao, W., Powell, J., Rosenwald, A., Ott, G., Muller-Hermelink, H., Gascoyne, R., Connors, J., Campo, E., Jaffe, E., Delabie, J., Smeland, E., Rimsza, L.,

Fisher, R., Weisenburger, D., Chano, W. & Staudt, L. (2008), 'Molecular subtypes of diffuse large B-cell lymphoma arise by distinct genetic pathways', *Proc. Natl. Acad. Sci.* **105**, 13520–13525.

Liu, J. S., Zhang, J. L., Palumbo, M. J. & Lawrence, C. E. (2003), 'Bayesian clustering with variable and transformation selections', *Bayesian statistics* **7**, 249–275.

Lv, J. & Fan, Y. (2009), 'A unified approach to model selection and sparse recovery using regularized least squares', *Annals of Statistics* **37**, 3498–3528.

MacQueen, J. (1967), Some methods for classification and analysis of multivariate observations, *in* 'Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, eds. L.M. LeCam and J. Neyman', Univ. of California Press, pp. 281–297.

Mardia, K., Kent, J. & Bibby, J. (1979), *Multivariate Analysis*, Academic Press.

Massy, W. (1965), 'Principal components regression in exploratory statistical research', *Journal of the American Statistical Association* **60**, 234–236.

Maugis, C., Celeux, G. & Martin-Magniette, M.-L. (2009), 'Variable selection for clustering with Gaussian mixture models', *Biometrics* **65**, 701–709.

McLachlan, G. J., Bean, R. W. & Peel, D. (2002), 'A mixture model-based approach to the clustering of microarray expression data', *Bioinformatics* **18**, 413–422.

McLachlan, G. J. & Peel, D. (2000), *Finite Mixture Models*, John Wiley & Sons, New York, NY.

McLachlan, G. J., Peel, D. & Bean, R. W. (2003), 'Modelling high-dimensional data by mixtures of factor analyzers', *Computational Statisitics and Data Analysis* **41**, 379–388.

Milligan, G. W. & Cooper, M. C. (1985), 'An examination of procedures for determining the number of clusters in a data set', *Psychometrika* **50**, 159–179.

Morley, M., Molony, C., Weber, T., Devlin, J., Ewens, K., Spielman, R. & Cheung, V. (2004), 'Genetic analysis of genome-wide variation in human gene expression', *Nature* **430**, 743–747.

Nowak, G. (2009), Some methods for analyzing high-dimensional genomic data, PhD thesis, Dept. of Statistics, Stanford University.

Nowak, G. & Tibshirani, R. (2008), 'Complementary hierarchical clustering', *Biostatistics* **9(3)**, 467–483.

Owen, A. B. & Perry, P. O. (2009), 'Bi-cross-validation of the SVD and the non-negative matrix factorization', *Annals of Applied Statistics* **3(2)**, 564–594.

Pan, W. & Shen, X. (2007), 'Penalized model-based clustering with application to variable selection', *Journal of Machine Learning Research* **8**, 1145–1164.

Park, M. Y. & Hastie, T. (2007), 'An L_1 regularization path algorithm for generalized linear models', *Journal of the Royal Statistical Society Series B* **69**(4), 659–677.

Parkhomenko, E., Tritchler, D. & Beyene, J. (2009), 'Sparse canonical correlation analysis with application to genomic data integration', *Statistical Applications in Genetics and Molecular Biology* **8**, 1–34.

Perou, C. M., Sorlie, T., Eisen, M. B., Rijn, M. V. D., Jeffrey, S. S., Rees, C. A., Pollack, J. R., Ross, D. T., Johnsen, H., Akslen, L. A., Fluge, O., Pergamenschikov, A., Williams, C., Zhu, S. X., Lonning, P. E., Borresen-dale, A., Brown, P. O. & Botstein, D. (2000), 'Molecular portraits of human breast tumours', *Nature* **406**, 747–752.

Picard, F., Robin, S., Lavielle, M., Vaisse, C. & Daudin, J. (2005), 'A statistical approach for array CGH data analysis', *BMC Bioinformatics* **6**, 6–27.

Pollack, J., Sorlie, T., Perou, C., Rees, C., Jeffrey, S., Lonning, P., Tibshirani, R., Botstein, D., Borresen-Dale, A. & Brown, P. (2002), 'Microarray analysis reveals a major direct role of DNA copy number alteration in the transcriptional program of human breast tumors', *Proceedings of the National Academy of Sciences* **99**, 12963–12968.

Price, A. L., Patterson, N. J., Weinblatt, M. E., Shadick, N. A. & Reich, D. (2006), 'Principal components analysis corrects for stratification in genome-wide association studies', *Nature Genetics* **38**, 904–909.

Raftery, A. & Dean, N. (2006), 'Variable selection for model-based clustering', *J. Amer. Stat. Assoc.* **101**, 168–178.

Rand, W. M. (1971), 'Objective criteria for the evaluation of clustering methods', *Journal of the American Statistical Association* **66**, 846–850.

Rosenwald, A., Wright, G., Chan, W. C., Connors, J. M., Campo, E., Fisher, R. I., Gascoyne, R. D., Muller-Hermelink, H. K., Smeland, E. B. & Staudt, L. M. (2002), 'The use of molecular profiling to predict survival after chemotherapy for diffuse large B-cell lymphoma', *The New England Journal of Medicine* **346**, 1937–1947.

Shen, H. & Huang, J. Z. (2008), 'Sparse principal component analysis via regularized low rank matrix approximation', *Journal of Multivariate Analysis* **101**, 1015–1034.

Stranger, B., Forrest, M., Clark, A., Minichiello, M., Deutsch, S., Lyle, R., Hunt, S., Kahl, B., Antonarakis, S., Tavare, S., Deloukas, P. & Dermitzakis, E. (2005), 'Genome-wide associations of gene expression variation in humans', *PLOS Genetics* **1(6)**, e78.

Stranger, B., Forrest, M., Dunning, M., Ingle, C., Beazley, C., Thorne, N., Redon, R., Bird, C., de Grassi, A., Lee, C., Tyler-Smith, C., Carter, N., Scherer, S., Tavare, S., Deloukas, P., Hurles, M. & Dermitzakis, E. (2007), 'Relative impact of nucleotide and copy number variation on gene expression phenotypes', *Science* **315**, 848–853.

Sugar, C. A. & James, G. M. (2003), 'Finding the number of clusters in a dataset: an information-theoretic approach', *Journal of the American Statistical Association* **98**, 750–763.

Tamayo, P., Scanfeld, D., Ebert, B. L., Gillette, M. A., Roberts, C. W. M. & Mesirov, J. P. (2007), 'Metagene projection for cross-platform, cross-species characterization of global transcriptional states', *PNAS* **104**, 5959–5964.

Tebbens, J. & Schlesinger, P. (2007), 'Improving implementation of linear discriminant analysis for the high dimension / small sample size problem', *Computational Statistics and Data Analysis* **52**, 423–437.

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *J. Royal. Statist. Soc. B.* **58**, 267–288.

Tibshirani, R., Hastie, T., Narasimhan, B. & Chu, G. (2002), 'Diagnosis of multiple cancer types by shrunken centroids of gene expression', *Proc. Natl. Acad. Sci.* **99**, 6567–6572.

Tibshirani, R., Hastie, T., Narasimhan, B. & Chu, G. (2003), 'Class prediction by nearest shrunken centroids, with applications to DNA microarrays', *Statistical Science* **18**, 104–117.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. & Knight, K. (2005), 'Sparsity and smoothness via the fused lasso', *J. Royal. Statist. Soc. B.* **67**, 91–108.

Tibshirani, R. & Walther, G. (2005), 'Cluster validation by prediction strength', *J. Comp. Graph. Stat.* **14**(3), 511–528.

Tibshirani, R., Walther, G. & Hastie, T. (2001), 'Estimating the number of clusters in a dataset via the gap statistic', *J. Royal. Statist. Soc. B.* **32**(2), 411–423.

Tibshirani, R. & Wang, P. (2008), 'Spatial smoothing and hotspot detection for CGH data using the fused lasso', *Biostatistics* **9**, 18–29.

Trendafilov, N. & Jolliffe, I. (2006), 'Projected gradient approach to the numerical solution of the scotlass', *Computational Statistics and Data Analysis* **50**, 242–253.

Trendafilov, N. & Jolliffe, I. (2007), 'DALASS: Variable selection in discriminant analysis via the LASSO', *Computational Statistics and Data Analysis* **51**, 3718–3736.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D. & Altman, R. (2001), 'Missing value estimation methods for DNA microarrays', *Bioinformatics* **16**, 520–525.

Venkatraman, E. & Olshen, A. (2007), 'A faster circular binary segmentation algorithm for the analysis of array CGH data', *Bioinformatics* **6**, 657–663.

von Luxburg, U. (2007), 'A tutorial on spectral clustering', *Statistics and Computing* **17**, 395–416.

Waaijenborg, S., Verselewel de Witt Hamer, P. & Zwinderman, A. (2008), 'Quantifying the association between gene expressions and DNA-markers by penalized canonical correlation analysis', *Statistical Applications in Genetics and Molecular Biology* **7**, Article 3.

Wang, S. & Zhu, J. (2008), 'Variable selection for model-based high-dimensional clustering and its application to microarray data', *Biometrics* **64**, 440–448.

Witten, D. & Tibshirani, R. (2009), 'Extensions of sparse canonical correlation analysis, with application to genomic data', *Statistical Applications in Genetics and Molecular Biology* **8(1)**, Article 28, http://www.bepress.com/sagmb/vol8/iss1/art28.

Witten, D. & Tibshirani, R. (2010), 'A framework for feature selection in clustering', *To appear in Journal of the American Statistical Association* .

Witten, D., Tibshirani, R. & Hastie, T. (2009), 'A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis', *Biostatistics* **10(3)**, 515–534.

Wold, S. (1978), 'Cross-validatory estimation of the number of components in factor and principal components models', *Technometrics* **20**, 397–405.

Xie, B., Pan, W. & Shen, X. (2008), 'Penalized model-based clustering with cluster-specific diagonal covariance matrices and grouped variables', *Electronic Journal of Statistics* **2**, 168–212.

Xu, P., Brock, G. & Parrish, R. (2009), 'Modified linear discriminant analysis approaches for classification of high-dimensional microarray data', *Computational Statistics and Data Analysis* **53**, 1674–1687.

Yuan, M. & Lin, Y. (2007), 'Model selection and estimation in regression with grouped variables', *Journal of the Royal Statistical Society, Series B* **68**, 49–67.

Zou, H. (2006), 'The adaptive lasso and its oracle properties', *Journal of the American Statistical Association* **101**, 1418–1429.

Zou, H. & Hastie, T. (2005), 'Regularization and variable selection via the elastic net', *J. Royal. Stat. Soc. B.* **67**, 301–320.

Zou, H., Hastie, T. & Tibshirani, R. (2006), 'Sparse principal component analysis', *Journal of Computational and Graphical Statistics* **15**, 265–286.