Hierarchical Clustering of Observations and Features in High-dimensional Data

by

Hongyang (Fred) Zhang

M.Sc., The University of British Columbia, 2012

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Statistics)

The University of British Columbia
(Vancouver)

August 2017

Abstract

In this thesis, we present new developments of hierarchical clustering in high-dimensional data. We consider different use cases of hierarchical clustering, namely, clustering observations for exploratory analysis and clustering high-dimensional features for adaptive feature grouping and ensembling.

We first focus on the clustering of observations. In high-dimensional data, the existence of potential noise features and outliers poses unique challenges to the existing hierarchical clustering techniques. We propose the Robust Sparse Hierarchical Clustering (RSHC) and the Multi-rank Sparse Hierarchical Clustering (MrSHC) to address these challenges. We show that via robust feature selection techniques, both RSHC and MrSHC can handle the potential existence of noise features and outliers in high-dimensional data and result in better clustering accuracy and interpretation comparing to the existing hierarchical clustering methods.

We then consider clustering of features in high-dimensional data. We propose a new hierarchical clustering technique to adaptively divide the large number of features into subgroups called Regression Phalanxes. Features in the same Regression Phalanx work well together as predictors in a pre-defined regression model. Then models built on different Regression Phalanxes are considered for further ensembling. We show that the ensemble of Regression Phalanxes resulting from the hierarchical clustering produces further gains in prediction accuracy when applied to an effective method like Lasso or Random Forests.

Lay Summary

There has been a surge in the number of high-dimensional data sets – data sets containing a large number of features and a relatively small number of observations – due to the rapidly growing ability to collect and store information. For example, in microarray studies, expression levels of thousands of genes are measured on only a few samples; X-ray experiments are typically conducted on a limited number of samples but the spectra over thousands of wavelengths are measured on each sample. Mining such high-dimensional or large-flat data is an urgent problem of great practical importance. In this thesis, we concentrate on the applications of hierarchical clustering, a widely used data mining technique, on high-dimensional data. We introduce several novel techniques that successfully extract and combine key information from the usually noisy high-dimensional data using hierarchical clustering. The proposed methods have well-established performance and broad applications in exploratory analysis and prediction.

Preface

This thesis is written up under the supervision of my supervisor Prof. Ruben H. Zamar. During our weekly research meetings, we discussed thoroughly regarding the research topics, proposed methods and the related analyses. For all the Chapters of this thesis, I conducted all the the computations, data analyses, first-drafts and follow-up revisions. Prof. Zamar and Prof. William Welch spent a huge amount of time checking the results and proof-reading.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to express my special appreciation and thanks to my advisor Professor Ruben H. Zamar, you have been a tremendous mentor and friend for me. I would like to thank you for encouraging my research and for allowing me to grow as a data scientist. Your advice on both research as well as on my career have been priceless.

I would also like to thank my committee members, professor William Welch and professor Matías Salibián-Barrera for serving as my committee members and for your brilliant comments and suggestions.

A special thanks to my mother Lirong Zhao, my father Ming Zhang and my grandma Xiuzhi Tian for your unconditional love and support. At the end I would like to express appreciation to my beloved wife Xin Zhao who has always been there for me through my ups and downs. Words cannot express how grateful I am to all of you.

Chapter 1

Introduction

There has been a surge in the number of high-dimensional data sets – data sets containing a large number of features and a relatively small number of observations – due to the rapidly growing ability to collect and store information in medical research and other fields. For example, in microarray studies, expression levels of thousands of genes (features) are measured on only a few samples; X-ray experiments are typically conducted on a limited number of samples (e.g. less than a hundred) but the spectra over thousands of wavelengths (features) are measured on each sample. Mining such high-dimensional or large-flat data is an urgent problem of great practical importance.

Clustering is an important data mining technique widely used in various fields. We mainly focus on the hierarchical clustering, one of the most widely used clustering algorithms with a lot of unique advantages for exploratory data analysis. In this thesis, we use hierarchical clustering for the following two tasks.

- Exploratory analysis – Hierarchical clustering of observations in high-dimensional data.

  Hierarchical clustering is a well established method for exploratory data analysis. However, in high-dimensional settings, the performance of traditional hierarchical clustering algorithms can be distorted due to the possible existence of noise features and data contaminations. Surprisingly, there exist only limited attempts to improve the performance of hierarchical clus-

tering in high-dimensional settings. In Chapter 2 and Chapter 3, we propose two novel hierarchical clustering frameworks which yield better performance than the existing hierarchical clustering methods in high-dimensional data.

- Regression – Hierarchical clustering of features in high-dimensional data.

  Ensemble methods prove to be useful in various regression tasks by producing superior prediction accuracies. In Chapter 4, we propose a novel ensemble method via hierarchical clustering of features. The goal is to discover subsets of features so that the features in each subset work well together as predictors in a pre-specified regression procedure. By defining a proper dissimilarity measure of the features, the hierarchical clustering is able to group the features together according to their combined predictive power. Then we propose to ensemble all the regression models built on different subsets of features to build a final ensemble model.

The rest of the chapter is organized as follows. In Section 1.1, we briefly describe the hierarchical clustering. Section 1.2 presents an introduction of hierarchical clustering of observations and existing proposals for high-dimensional data. Section 1.3 presents an introduction of hierarchical clustering of features and briefly describes some of the regression methods we use in the following chapters. Finally, Section 1.4 presents the organization of the remainder of the thesis.

1.1 Hierarchical Clustering

In data mining and statistics, hierarchical clustering is a method of clustering analysis which aims to categorize observations into a hierarchical set of groups. There are two general strategies for hierarchical clustering:

- Agglomerative or "bottom up" approach.

  In the agglomerative approach, each observation forms its own cluster at the beginning of the process. Then the pairs of clusters are iteratively merged to form clusters of higher hierarchy.

- Divisive or "top down" approach.

In the divisive approach, a single cluster containing all the observations gets initialized, which further splits into sub-clusters. Splits are performed recursively to generate the hierarchy of clusters.

Unless stated otherwise, we take agglomerative approach as the default approach for hierarchical clustering throughout this thesis. This is not only because the agglomerative approach is usually considered as the default approach for hierarchical clustering but also because it is less computational burdensome comparing to the divisive approach. The merges in hierarchical clustering are performed in a greedy manner. In each iteration, only one pair of clusters are merged. In order to choose the pair of clusters being merged, a measure of dissimilarity between clusters of observations needs to be specified. This is usually achieved by making appropriate choices of the dissimilarity measure (a measure of distance between pairs of observations), and the linkage criterion which defines the dissimilarity of clusters as a function of the pairwise distances of observations (defined by the chosen dissimilarity measure) in the clusters. The results of hierarchical clustering are usually presented in a tree structure called dendrogram. In the following sections, we present some common choices of both the dissimilarity measures and the linkage criteria, some sample dendrograms and a brief discussion on the advantages and disadvantages of the hierarchical clustering methods.

1.1.1 Dissimilarity Measures

Suppose we have a pair of observations represented as p-dimensional vectors $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)^T$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_p)$, a selection of commonly used dissimilarity measures as follows.

- Euclidean distance: $||\boldsymbol{x} - \boldsymbol{y}||_2 = \sqrt{\sum_i (x_i - y_i)^2}$.
- Squared Euclidean distance: $||\boldsymbol{x} - \boldsymbol{y}||_2^2 = \sum_i (x_i - y_i)^2$.
- Manhattan distance: $||\boldsymbol{x} - \boldsymbol{y}||_1 = \sum_i |x_i - y_i|$.
- Maximum distance: $||\boldsymbol{x} - \boldsymbol{y}||_\infty = \max_i |x_i - y_i|$.
- Mahalanobis distance: $\sqrt{(\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{S}^{-1} (\boldsymbol{x} - \boldsymbol{y})}$, where $\boldsymbol{S}$ is the covariance matrix of the data set.

It is apparent that different choices of the dissimilarity measure will affect, sometimes significantly, on the resulting distance of the observations. A pair of observations may be considered close by using certain dissimilarity measures but farther away by others. For example, suppose there are two cluster centres $(2,2,2)$ and $(3,0,0)$. Then $(0,0,0)$ is considered to be closer to $(3,0,0)$ with squared Euclidean distance. However, with maximum distance, it is considered to be closer to $(2,2,2)$. Therefore, the choice of the dissimilarity measure can consequently affect the clustering results, especially the shape of the resulting clusters.

1.1.2 Linkage Criteria

Suppose we have two clusters X and Y each containing several observations and the chosen dissimilarity measure is $d(\boldsymbol{x},\boldsymbol{y})$. Some commonly choices of linkage criteria and their corresponding calculations of distance between X and Y are presented as follows.

- Complete linkage: $\max\{d(\boldsymbol{x},\boldsymbol{y}) : \boldsymbol{x} \in X, \boldsymbol{y} \in Y\}$.
- Single linkage: $\min\{d(\boldsymbol{x},\boldsymbol{y}) : \boldsymbol{x} \in X, \boldsymbol{y} \in Y\}$.
- Average linkage: $\frac{1}{|X||Y|}\sum_{\boldsymbol{x}\in X}\sum_{\boldsymbol{y}\in Y} d(\boldsymbol{x},\boldsymbol{y})$.
- Ward's linkage: the distance is measured by the decrease in variance for the clusters being merged. At each merging step, a pair of clusters $A = \{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_{n_A}\}$ and $B = \{\boldsymbol{b}_1,\ldots,\boldsymbol{b}_{n_B}\}$ are merged if the following criterion is minimized.

  $$\frac{n_A n_B}{n_A+n_B}\left(\frac{2}{n_A n_B}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B} d(\boldsymbol{a}_i,\boldsymbol{b}_j) - \frac{1}{n_A^2}\sum_{i=1}^{n_A}\sum_{j=1}^{n_A} d(\boldsymbol{a}_i,\boldsymbol{a}_j) - \frac{1}{n_B^2}\sum_{i=1}^{n_B}\sum_{j=1}^{n_B} d(\boldsymbol{b}_i,\boldsymbol{b}_j)\right)$$

  If squared Euclidean distance is used for $d(\cdot,\cdot)$, the above criterion is equivalent to Ward's minimum variance criterion [30]. Other distance measures such as Manhattan or Mahalanobis distance are also used as $d(\cdot,\cdot)$ in practice [2].

The linkage criterion defines the dissimilarity of clusters as a function of the pairwise distances of observations (defined by the chosen dissimilarity measure) in the

clusters. Given a chosen dissimilarity measure, difference choices of the linkage criterion will yield sometimes significantly different clusters.

1.1.3 Advantages and Disadvantages

Hierarchical clustering has a broad range of applications such as microarray data analysis, digital imaging, stock prediction, text mining, etc. This is because it has a unique set of advantages over some other clustering techniques such as K-means, model-based clustering, etc. Some of the advantages are listed as follows.

- No need to specify the number of clusters.

  Clustering techniques such as K-means require to specify the target number of clusters in advance. However, the target number of clusters is usually unknown. On the contrary, hierarchical clustering results in a hierarchy of clusters visualized as a dendrogram, so there is no need to specify the number of clusters in advance. After visualizing the dendrogram, one can then get the desired number of clusters by cutting the dendrogram accordingly. Moreover, dendrograms will reveal any intrinsic nested cluster structures in the data that are otherwise hard to discover.

- Flexibility

  - Flexibility on the choice of linkage criteria.

    There are numerous choices of linkage criteria available, which adapts to different types of data and clustering goals.

  - Versatility due to numerous combinations of dissimilarity measures and linkage criteria.

    Due to the flexibility on both the choice of dissimilarity measures and the choice of linkage criteria, hierarchical clustering is very versatile in discovering clusters of different shapes and it is usually possible to find a certain combination of dissimilarity measure and linkage criteria that satisfy your need. In practice, the combination of dissimilarity measure and linkage criteria can be chosen based on the perceived usefulness of the resulting dendrograms.

A known disadvantage of agglomerative hierarchical clustering is the high computational cost. The complexity of the agglomerative approach is $O(n^2 \log(n))$ with n denoting the number of observations, which does not scale well for a large number of observations. However, in our applications, either the data set contains a limited number of observations (high-dimensional flat data sets) or we conduct smart filtering to control the computational time of hierarchical clustering.

1.2 Hierarchical Clustering of Observations in High-dimensional Data

Classical hierarchical clustering, likewise most of the traditional clustering techniques, is challenged by high-dimensional data sets. Although a large number of features may allow for a better coverage of clustering features (i.e. features that help explain the true underlying clusters), a large proportion of the features may contain no information about the cluster structures. That is, clusters can usually be well separated from a relatively small number of features, but not if all the features are used to compute the distance between objects due to the perturbation introduced by noise or non-informative features. Furthermore, interpretability can be impeded when the clustering procedure uses an unnecessarily large number of variables.

Therefore, feature selection becomes a key feature of clustering methods when applying them to high-dimensional data sets. The goal of feature selection is to keep only the important features for clustering while discarding the noise features. Clustering methods with feature selection capabilities are often referred to as sparse clustering methods. In comparison with the classical clustering methods where all the features are used, sparse clustering has the following advantages:

- If the clusters differ only with respect to a small number of features, then sparse clustering is likely to identify the clusters more accurately than non-sparse clustering methods.

- The results of sparse clustering are more interpretable because the clusters are determined by a smaller number of features.

Multiple feature selection approaches have been proposed for clustering meth-

ods such as K-means (e.g. Witten and Tibshirani [38], Sun et al. [31]) and model-based clustering (e.g. Raftery and Dean [25], Pan and Shen [23], Wang and Zhu [37], Xie et al. [40]). However, there has been less research for hierarchical clustering. A brief survey of such proposals is given below.

Let $\boldsymbol{X}$ be an $n \times p$ data matrix, with n observations and p features. Let $d_{i,i'} = d(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$ be a measure of dissimilarity between observations $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$ ($1 \leq i, i' \leq n$), which are the rows i and i' of the data matrix $\boldsymbol{X}$. We will assume that d is additive in the features: $d(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = d_{i,i'} = \sum_{j=1}^{p} d_{i,i',j}$, where $d_{i,i',j}$ indicates the dissimilarity between observations i and i' along feature j. Unless specified otherwise, our examples and simulations take d equal to the squared Euclidean distance, $d_{i,i',j} = (X_{ij} - X_{i'j})^2$. However, other dissimilarity measures are possible, such as the absolute difference $d_{i,i',j} = |X_{ij} - X_{i'j}|$. A nice property of hierarchical clustering is that the potential outliers are clearly exposed in the dendrogram. The absolute difference is more robust than the squared Euclidean distance in the presence of outliers, however, a robust dissimilarity measure such as trimmed absolute distance may not be desirable in the context of hierarchical clustering. For example, we generate a sample data set containing 60 observations and 20 features. The observations are well separated into 3 clusters of equal size. A value in the data set is replaced with a large outlier. As we can see from Figure 1.1, the contaminated observation is exposed in its standalone cluster with squared Euclidean distance. On the contrary, it is missed from the dendrogram under the trimmed absolute distance since the robust measure down-weights or ignores its distances from the rest of the samples.

Friedman and Meulman [17] proposed *clustering objects on subsets of attributes* (COSA). COSA employs a criterion, related to a weighted version of K-means clustering, to automatically detect subgroups of objects that preferentially cluster on subsets of the attribute variables rather than on all of them simultaneously. An extension of COSA for hierarchical clustering was also proposed. The algorithm is quite complex and requires multiple tuning parameters. Moreover, as noted by Witten and Tibshirani [38], this proposal does not truly result in a sparse clustering because all the variables have nonzero weights.

Witten and Tibshirani [38] proposed a new framework for sparse clustering that

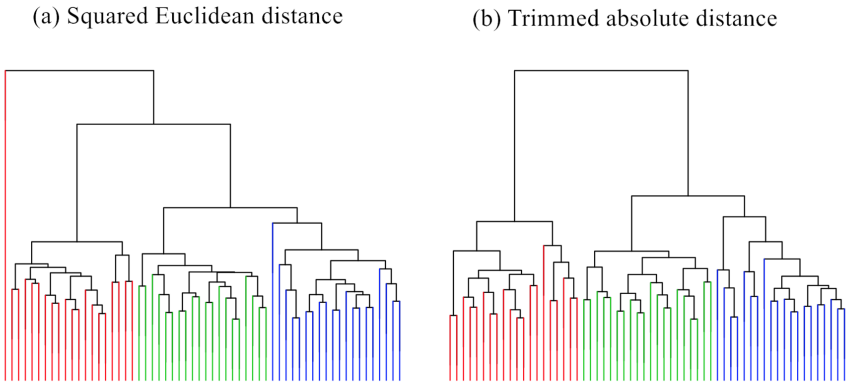

Figure 1.1: Dendrograms with squared Euclidean distance and trimmed absolute distance (10% largest distances are trimmed) on a data set with a outlier. Colors are used to indicate the true underlying cluster.

can be applied to procedures that optimize a criterion of the form

$$\max_{\Theta \in G} \left\{ \sum_{j=1}^{p} f_j(\mathbf{X}_j, \Theta) \right\}, \tag{1.1}$$

where $\mathbf{X}_j = (X_{1j}, X_{2j}, ..., X_{nj})^T \in \mathbb{R}^n$ denotes the observed j-th feature.

Each $f_j(\mathbf{X}_j, \Theta)$ is a function that solely depends on the j-th feature and Θ is a set of unknown parameters taking values on G. For example, in K-means, it can be shown that for a fixed number K of clusters,

$$f_j(\mathbf{X}_j, \Theta) = \frac{1}{n} \sum_{i=1}^{n} \sum_{i'=1}^{n} d_{i,i',j} - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,i' \in C_k} d_{i,i',j},$$

where n_k is the number of observations in cluster k, C_k is the set of indices of observations in cluster k and $d_{i,i',j}$ is the squared Euclidean distance between observation i and i' on the j-th feature. In this chapter, we only consider squared Euclidean measure for $d_{i,i',j}$.

The optimizing criterion in (1.1) is not sparse because the criterion involves all the features. To introduce sparsity Witten and Tibshirani [38] modified criterion

(1.1) as follows:

$$\max_{\boldsymbol{w},\Theta\in G}\left\{\sum_{j=1}^{p} w_j f_j(\mathbf{X}_j,\Theta)\right\} \text{ subject to } ||\boldsymbol{w}||_2^2 \leq 1,\, ||\boldsymbol{w}||_1 \leq s \text{ and } w_j \geq 0. \tag{1.2}$$

Here $\boldsymbol{w} = (w_1, w_2, \ldots, w_p)$ is a vector of weights for each feature, $||\boldsymbol{w}||_2^2$ is squared L2-norm on $\boldsymbol{w}$, and $||\boldsymbol{w}||_1$ is L1-norm on $\boldsymbol{w}$. A feature with zero-weight is clearly not used in the criterion.

Hierarchical clustering does not optimize a criterion like (1.1) and, therefore, does not directly fit into Witten and Tibshirani [38] sparse clustering framework (1.2). To overcome this difficulty they casted the dissimilarity matrix $\left\{d_{i,i'}\right\}_{n\times n}$ as the solution of an optimization problem as follows:

$$\max_{\mathbf{U}\in\mathbb{R}^{n\times n}}\left\{\sum_{j=1}^{p}\sum_{i,i'=1}^{n} d_{i,i',j}U_{i,i'}\right\} \text{ subject to } \sum_{i,i'=1}^{n} U_{i,i'}^2 \leq 1. \tag{1.3}$$

It can be shown that the solution $\widehat{U}_{i,i'}$ to (1.3) is proportional to the dissimilarity matrix, that is, $\widehat{U}_{i,i'} \propto d_{i,i'}$. The criterion in (1.3) is a special case of (1.1) when we let $f_j(\mathbf{X}_j,\Theta) = \sum_{i,i'=1}^{n} d_{i,i',j}U_{i,i'}$. Now sparse hierarchical clustering can be achieved by obtaining a sparse dissimilarity matrix. Now *the sparse hierarchical clustering criterion* (note that this is a preprocessing step of the hierarchical clustering) can be defined as follows:

$$\max_{\boldsymbol{w},\mathbf{U}\in\mathbb{R}^{n\times n}}\left\{\sum_{j=1}^{p} w_j \sum_{i,i'=1}^{n} d_{i,i',j}U_{i,i'}\right\} \text{ subject to } \sum_{i,i'=1}^{n} U_{i,i'}^2 \leq 1, ||\boldsymbol{w}||_2^2 \leq 1, ||\boldsymbol{w}||_1 \leq s. \tag{1.4}$$

The constraint $w_j \geq 0$ has been removed because $d_{i,i',j} \geq 0$ for all $1 \leq i,i' \leq n$ and $1 \leq j \leq p$. The solution to (1.4) can be obtained using sparse principal component (SPC) proposed in Witten et al. [39] as follows: Let $\boldsymbol{u}$ be a vector of length n^2 that contains all elements in $(U_{i,i'})_{n\times n}$ and $\boldsymbol{D}$ be a $n^2 \times p$ matrix whose j-th column contains the n^2 elements of $\left\{d_{i,i',j}\right\}_{n\times n}$ – the dissimilarity matrix calculated from the j-th feature alone. Now the criterion in (4) is equivalent to the following:

$$\max_{\boldsymbol{w},\boldsymbol{u}}\left\{\boldsymbol{u}^T\boldsymbol{D}\boldsymbol{w}\right\} \text{ subject to } ||\boldsymbol{u}||_2^2 \leq 1, ||\boldsymbol{w}||_2^2 \leq 1, ||\boldsymbol{w}||_1 \leq s. \tag{1.5}$$

This reduces to applying SPC on *the transformed dissimilarity matrix*, $\boldsymbol{D}$. It can also be shown that the solution to (1.5) satisfies: $\widehat{\boldsymbol{u}}^T \propto \boldsymbol{D}\widehat{\boldsymbol{w}}$. As $\widehat{\boldsymbol{w}}$ is sparse, so is $\widehat{\boldsymbol{u}}$. Thus, by re-arranging the elements in $\widehat{\boldsymbol{u}}$ into a $n \times n$ dissimilarity matrix $\widehat{\boldsymbol{U}}$, we obtain a sparse dissimilarity matrix which only contains the information from a subset of selected features. Finally, sparse hierarchical clustering can be obtained by applying classical hierarchical clustering on the sparse dissimilarity matrix $\widehat{\mathbf{U}}$. Witten and Tibshirani [38] showed, using a simulated dataset and a genomic dataset, that their proposed sparse hierarchical clustering results in more accurate identification of the underlying clusters and more interpretable results than standard hierarchical clustering and COSA when applied on datasets with noise features.

Witten and Tibshirani [38] demonstrated on a simulated and a real data set that SHC results in accurate identification of the underlying clusters. Moreover, SHC gives far more interpretable results than classical hierarchical clustering when the data set contains noise features. However, there are some limitations of the SHC framework that jeopardize its performance in real applications. We present these limitations below.

1. Unable to discover features with complex structures.

   The structures of features in high-dimensional data are usually complex. We show that when features contain complex structures, SHC has its limitation in discovering the important features from the noise.

2. Vulnerable to data contaminations.

   Given the large number of features in high-dimensional data sets, the presence of data contaminations becomes more likely. The SHC framework is sensitive to the outliers and its clustering performance deteriorates severely under data contamination.

3. Non-scalable to even medium-sized data. SHC operates on the transformed dissimilarity matrix $\boldsymbol{D}$ (a $n^2 \times p$ matrix). Therefore, it won't scale to medium-sized data sets with relatively big n.

In Chapter 2 and Chapter 3, we propose two novel sparse hierarchical cluster-

ing methods to remedy the three limitations of the existing SHC method simultaneouly.

1.3 Hierarchical Clustering of Features in High-dimensional Data

Ensemble methods prove to be useful in various regression tasks by producing superior prediction accuracies. In Chapter 4, we propose a novel ensemble method via hierarchical clustering of features. The goal is to discover subsets of features so that the features in each subset work well together as predictors in a pre-specified regression procedure called "base regression model". By defining a proper dissimilarity measure of the clusters of features, the hierarchical clustering is able to group the features together according to their combined predictive power. Then we propose to ensemble all the regression models built on different subsets of features to build a final ensemble model.

This ensemble approach is very different from the existing ensemble approaches. It separates the feature space into distinct clusters on which different base regression models are trained for further ensemble. This approach is very flexible since in principle, any regression methods, even the existing ensemble methods, can be used as base regression models. We briefly review the candidate base regression models we considered in this thesis, namely Lasso and Random Forests.

1.3.1 Lasso

Lasso (least absolute shrinkage and selection operator), proposed by Tibshirani [32], is a regularized linear regression method method that performs feature selection in linear models via regularization. It constrains the size of the regression coefficients by forcing the sum of the absolute values of their values to be less than a chosen value, which shrinks the coefficients and forces certain coefficients to exactly zero. As a result, the model fitting process automatically select only a subset of the features (the ones with non-zero coefficients) in the final Lasso model. It is shown that via regularization and shrinkage, Lasso is able to enhance the prediction accuracy and interpretability of the model at the same time.

Consider a data set $\boldsymbol{X}$ containing n observations and p features, each obser-

vation corresponding to a single response in $\boldsymbol{y}$. Denote the i-th observation as $\boldsymbol{x}_i = (x_1, x_2, \ldots, x_p)^T$ and its corresponding response as y_i. Lasso solves the following L1-regularized least square problem.

$$\min_{\beta_0, \boldsymbol{\beta}} \left\{ \frac{1}{N} \sum_{i=1}^{N} (y_i - \beta_0 - \boldsymbol{x}_i^T \boldsymbol{\beta})^2 \right\} \text{ subject to } \sum_{j=1}^{p} |\beta_j| \leq t, \tag{1.6}$$

where t is a prespecified tuning parameter that controls the amount of regularization and shrinkage.

Equation 1.6 can re-written in a more compact fashion.

$$\min_{\beta_0, \boldsymbol{\beta}} \left\{ \frac{1}{N} \|\boldsymbol{y} - \beta_0 - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \right\} \text{ subject to } \|\boldsymbol{\beta}\|_1 \leq t, \tag{1.7}$$

where $\|Z\|_p = \left(\sum_{i=1}^{N} |Z_i|^p\right)^{1/p}$ represents the L_p norm.

Since

$$\hat{\beta}_0 = \bar{\boldsymbol{y}} - \bar{\boldsymbol{x}}^T \boldsymbol{\beta}, \tag{1.8}$$

therefore,

$$y_i - \hat{\beta}_0 - \boldsymbol{x}_i^T \boldsymbol{\beta} = y_i - (\bar{\boldsymbol{y}} - \bar{\boldsymbol{x}}^T \boldsymbol{\beta}) - \boldsymbol{x}_i^T \boldsymbol{\beta} = (y_i - \bar{\boldsymbol{y}}) - (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T \boldsymbol{\beta}, \tag{1.9}$$

it is a standard procedure to normalize the features – subtract feature means and then standardize features $\left(\sum_{i=1}^{N} x_{ij}^2 = 1\right)$. As a result, the Lasso solution does not depend on the measurement scale of the features.

Suppose the features in $\boldsymbol{X}$ are normalized, we can further rewrite the optimization problem as follows.

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \left\{ \frac{1}{N} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \right\} \text{ subject to } \|\boldsymbol{\beta}\|_1 \leq t. \tag{1.10}$$

This is equivalent to the following problem according to the Lagrangian form.

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \left\{ \frac{1}{N} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\}. \tag{1.11}$$

There is a one-to-one relationship between t and λ but the exact relationship varies

and it's data dependent.

By solving Equation (1.11), which is a convex optimization problem, the Lasso coefficients can be obtained.

1.3.2 Random Forests

To better understand Random Forests, we first presents two closely related preliminary methods: Regression Tree and Bagging.

Regression Tree

A regression tree is a binary decision tree. The root node of the tree contains all the training sample, which gets split into two child nodes. The the two child nodes split into their own child nodes recursively. Each split is determined by choosing a feature and its corresponding best split value. The goal of the split is to maximizing the gain in homogeneity in a greed matter. In the context of regression tree, the gain in homogeneity is usually represented by the decrease in sample variances of responses, i.e., sample response variance of the parent node subtracting the sum of sample response variances of the two child nodes. Therefore, among all the features, the feature and its split value that allow the maximum amount of gain for the current node are chosen. More specifically, at a particular node, the tree growing/splitting process chooses a split from all possible splits so that the resulting left and right child nodes are as homogeneous as possible. This is done by first finding each feature's best split value and then searching for the split that achieves the maximum gain. Since different features and split values may be chosen for different splits, a regression tree essentially partitions the feature space successively into smaller hyper-rectangles (represented by nodes) and each hyper-rectangle is as homogeneous as possible.

The tree splitting process stops according to the "stopping rules". A node will not be split and becomes a terminal node if:

- the node contains observations with identical responses (no more gain in homogeneity can be achieved),
- all the observations in the node have identical values for each feature,
- the tree size reaches the user-specified maximum tree size,

- the size of the node is smaller than the user-specified minimum node size,
- the child node size is smaller than the user-specified minimum child node size, and
- the amount of gain from the best split is smaller than the user-specified minimum amount.

For further reference, please see Breiman et al. [7] and Chapter 9 of Venables and Ripley [36].

Once the tree splitting process finishes, each branch of the tree ends in a terminal node. Each observation falls into one and exactly one terminal node, and each terminal node corresponds to a unique set of splits.

Bootstrap Aggregating (Bagging)

Bootstrap Aggregating, or bagging, is a general ensemble framework proposed by Breiman [5]. Bagging fits different regression or classification models on several bootstrap samples for further aggregation. A bootstrap sample is obtained by selecting a random sample with replacement of equal size of the training observations [14]. More specifically, given a training data set $\boldsymbol{X} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_n^T)^T$ with the corresponding response $\boldsymbol{y} = (y_1, \ldots, y_n)^T$, bagging has the followings steps.

- For $b = 1, \ldots, B$:
  - Sample, with replacement, n training observations from $\boldsymbol{X}$ and $\boldsymbol{y}$ accordingly. Denote the sample as $\boldsymbol{X}_b$ and $\boldsymbol{y}_b$.
  - Fit a regression or classification model M_b on $\boldsymbol{X}_b$ and $\boldsymbol{y}_b$.
- The prediction of a new observation $\boldsymbol{x}'$ can be calculated by averaging the predictions from all the B models $M_1, \ldots, M_B$ as $\hat{y}' = \frac{1}{B}\sum_{b=1}^{B} M_b(\boldsymbol{x}')$, or for classification applications, a majority vote can be applied.

Bagging often results in better prediction performance since it decreases the variance of the model by averaging, without increasing the bias. This benefits the methods that are not stable, namely, methods with high variance (perturbation of a training set leads to significant changes in predictions) and low bias (the fitting accuracy is high despite changes in training sets). In the case of a regression trees,

its predictions are highly sensitive to noise in the training sets, and the average of many regression trees becomes less sensitive, as long as the trees are not correlated. In order to reduce the correlation among trees, bootstrap sampling is an effective method as it de-correlate the trees by showing them different training sets.

The tuning parameter B in bagging represents the number of models (e.g. regression trees). Building a few hundred models is a common practice, and the number varies according to the size of the training set. In practice, B can be chosen via cross-validation, or according to the out-of-bag (OOB) error. The OOB error is calculated as follows. For each training observation $\boldsymbol{x}_i$, the mean prediction error is calculated using only the models that did not include $\boldsymbol{x}_i$ in their bootstrap sample.

Random Forests

Random Forests, a tree-based ensemble method proposed by Breiman [6], can be considered as an extension of bagging. Random Forests differ from bagging of trees mainly in the following way. In bagging, each tree is built on a bootstrap sample and each split is chosen from all candidate splits from all the features. However, in Random Forests, each split is chosen from only a random sample of m_{try} features instead of all the features. This extra source of randomness in tree splits further de-correlate the trees in Random Forests on top of the bootstrap samples and leads to better ensemble performance by further reducing the average correlation/variance. Given a training data set $\boldsymbol{X} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_n^T)^T$ with the corresponding response $\boldsymbol{y} = (y_1, \ldots, y_n)^T$, the steps of Random Forests are presented as follows.

- For $b = 1, \ldots, B$:
  - Sample, with replacement, n training observations from $\boldsymbol{X}$ and $\boldsymbol{y}$ accordingly. Denote the sample as $\boldsymbol{X}_b$ and $\boldsymbol{y}_b$.
  - Build a fully grown, unpruned tree on $\boldsymbol{X}_b$ and $\boldsymbol{y}_b$.
  - At each node of the tree, randomly select m_{try} features from which the best split-point of that node is chosen.
- The prediction of a new observation $\boldsymbol{x}'$ can be calculated by averaging the predictions from all the B models $M_1, \ldots, M_B$ as $\hat{y}' = \frac{1}{B}\sum_{b=1}^{B} M_b(\boldsymbol{x}')$, or for classification applications, a majority vote can be applied.

Fully grown trees in Random Forests lead to high prediction variability, however, the ensemble over many trees reduces the average variance and avoid overfitting.

The prediction accuracy of Random Forests can be assessed using the OOB errors. Random Forests also offers evaluation of importance of feature: Each feature is randomly permuted in the OOB samples and its impact on prediction error is measured; features corresponding to larger impact on the prediction error are considered more important. Other than the number of trees B, Random Forests has another important tuning parameter m_{try} which may impact its performance significantly. Breiman [6] suggests to use $m_{try} = \lfloor \sqrt{p} \rfloor$ for classification problems with p features, and $m_{try} = \lfloor p/3 \rfloor$ for regression problems with a minimum node size of 5. In practice, it is common to build a few hundred trees in Random Forests.

1.4 Thesis Organization

The rest of the thesis is organized as follows:

- Chapter 2. Robust Sparse Hierarchical Clustering.

  We first show how the existing SHC framework can be severely affected by outliers. Then we propose a novel approach called Robust Sparse Hierarchical Clustering which is robust to outliers in high-dimensional data. Based on this Chapter, a paper entitled "Robust sparse hierarchical clustering", co-authored by Leung, Zhang and Zamar, and myself will be submitted for publication.

- Chapter 3. Multi-rank Sparse Hierarchical Clustering.

  We first show the limitation of the SHC framework when clustering features have a complex structure. Then we propose a new framework called MrSHC which is more powerful when dealing with more complex structures. Based on this Chapter, a paper entitled "Multi-rank sparse hierarchical clustering", co-authored by Zhang and Zamar, is available in ArXiv, and will be soon submitted for publication.

- Chapter 4. Regression Phalanxes.

We focus on building an ensemble method for regression tasks via hierarchical clustering of features in high-dimensional data. We call the general ensemble framework Regression Phalanxes. We show that via simulation studies and several real data examples, Regression Phalanxes can further boost the performance of a good performing regression procedure such as Lasso or Random Forests. Based on this Chapter, a paper entitled "Regression Phalanxes", co-authored by Zhang, Welch and Zamar, will be submitted for publication.

Chapter 2

Robust Sparse Hierarchical Clustering

2.1 Introduction

In Section 1.2, we provide a brief review of the existing research on sparse hierarchical clustering. We mainly focus on the SHC approach proposed by Witten and Tibshirani [38]. It is demonstrated on a simulated and a real data set that SHC results in more accurate identification of the underlying clusters as well as more interpretable results than classical hierarchical clustering when the data set is high-dimensional and contains potentially a large portion of noise features. However, as we noted in Section 1.2, there are some limitations of the SHC framework. In this chapter, we focus on its vulnerability to data contaminations. Sparse clustering methods tend to suffer from data contaminations. Kondo et al. [19] pointed out that sparse K-means can be severely affected by a single outlying observation and proposed robust sparse K-means clustering (RSKC) to remedy this problem. Although classical hierarchical clustering methods themselves are fairly robust to outliers, we will show that a very small fraction of outliers can severely upset the results of SHC. In order to address this issue, we propose robust sparse hierarchical clustering (RSHC) with respect to outlying observations.

The rest of this chapter is organized as follows. We first provide some discussion and illustration of the effect of outliers on SHC. Then we propose two RSHC

methods based on L1-penalty and τ-estimators. Finally, we compare the RSHC approaches with other alternatives using simulation studies and two real data sets.

2.2 Effect of Outliers on Sparse Hierarchical Clustering

The performance of SHC may be negatively affected by a small fraction of outliers, which often happens in large and flat data sets. We demonstrate this using the following simulated example.

Consider a data set with 120 observations and 100 features. The observations are generated from 3 underlying clusters defined by the first two features:

$$X_{ij} = \begin{cases} \mu_i + \varepsilon_{ij} & j = 1,2 \\ \varepsilon_{ij} & j = 3,\ldots,100 \end{cases}$$

where $\varepsilon_{ij} \sim_{i.i.d} N(0,1)$ and

$$\mu_i = \begin{cases} -\mu & 1 \leq i \leq 40 \\ 0 & 41 \leq i \leq 80 \\ \mu & 81 \leq i \leq 120 \end{cases}$$

We choose $\mu = 3$.

We then consider the same data set but with outliers. The outliers are generated by replacing a total of 10 randomly selected entries in the noise features by values from $N(0,20)$.

Figure 2.1 shows the dendrograms from classical hierarchical clustering (HC) and SHC applied to the simulated data sets without and with outliers. An ideal clustering method would separate the three clusters labelled with red, green and blue respectively. In the case of no outliers, SHC outperforms HC and separates the underlying 3 clusters well. In the presence of outliers (labelled by black), both methods fail to identify the underlying clusters and generate mixed clusters.

To understand this behavior, we examine the expression for the feature weights which are found by solving the subgradient equation of (1.5) with respect to w_j

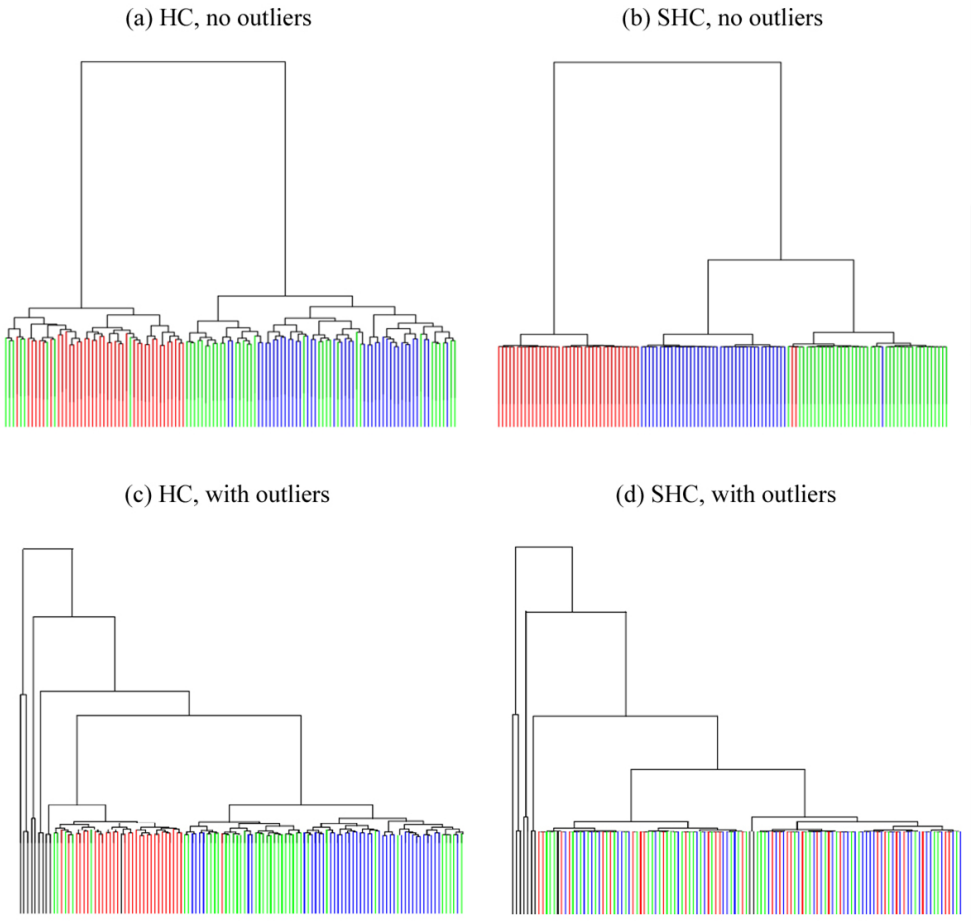

Figure 2.1: Dendrograms (with Ward's linkage) from HC and SHC applied to data sets without and with outliers. Colors are used to indicate the true underlying cluster.

(fixing $\boldsymbol{u}$):

$$w_j = \frac{S(\boldsymbol{D}_j^T \boldsymbol{u}, \Delta)}{||\boldsymbol{S}(\boldsymbol{D}^T \boldsymbol{u}, \Delta)||_2}, \quad \Delta > 0, \quad j = 1, ..., p,$$

where $\boldsymbol{D}_j$ is the j-th column of $\boldsymbol{D}$, S denotes the soft thresholding operator ($S(a, \Delta) = \text{sgn}(a)(|a| - \Delta)_+$, where $\Delta > 0$ is a constant, and $x_+ = \max(x, 0)$). See Witten et al. [39] for details. Notice that if some components in feature j, $\boldsymbol{D}_j$, are unusually

large, most of the weights will be assigned to that feature by the soft thresholding operator, that is, the corresponding weight w_j will be large. As an extreme case, all weights may be assigned to the contaminated features, breaking down SHC.

2.3 Robust Sparse Hierarchical Clustering

An key building block of SHC is sparse principal component (SPC) analysis, which needs to be robustified to achieve robust SHC.

First we introduce a new robust sparse principal component method based on τ-estimators (RSPC-τ). Then we propose two approaches for robust sparse hierarchical clustering (RSHC) based on RSPC-τ. Finally we provide a permutation method to choose the sparsity tuning parameters.

2.3.1 Robust Sparse Principal Component Based on τ-estimators

Consider a data set $\mathbf{Z}$ with n observations and p features. Let $\boldsymbol{a} = (a_1, \ldots, a_n)^T$, $\boldsymbol{b} = (b_1, \ldots, b_p)^T$, and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)^T$. In the context of principal component analysis, a_i ($i = 1, \ldots, n$) are the scores of the first principal component, b_j ($j = 1, \ldots, p$) are the loadings or weights for each feature, and $\boldsymbol{\mu}$ is the center of the data. Then the rank-1 approximation to $\mathbf{Z}$ is

$$\boldsymbol{Z}^{(1)}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\mu}) = \mathbf{1}_n \boldsymbol{\mu}^T + \boldsymbol{a}\boldsymbol{b}^T,$$

where $\mathbf{1}_n$ is a $n \times 1$ vector of 1's. The parameters, $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{\mu}$, can be estimated by solving the following minimization problem:

$$\min_{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\mu}} \sum_{j=1}^{p} s_j^2, \quad \text{subject to } ||\boldsymbol{b}||_2^2 = 1, \tag{2.1}$$

where s_j^2 is the sample variance of the residuals $\{r_{1j}, ..., r_{nj}\}$, where $r_{ij} = r_{ij}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\mu}) = Z_{ij} - Z_{ij}^{(1)}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\mu})$ and $Z_{ij}^{(1)} = \mu_j + a_i b_j$.

Boente and Salibian-Barrera [3] introduced robust principal component by using robust scale estimator instead of sample variance in (2.1) as follows:

$$\min_{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\mu}} \sum_{j=1}^{p} \widehat{\sigma}_j^2, \quad \text{subject to } ||\boldsymbol{b}||_2^2 = 1, \tag{2.2}$$

where $\widehat{\sigma}_j$ is a robust scale estimator for the residuals (see also Maronna [22]).

We propose robust sparse principal component by introducing an L1 penalty on the feature weights, $\boldsymbol{b}$, in (2.2):

$$\min_{\boldsymbol{a},\boldsymbol{b},\boldsymbol{\mu}} \left\{ \sum_{j=1}^{p} \widehat{\sigma}_j^2 + \lambda ||\boldsymbol{b}||_1 \right\}, \quad \text{subject to } ||\boldsymbol{b}||_2^2 = 1, \tag{2.3}$$

where λ is a tuning parameter that controls the sparseness of the feature weights, $\boldsymbol{b}$.

We use τ-estimators of scale for $\widehat{\sigma}_j$, $j = 1, ..., p$, because the estimator achieves robustness when there is contamination in the data and otherwise, it will produce similar results compared to the standard deviation [41].

To solve (2.3), we propose an iterative approximation approach, which is described in detail in the Appendix.

2.3.2 Robust Sparse Hierarchical Clustering

SHC uses SPC to handle noise variables. Likewise, our robust sparse hierarchical clustering uses RSPC-τ to handle both noise variables and outliers. We propose two approaches to perform RSHC. The first approach is a direct robustification of SHC, which we call Direct Robust Sparse Hierarchical Clustering (DRSHC). The second approach introduces a faster procedure which we call Fast Robust Sparse Hierarchical Clustering (FRSHC). We include DRSHC only for completion and comparison purposes, but we shall show that FRSHC outperforms DRSHC in both computational efficiency and clustering accuracy.

Direct Robust Sparse Hierarchical Clustering

DRSHC is a direct robustification of SHC, where we apply RSPC-τ on the $n^2 \times p$ tranformed dissimilarity matrix $\boldsymbol{D}$ instead of SPC. Details of DRSHC are presented in Algorithm 1.

Fast Robust Sparse Hierarchical Clustering

We propose another approach, FRSHC, which differs from DRSHC mainly in two aspects.

Algorithm 1 DRSHC

1: **Input:** A data matrix $\boldsymbol{X}_{n\times p}$, tuning parameter λ.
2: Compute the $n^2 \times p$ matrix $\boldsymbol{D}$.
3: Apply RSPC-τ using λ to $\boldsymbol{D}$ to obtain $\widehat{\boldsymbol{a}}_D$, $\widehat{\boldsymbol{b}}_D$ and $\widehat{\boldsymbol{\mu}}_D$. The number of non-zero weights in $\widehat{\boldsymbol{b}}_D$ is denoted q_D.
4: Set $\widehat{\boldsymbol{u}} = \boldsymbol{D}\widehat{\boldsymbol{b}}_D$, which is a vector of length n^2.
5: Re-arrange the elements in $\widehat{\boldsymbol{u}}$ into a $n \times n$ sparse dissimilarity matrix $\widehat{\boldsymbol{U}}$.
6: Apply classical hierarchical clustering (with any linkage of choice) to $\widehat{\boldsymbol{U}}$, which only involves q_D features.
7: **Output:** A dendrogram $\mathscr{HC}$, sparse weights for the p features $\widehat{\boldsymbol{w}} = \widehat{\boldsymbol{b}}_D$.

In FRSHC, RSPC-τ is directly applied to the original data $\boldsymbol{X}$ (a $n \times p$ matrix) instead of the transformed dissimilarity matrix $\boldsymbol{D}$ (a $n^2 \times p$ matrix). The intuitive for using $\boldsymbol{D}$ is rather unclear. The reason why $\boldsymbol{X}$ is used is because of the following reasons. A single feature is considered to be important if it contains enough variability so that it can potentially separate the clusters (in the extreme case, a feature with constant values will have no effect in separating clusters). Our goal is to find a subset of variables that separate the underlying clusters, therefore, we look for the sparse principal components that represent most of the variance in the original data. Since $\boldsymbol{D}$ is just a transformation of $\boldsymbol{X}$, if we can discover enough variance in $\boldsymbol{D}$, we would be able to discover the same variations in $\boldsymbol{X}$. Hence, FRSHC copes better with larger values of n due to the considerable difference in size between $\boldsymbol{X}$ and $\boldsymbol{D}$.

The sparse weights obtained from FRSHC can be negative because unlike $\boldsymbol{D}$, $\boldsymbol{X}$ may contain negative entries. The non-zero negative weights indicate that the corresponding variables have some influence for constructing the sparse PC. For our purposes, this could be recognized by assigning these variables weights proportional to the absolute value of the original negative weights. An alternative simpler approach would be to assign weight 1 to all the features with non-zero weights. This is essentially distinguishing important and unimportant variables. This will give a simpler interpretation for the clustering results. To distinguish between the procedures resulting from these two weighting schemes, we refer to them as FRSHC-AW (FRSHC with Absolute Weights) and FRSHC-IW (FRSHC with Indicator Weights) respectively.

Details of FRSHC are presented in Algorithm 2.

Algorithm 2 FRSHC

1: **Input:** A data matrix $\boldsymbol{X}_{n\times p}$, tuning parameter λ.
2: Apply RSPC-τ using λ to $\boldsymbol{X}$ to obtain $\widehat{\boldsymbol{a}}_X$, $\widehat{\boldsymbol{b}}_X$ and $\widehat{\boldsymbol{\mu}}_X$. The number of non-zero weights in $\widehat{\boldsymbol{b}}_X$ is denoted q_X.
3: Calculate the sparse dissimilarity matrix $\widehat{\boldsymbol{U}}$ as follows:
4: **For FRSHC-AW**:

- Calculate $\boldsymbol{D}$.
- Let $|\widehat{\boldsymbol{b}}_X|$ be the vector of absolute values of the elements in $\widehat{\boldsymbol{b}}_X$.
- Set $\widehat{\boldsymbol{u}} = \boldsymbol{D}|\widehat{\boldsymbol{b}}_X|$, a vector of length n^2.
- Re-arrange the elements in $\widehat{\boldsymbol{u}}$ into a $n \times n$ sparse dissimilarity matrix $\widehat{\boldsymbol{U}}$.

For FRSHC-IW:

- Calculate $\widehat{\boldsymbol{U}}$ directly using the features with non-zero weights in $\widehat{\boldsymbol{b}}_X$.

5: Apply classical hierarchical clustering (with any linkage of choice) to $\widehat{\boldsymbol{U}}$, which only involves q_X features.
6: **Output:** A dendrogram $\mathscr{HC}$, sparse weights for the p features $\widehat{\boldsymbol{w}} = \widehat{\boldsymbol{b}}_X$.

Remark 2.3.1. *In principle, other robust sparse principal component methods could be used in place of RSPC-τ. For comparison, we consider the method proposed in Croux et al. [11], which we call RSPC-Croux. The corresponding direct and fast robust sparse hierarchical clustering procedures are referred to as DRSHC-Croux and FRSHC-Croux.*

Choice of Tuning Parameters

In this section, we propose a permutation approach for choosing the tuning parameter λ of DRSHC and FRSHC. The procedure is described as follows:

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

2. For each candidate λ:

(a) Compute $J(\lambda) = \sum_{j=1}^{p} \widehat{\tau}_j^2 + \lambda ||\boldsymbol{b}||_1$, the objective obtained from optimizing RSPC-τ criterion (2.3) with λ on data $\boldsymbol{X}$ (for FRSHC) or transformed dissimilarity matrix $\boldsymbol{D}$ (for DRSHC).

(b) For $b = 1, \ldots, B$, compute $J_b(\lambda)$, the objective obtained by performing RSPC-τ with λ on the data $\boldsymbol{X}_b$ (for FRSHC) or the corresponding $\boldsymbol{D}_b$ (for DRSHC).

(c) Calculate $\text{Gap}(\lambda) = \frac{1}{B} \sum_{b=1}^{B} \log(J_b(\lambda)) - \log(J(\lambda))$.

3. Among a set of candidates $0 < \lambda_1 < \ldots < \lambda_k$ (e.g. $k = 20$), calculate the second order differences of $\text{Gap}(\lambda_i), i = 1, \ldots, k$, and choose λ^* that minimizes the differences $\lambda^* = \text{argmin}\{\text{Gap}(\lambda_{i+1}) - 2\text{Gap}(\lambda_i) + \text{Gap}(\lambda_{i-1})\}$. This corresponds to either the most prominent local maximum or a sudden increase in the Gap values. The behavior of this step is illustrated in Figure 2.2. The $\text{Gap}(\lambda_i), i = 1, \ldots, k$, obtained from a simulated example is plotted against the corresponding λ_i, and the red point corresponds to the chosen λ^i. We can see that this point is both the most prominent local maximum and also the point after a sudden increase. Extensive simulation results and real data applications confirm that this criterion works well in practice.

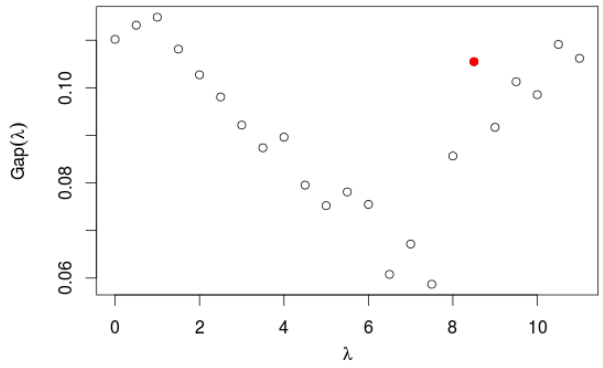

Figure 2.2: An illustration of choosing tuning parameter λ based on second order differences of $\text{Gap}(\lambda_i), i = 1, \ldots, k$

The resulting DRSHC and FRSHC with automatically selected λ are denoted as DRSHC-Gap and FRSHC-Gap respectively.

For cases where target numbers of important features are suggested by subject-area knowledge (as in the real data examples in Section 2.5), we propose using a backward approach to determine λ that will result in the target number of features q_X (for FRSHC) or q_D (for DRSHC).

Given q_X (or q_D), we choose λ by the following bisection procedure:

1. Choose $\lambda = \lambda^{(Mid)}$, where $\lambda^{(Mid)} = (\lambda^{(UB)} + \lambda^{(LB)})/2$, $\lambda^{(LB)} = 0$ and $\lambda^{(UB)} = c$. We use $c = 20$, but other choices can be considered.

2. Apply FRSHC (or DRSHC) using λ from Step 1.

3. If the resulting number of features for clustering from Step 2 is less than q_X (or q_D) then we update $\lambda^{(UB)} = \lambda$, else update $\lambda^{(LB)} = \lambda$.

4. Repeat Step 1–3 until we obtain $q_X \pm 2$ (or $q_D \pm 2$) features for clustering.

We have seen in our simulation studies that the above procedure converges quickly in a few iterations.

2.4 Simulation Study

We conduct simulation studies to compare the quality of dendrograms and the accuracy of feature selection of the following methods: classical hierarchical clustering (HC), SHC, DRSHC, FRSHC-AW, FRSHC-IW, FRSHC-Croux-AW, FRSHC-Croux-IW, FRSHC-Gap-AW and FRSHC-Gap-IW. We do not include DRSHC-Croux and DRSHC-Gap in the simulation studies due to their high computational burden. We show the results for all the methods with Ward's linkage. Similar results (not shown here) are obtained with other linkages.

We generate data sets $\mathbf{X}$ with $n = 60$ observations and $p = 1000$ features as follows. The observations are generated from 3 main underlying clusters C1, C2 and C3. Moreover, C1 and C2 have two nested sub-clusters labeled as C1a and C1b, and C2a and C2b. See Figure 2.3 for a graphical illustration.

More precisely, the clusters are determined by $q = 100$ features as follows:

$$X_{ij} = \begin{cases} \mu_i + \varepsilon_{ij} & j = 1, ..., 100 \\ \varepsilon_{ij} & j = 101, ..., 1000 \end{cases}$$

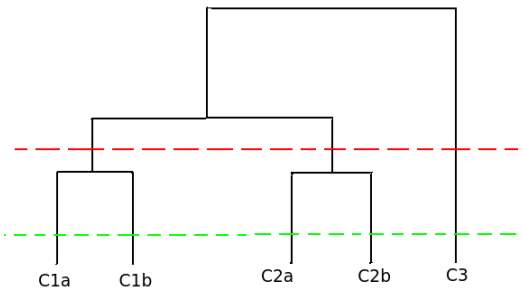

Figure 2.3: A graphical illustration of the hierarchical structure of the simulated clusters

where $\varepsilon_{ij} \sim_{i.i.d} N(0,1)$ and

$$\mu_i = \begin{cases} 0 & i = 1,\ldots,12 \ (i \in C1a) \\ \mu & i = 13,\ldots,24 \ (i \in C1b) \\ \mu+1 & i = 25,\ldots,36 \ (i \in C2a) \\ 2\mu+1 & i = 37,\ldots,48 \ (i \in C2b) \\ 2\mu+2 & i = 49,\ldots,60 \ (i \in C3) \end{cases}$$

We show the results for $\mu = 0.5$. Similar conclusions are obtained for other choices of μ.

We contaminate the data sets using outliers generated from the following models:

M0. No outliers.

M1. One large-mean outlier in one noise feature. A randomly selected observation of a noise feature is replaced by a value from $N(25,1)$.

M2. One large-mean outlier in one clustering feature. A randomly selected observation of a clustering feature is replaced by a value from $N(25,1)$.

M3. Large-variance outliers in noise features. For each of the five clusters, randomly select two entries in the noise features to be replaced by values

from $N(0, 15^2)$.

M4. Large-variance outliers in clustering features. For each of the five clusters, randomly select two entries in the clustering features to be replaced by values from $N(0, 15^2)$.

M5. Large-variance outliers in noise and clustering features. For each of the five clusters, randomly select two entries in the noise features and two entries in the clustering features to be replaced by values from $N(0, 15^2)$.

M6. Mild-mean outliers in noise features. Randomly select 50 noise features, and replace five random entries from each by values from $N(5, 1)$.

M7. Overall variance increase in some noise features. Fifty randomly selected noise features are generated from $N(0, 2^2)$.

We generate 100 data sets for each contamination model.

The quality of the resulting dendrograms is evaluated as follows. The dendrograms are cut at a high and a low level which are chosen to obtain exactly three and five clusters, respectively. Clusters with two or less observations are discarded. The classification error rate (CER) is then used to assess clustering accuracy [see 10]. We do this by first comparing the resulting labels from the three main clusters against the underlying true labels (C1, C2, C3). Next, we compare the resulting labels from the five clusters against the underlying true labels (C1a, C1b, C2a, C2b, C3).

The accuracy of feature selection is evaluated by the recall rate (RR). Let $\mathscr{J}$ be the set of indices corresponding to all the clustering features, and $|\mathscr{J}| = q$. The recall rate (RR) is calculated as follows:

$$RR(\mathscr{J}) = \frac{\sum_{j \in \mathscr{J}} I(\widehat{w}_j \neq 0)}{q},$$

where $I(\cdot)$ is an indicator function.

For SHC, DRSHC, FRSHC-AW, FRSHC-IW, FRSHC-Croux-AW, and FRSHC-Croux-IW, the target number of clustering features is given and equal to 100. For FRSHC-Gap-AW and FRSHC-Gap-AW, the number of clustering features is chosen automatically. The FRSHC-Gap methods generally identify the correct number

		HC	SHC	DRSHC	FRSHC		FRSHC-Gap		FRSHC-Croux	
					AW	IW	AW	IW	AW	IW
M0	K=3	0.024	0.000	0.006	0.000	0.000	0.001	0.000	0.170	0.046
	K=5	0.147	0.061	0.095	0.056	0.055	0.059	0.054	0.180	0.142
M1	K=3	0.024	0.393	0.006	0.000	0.000	0.001	0.000	0.172	0.049
	K=5	0.147	0.263	0.089	0.055	0.055	0.059	0.054	0.184	0.145
M2	K=3	0.024	0.313	0.006	0.000	0.000	0.000	0.001	0.178	0.049
	K=5	0.147	0.249	0.095	0.055	0.057	0.057	0.054	0.177	0.141
M3	K=3	0.025	0.408	0.005	0.000	0.000	0.001	0.000	0.166	0.033
	K=5	0.146	0.276	0.092	0.054	0.056	0.060	0.052	0.179	0.134
M4	K=3	0.025	0.299	0.009	0.003	0.002	0.006	0.002	0.144	0.054
	K=5	0.146	0.242	0.089	0.056	0.055	0.061	0.053	0.173	0.139
M5	K=3	0.029	0.375	0.012	0.002	0.001	0.006	0.002	0.143	0.059
	K=5	0.151	0.270	0.099	0.058	0.058	0.062	0.057	0.174	0.135
M6	K=3	0.035	0.234	0.006	0.001	0.001	0.001	0.000	0.184	0.064
	K=5	0.153	0.228	0.096	0.059	0.079	0.069	0.067	0.191	0.155
M7	K=3	0.035	0.338	0.018	0.000	0.000	0.000	0.000	0.460	0.309
	K=5	0.156	0.282	0.111	0.062	0.071	0.058	0.061	0.321	0.261

Table 2.1: Average CER for different clustering methods. The maximum SE of CER in this table is 0.01, while the maximum SE of CER for FRSHC methods is 0.003

of features used for clustering with the average numbers in **M0**–**M7** are 100.05, 100.22, 101.21, 100.87, 100.37, 99.94, 98.380, and 100.152, respectively.

Table 2.1 presents the average CER. When data sets are not contaminated, sparse methods in general produce more accurate clustering results than the classical HC; SHC and the FRSHC methods yield the smallest average CER, and DRSHC gives slightly higher CER. Under **M1**–**M7**, SHC gives poor clustering results. On the other hand, HC is robust against outliers but clearly affected by the large number of noise features. Finally, DRSHC and the FRSHC methods are highly robust against outliers and noise features. There is not much difference in performance between the FRSHC methods with absolute and indicator weights.

All the methods except for SHC show better clustering accuracy when dendrograms are cut into three rather than five clusters, as expected.

Table 2.2 presents the average RR. When data sets are not contaminated, SHC identifies most of the clustering features with the highest RR. DRSHC, the FRSHC methods, and the FRSHC-Gap methods give slightly smaller RR. Under **M1**–**M7**, SHC gives the lowest RR because it gives most of the weights to the contaminated

	SHC	DRSHC	FRSHC	FRSHC-Gap	FRSHC-Croux
M0	0.993	0.807	0.972	0.990	0.393
M1	0.487	0.802	0.972	0.989	0.401
M2	0.545	0.800	0.972	0.980	0.397
M3	0.429	0.815	0.971	0.980	0.404
M4	0.543	0.830	0.964	0.984	0.409
M5	0.441	0.810	0.962	0.984	0.398
M6	0.544	0.834	0.933	0.969	0.388
M7	0.529	0.790	0.949	0.972	0.099

Table 2.2: Average RR for different clustering methods. RR for AW and IW methods are the same due to the use of indicator function for non-zero weights. The maximum SE of RR in this table is 0.017, while the maximum SE of RR for FRSHC methods is 0.008.

	HC	SHC	DRSHC	FRSHC	FRSHC-Gap	FRSHC-Croux
M0	0.014	0.161	45.940	0.315	163.115	14.038
M1	0.013	0.190	45.811	0.316	164.811	14.118
M2	0.013	0.197	45.382	0.315	164.989	13.885
M3	0.013	0.271	45.980	0.314	165.623	14.182
M4	0.014	0.264	46.886	0.332	167.977	13.927
M5	0.014	0.272	46.730	0.338	170.452	14.036
M6	0.015	0.301	46.017	0.384	134.041	13.950
M7	0.015	0.158	42.166	0.311	119.865	12.880

Table 2.3: Average CPU times in seconds of a 2.8 GHz Intel Xeon for different clustering methods

features. On the other hand, DRSHC, the FRSHC methods, and the FRSHC-Gap methods are only mildly affected by the contamination and give most of the weights to the clustering features.

Table 2.3 presents the average computing times. The computing times for the FRSHC methods are comparable to HC and SHC. For larger n, say 120, the FRSHC methods become faster than SHC. DRSHC is more computationally intensive. The FRSHC-Gap methods give the longest computing time because of the computationally intensive permutation approach used to automatically select the number of clustering features.

2.5 Application to Microarray Data Sets

We consider two breast cancer microarray data sets [24, 35]. For each data set, we apply the following hierarchical clustering methods: HC, SHC, SHC-Gap, FRSHC-AW, FRSHC-IW, FRSHC-Gap-AW, and FRSHC-Gap-IW, where SHC-Gap represents the SHC with the number of clustering features chosen automatically by the permutation approach [see 38]. Dendrograms are created using Ward's linkage so that the results are comparable with those in the original papers.

2.5.1 Breast Cancer Data Set in Perou et al. [24]

The data set was first published in Perou et al. [24] and later analyzed in Witten and Tibshirani [38]. It contains 1753 gene expression levels (features) for 62 samples (observations) to profile surgical specimens of human breast tumors. Perou et al. [24] categorized the 62 samples into four groups (clusters): basal-like, Erb-B2, normal breast-like, and ER+. Perou et al. [24] suggested that the four underlying clusters could be explained by only 496 of the 1753 features, which was confirmed by Witten and Tibshirani [38]. Two misclassified samples were identified by Witten and Tibshirani [38]. The data set was pre-processed before being published. As such, there are no outliers in the data set.

Figure 2.4 shows the dendrograms resulting from HC, SHC, FRSHC-AW, and FRSHC-IW, with the number of clustering features manually fixed at 496 for SHC and the FRSHC methods as suggested. Colors are used to indicate the suggested four tumor groups.

The dendrogram generated from SHC found four main clusters, as expected. Two red samples were misclassified as orange. This is surprising because the red and orange clusters are otherwise very well separated in the SHC dendrogram. The FRSHC methods identify the blue and green clusters, however, the dendrograms further suggest that the red cluster may be further split into two sub-clusters, one of which is close to the orange cluster. This is more consistent with the misclassification of the two red samples.

To assess the performance of the different methods in the presence of outliers, we randomly select four observations, and for each of them, the value of a single random feature is replaced by a value from $N(0, 15)$. Figure 2.5 shows the

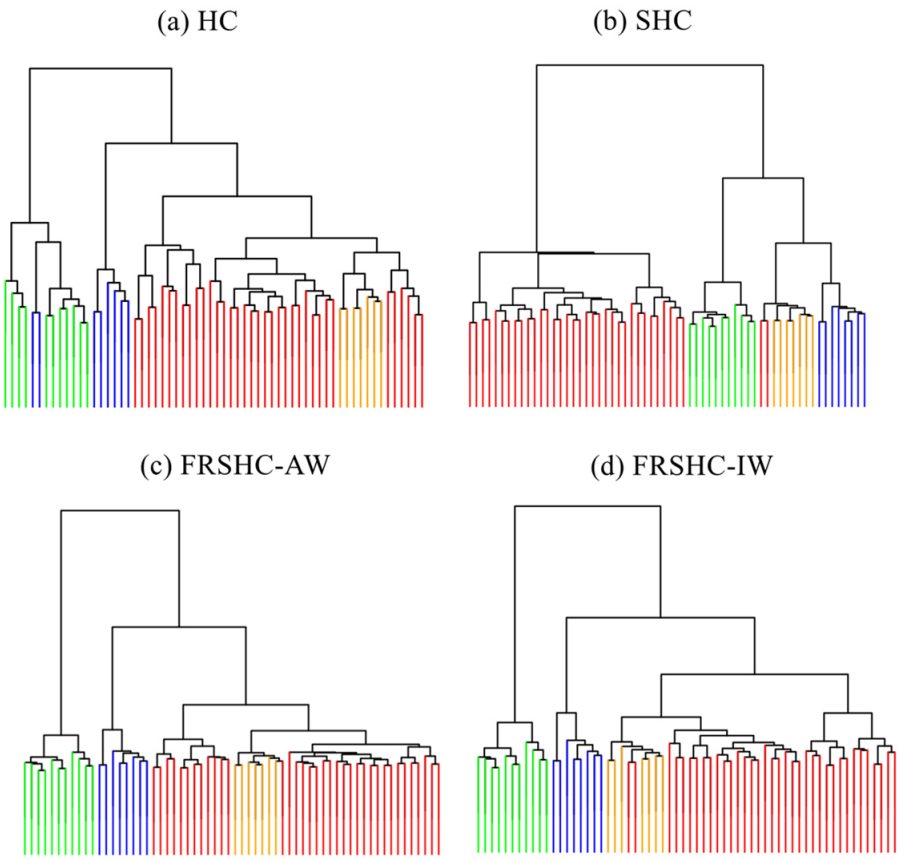

Figure 2.4: Dendrograms generated by HC, SHC, FRSHC-AW and FRSHC-IW with 496 clustering features for the clean Perou et al. [24] data

resulting dendrograms from HC, SHC, FRSHC-AW, and FRSHC-IW applied on the contaminated data set, with the number of clustering features manually fixed at 496 for SHC and the FRSHC methods. HC is mildly affected by the outliers, as expected. On the other hand, SHC produces mixed clusters. The FRSHC methods are robust against the outliers and give very similar clustering patterns and hierarchical structures as in the original data set.

Assuming that the number of clustering features is unknown, we repeat the analysis by applying SHC-Gap and the FRSHC-Gap methods on the original and

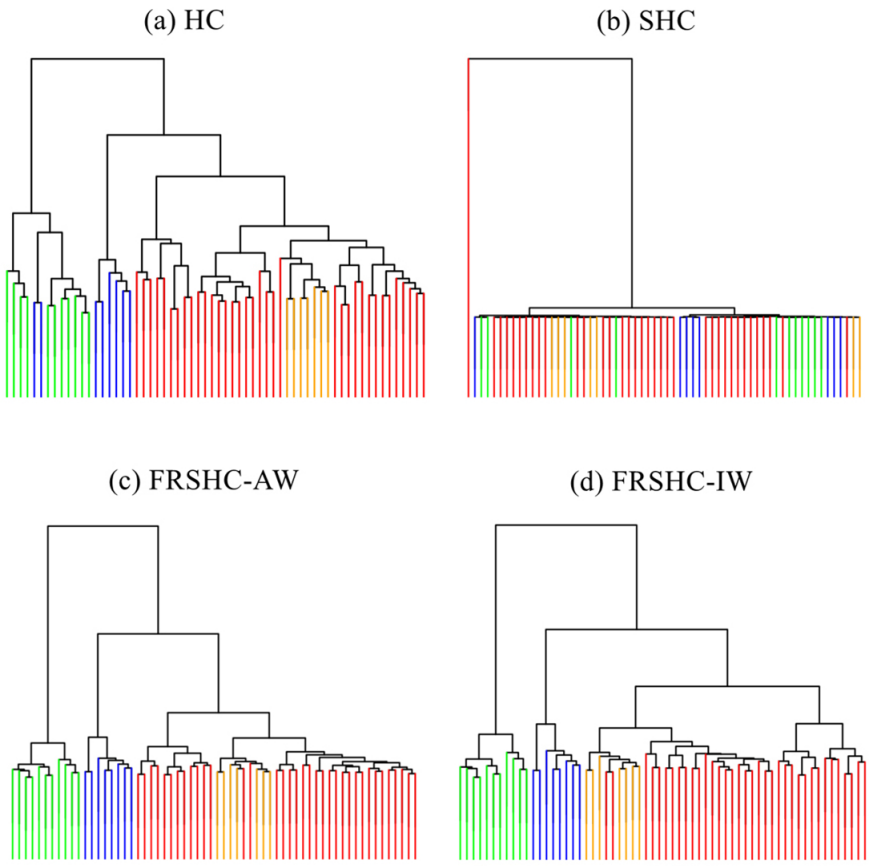

Figure 2.5: Dendrograms generated by HC, SHC, FRSHC-AW and FRSHC-IW with 496 clustering features for the contaminated Perou et al. [24] data

contaminated data sets. Notice that there is no difference in feature selection between FRSHC-Gap-AW and FRSHC-Gap-IW. For the original data set, SHC-Gap selects 112 clustering features, while the FRSHC-Gap methods select 396. For the contaminated data set, SHC-Gap selects 49 clustering features while the FRSHC-Gap methods select 418. The resulting dendrograms from SHC-Gap and the FRSHC-Gap methods give similar clustering patterns and hierarchical structures, compared to those from their manual counterparts, respectively (results not

shown).

2.5.2 Breast Cancer Data Set in van't Veer et al. [35]

The data set was presented and analyzed in van't Veer et al. [35]. It consists of 4751 gene expression levels for 77 primary breast tumor samples. A supervised classification technique was used in van't Veer et al. [35], revealing that only a subset of 70 out of the 4751 genes may help discriminating patients that have developed distant metastasis within five years.

Figure 2.6 shows the resulting dendrograms generated from HC, SHC, FRSHC-AW, and FRSHC-IW, with the number of clustering features fixed at 70, as suggested. The FRSHC methods give slightly more accurate clustering results than HC and SHC, which may be explained by a mild outlier identified from all the dendrograms. When the number of clustering features is assumed to be unknown, SHC-Gap selects 1046 features, while the FRSHC-Gap methods select 111. The dendrograms from SHC-Gap and the FRSHC-Gap methods are similar to their manual counterparts, as in the previous example (result not shown).

2.6 Discussion

Classical hierarchical clustering method is moderately robust against outlying observations, but it has been shown to be non-robust against noise features. On the other hand, sparse hierarchical clustering in Witten and Tibshirani [38] is designed to be robust against noise features. However, we show by simulation studies and real data examples that it is non-robust against outlying observations. Based on these considerations, we propose FRSHC, to achieve robust sparse hierarchical clustering that are robust against outliers and noise features. Moreover, FRSHC scales better than SHC for larger n. We investigate two possible ways of applying weights to the clustering features found in FRSHC, namely, absolute weight (FRSHC-AW) and indicator weights (FRSHC-IW). Simulation studies show that both variants of FRSHC deliver better clustering and feature selection accuracy with less computational time comparing to DRSHC. A permutation approach is proposed to automatically choose the tuning parameter λ, which determines the number of clustering features. Another aspect of our implementation is to allow

the user to choose the number of features to be selected.

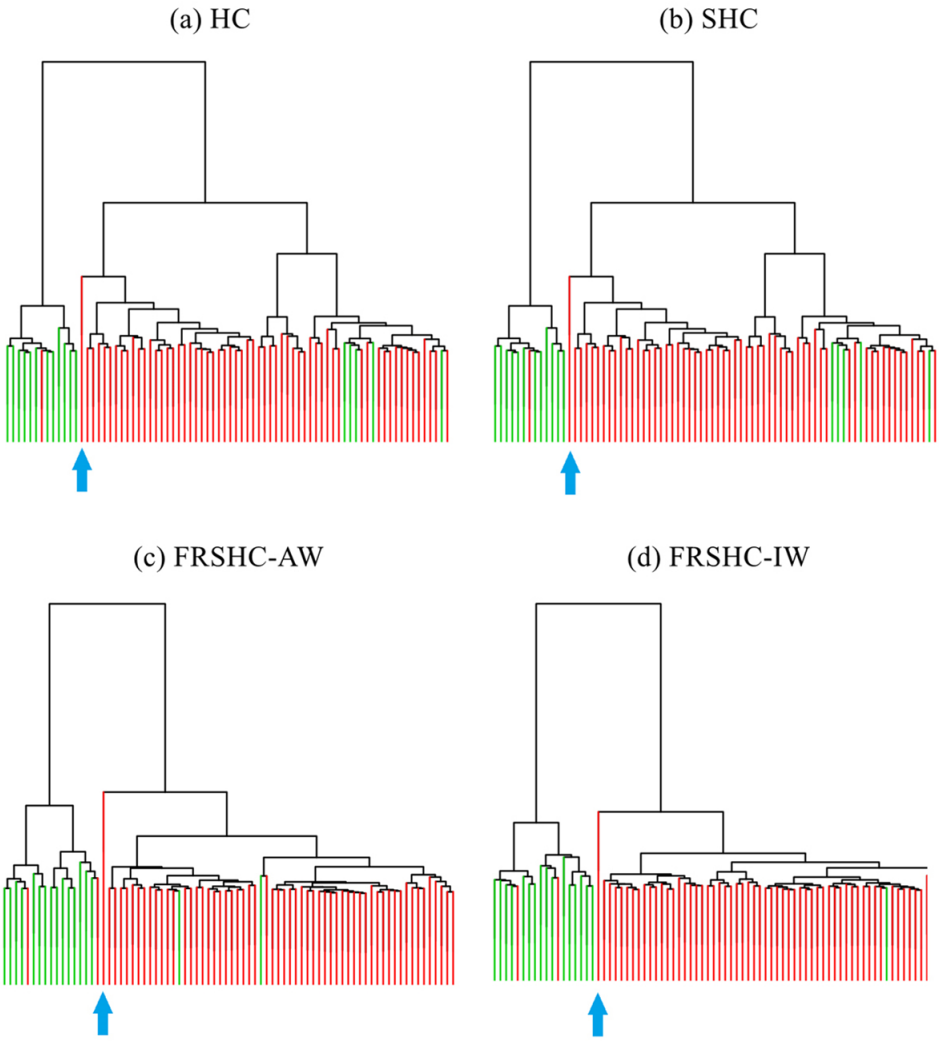

Figure 2.6: Dendrograms generated by HC, SHC, FRSHC-AW and FRSHC-IW with 70 clustering features for van't Veer et al. [35]'s data. The mild outlier is pointed by the arrow.

Chapter 3

Multi-rank Sparse Hierarchical Clustering

3.1 Introduction

In this chapter, we continue the study of sparse hierarchical clustering. In Section 1.2, two main disadvantages of the existing SHC framework are presented. In Chapter 2, we propose a novel robust framework to address the robustness issue. In this chapter, we mainly focus on the other issue mentioned in Section 1.2, namely, incapability of discovering features with complex structures. In high-dimensional data sets, the structures of the features, especially the important features, are usually complex. If the total variance within the important features cannot be mostly projected onto a single dimension, then we call the structure of the important features complex. For example, Figure 3.1 shows examples of simple and complex structures within two important features. We show that when features contain complex structures, SHC has its limitation in discovering the important features from the noise. Then we propose an improved sparse hierarchical clustering framework called the multi-rank sparse hierarchical clustering (MrSHC) framework. Through extensive simulation studies and real data examples, we show that MrSHC is a more flexible framework, and is able to discover the important features more efficiently while producing better clustering results comparing to the SHC framework.

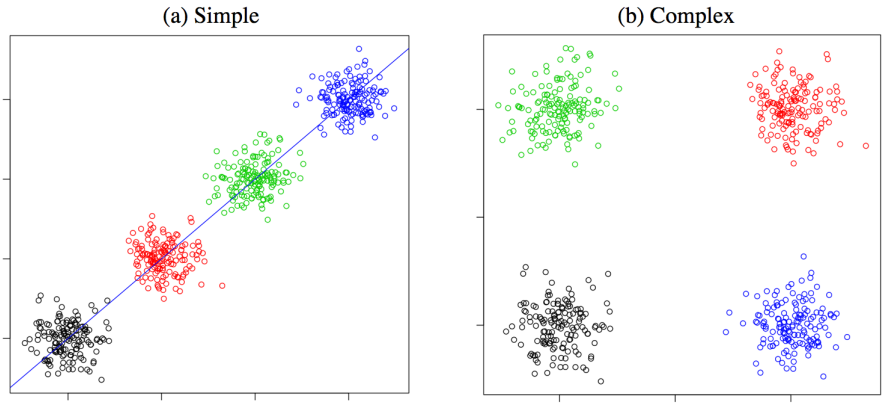

Figure 3.1: Examples of simple and complex structures within two important features. In simple structure, the variance in important features can be mostly projected to a single dimension. The variance of the complex structure can be discovered if projected to more than one dimension.

3.2 Limitations of Witten and Tibshirani [38]'s Sparse Hierarchical Clustering

SHC essentially applies SPC criterion to $\boldsymbol{D}^*$ and obtains the best rank-1 sparse approximation of $\boldsymbol{D}^*$ given a sparsity constraint, i.e. criterion (1.5). The clustering features are chosen according to the non-zero loadings in the first sparse principal component resulted from the rank-1 sparse approximation. This approach and consequently our RSHC approaches share the same limitations when the clustering features may not be fully identified by a single sparse principal component (see Figure 3.1). In other words, the clustering features may not be properly recovered by only rank-1 approximation. The following simulated example illustrates this situation.

We generate a data set $\boldsymbol{X}$ as follows: $\boldsymbol{X}$ contains $n = 20$ observations with $p = 15$ features, i.e. $\boldsymbol{X}_{n\times p} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)^T$, where $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,p})^T$, $1 \leq i \leq n$. The observations are organized in four clusters of size 5. Let Y_i, $(i = 1, \ldots, n)$ denote the cluster memberships. Then x_{ij} $(i = 1, \ldots, n)$ is generated from $N(\mu_j(Y_i), 0.1)$ for $j = 1, \ldots, 4$, and $N(\mu_j(Y_i), 1)$ for $j = 5, \ldots, p$.

A sketch of $\boldsymbol{\mu}_j(Y_i)$ is presented in the table below:

Y_i	$\mu_1(Y_i)$	$\mu_2(Y_i)$	$\mu_3(Y_i)$	$\mu_4(Y_i)$	$\mu_5(Y_i) \ldots \mu_P(Y_i)$
1	1	1	1	1	0 ... 0
2	-1	-1	1	1	0 ... 0
3	-1	-1	-1	-1	0 ... 0
4	1	1	-1	-1	0 ... 0

We apply SHC to $\boldsymbol{X}$. By gradually increasing the sparsity constraint, we obtain the sequence of the first 9 chosen variables $\{V_{13}, V_{11}, V_6, V_1, V_2, V_{14}, V_8, V_4, V_3\}$. The first three chosen features are noise features. As a result, the dendrogram generated from the first four chosen features (which is suggested by Witten and Tibshirani [38]'s auto-selection method) gives mixed clusters (See Figure 3.2). The clustering result is still unsatisfactory even if seven variables are chosen (results not show here). Moreover, five noise features are selected before all the four clustering features are chosen.

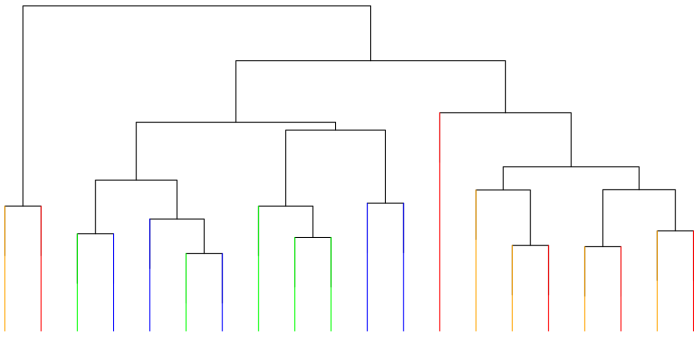

Figure 3.2: Dendrogram generated with the first four chosen features $\{V_{13}, V_{11}, V_6, V_1\}$ from Witten and Tibshirani's sparse hierarchical clustering framework

3.3 Multi-rank Sparse Hierarchical Clustering

To remedy this limitation of SHC, we propose the multi-rank sparse hierarchical clustering (MrSHC). Similar to SHC, MrSHC uses SPC as an important building block for feature selection, but MrSHC is different in the following aspects.

- As motivated by the good performance of FRSHC, MrSHC applies SPC di-

rectly to the original data $\boldsymbol{X}$.

- Indicator weights for the non-zero loadings in sparse PCs are used for a simpler interpretation for the clustering results. Indicator weights also allow MrSHC to match classical hierarchical clustering if no sparsity constraint is applied.

- MrSHC identifies and recovers the clustering features using multi-rank sparse approximation through SPC. In other words, the clustering features are chosen according to the non-zero loadings in multiple sparse PCs.

MrSHC is very different from the traditional approach where high-dimensional data are clustered based on the first few principal components. First, MrSHC applies SPC instead of traditional PCA to the data, also, MrSHC chooses the original features for clustering. The chosen features can be closely approximated by sparse low-rank approximations, in other words, they should have similar patterns of variations (e.g. if the features can be closely approximated by rank-1 approximation, then each of the features should have similar variation as the first PC). In clustering, similar patterns of variations usually represent information of clusters, and thus, the features chosen from MrSHC should contain key informations of clusters. This is confirmed by the simulated and real data examples in later sections.

We outline the MrSHC framework under the cases in Table 3.1.

Case	q	r
1	Known	Known
2	Known	Unknown
3	Unknown	Unknown

Table 3.1: Different cases for MrSHC; q is the target number of selected features and r is the rank of the SPC approximation

.

3.3.1 Case 1: Known q and r

Suppose the target number of clustering features q and the appropriate rank r of the SPC approximation are known. MrSHC applies SPC to $\boldsymbol{X}$ and obtain the first r

sparse PCs. In MrSHC, a feature is considered selected if it has a non-zero loading in any of the r sparse PCs. With an appropriately chosen sparsity constraint λ, we can get q or approximately $q \pm d$ (say $d = 1$) chosen features from the r sparse PCs. MrSHC chooses λ using a bi-section approach presented in Algorithm 3.

Algorithm 3 Feature set selection

1: **Input:** $\boldsymbol{X}_{n\times p}$, q, r, d (default $d = 0$).
2: Assign $\lambda^- = 1$ and $\lambda^+ = \sqrt{n}$ (λ^+ can be set smaller in practice).
3: **Repeat** Step 4-8.
4: Apply SPC to $\boldsymbol{X}$ with $\lambda = (\lambda_k^- + \lambda_k^+)/2$; obtain the first r sparse PCs.
5: $q^* :=$ the number of variables with non-zero loadings in any of the r sparse PCs.
6: $\mathscr{C}_r :=$ the set of q^* chosen variables.
7: **Break** if $q - d \le q^* \le q + d$.
8: If $q^* > q + d$, $\lambda^+ = \lambda^*$; if $q^* < q - d$, $\lambda^- = \lambda^*$.
9: **Output:** $\mathscr{C}_r$, q^*.

We have seen in the simulations and real data examples that Algorithm 3 will finish in a few iterations.

Given q and r, a set of chosen features $\mathscr{C}_r$ can be obtained from Algorithm 3. Then MrSHC simply generates a dendrogram (with any linkage of choice) based on the features in $\mathscr{C}_r$.

3.3.2 Case 2: Known q, Unknown r

Suppose q is known, but not r. To choose r, MrSHC first applies Algorithm 3 with increasing ranks r_i, $i = 1, \ldots, R$ (different R can be chosen; we choose $R = 8$ here). Given rank r_i, let $\mathscr{C}_{r_i}$ denote the candidate feature set obtained from Algorithm 3. Different ranks r_i will be compared through their corresponding $\mathscr{C}_{r_i}$. MrSHC assesses the quality of a feature set $\mathscr{C}_{r_i}$ through the dendrogram generated from its features. To be more specific, a feature set $\mathscr{C}_{r_i}$ is evaluated according to the following aspects.

- The number of well-separated clusters discovered from the dendrogram generated from $\mathscr{C}_{r_i}$. MrSHC uses a multi-layer pruning approach to obtain the well-separated clusters from a dendrogram. We introduce a "reference num-

ber of clusters" in the multi-layer pruning to facilitate later comparisons (described below).

- The degree of separation of the discovered clusters, which is evaluated by silhouette values [27].

Given the number of discovered clusters and the silhouette values, an iterative selection approach is proposed to choose the final rank r.

Multi-layer pruning (MLP) prunes the dendrogram from the top to the bottom, with each split evaluated by the Gap statistics [33]. We introduce a "reference number of clusters" K in MLP, which is both an "upper bound" and a "lower bound". It is an upper bound of the number of clusters discovered in MLP. When K is chosen properly, MLP will produce labels for K clusters for most of the input dendrograms generated from $\mathscr{C}_{r_i}$, $i = 1, \ldots, R$. This facilitates later comparisons since labels with different number of clusters are in general difficult to compare. It is also a lower bound of the number of clusters that are expected to be discovered. If less than K clusters are discovered from a dendrogram according to MLP, such a dendrogram and its corresponding feature set are considered to be of low quality since key clusters may be missing. Therefore, such feature sets and their corresponding ranks are screened out and excluded from the later comparisons. Details of MLP are presented in Algorithm 4. The reference number of clusters K can be chosen based on subject area knowledge. If not specified, we set the default reference number of clusters to be $\max\{2, K_0\}$, where K_0 is set as follows: apply MLP with $K = +\infty$ to the dendrograms generated from $\mathscr{C}_i$, $i = 1, \ldots, R$, then K_0 is the largest output number of leaf nodes from MLP. We have seen that in practice, a reliable K_0 can usually be found by applying MLP to $\mathscr{C}_i$, $i = 1, 2, 3$.

Suppose there are M ($M \leq R$) left over dendrograms after screening out the ones with less than K clusters. Let r_j, $\mathscr{C}_{r_j}$ and $\mathscr{L}_{r_j}$ ($j = 1, \ldots, M$) denote their corresponding ranks, feature sets and labels (for K clusters from MLP), respectively. Given $\mathscr{C}_{r_j}$ and $\mathscr{L}_{r_j}$, the degree of separation of the corresponding K clusters can be evaluated by the average silhouette value S_{r_j}. High average silhouette values indicate well-separated clusters, and thus, are preferred. If two ranks lead to similar average silhouette values, the lower one is preferred since the feature set associated with the higher rank is more likely to contain noise variables. Therefore, among

Algorithm 4 Multi-layer pruning (MLP)

1: **Input:** A dendrogram $\mathscr{D}$, number of bootstrap samples B for the Gap statistics, and reference number of clusters K.
2: Assign the root node of $\mathscr{D}$ as the current node; mark current node as active.
3: **Repeat** Step 4-7.
4: Split the current node sequentially according to $\mathscr{D}$; obtain increasing number of clusters.
5: Evaluate different numbers of clusters from Step 4 using the Gap statistics.
   - If the chosen number of clusters is 1, set the current node as inactive;
   - Otherwise, split the current node into two active leaf nodes.
6: **Break** if either of the following applies:
   - Number of leaf nodes (both active and inactive) is equal to K.
   - All the leaf nodes are inactive.
7: Assign the active leaf node with the highest height in $\mathscr{D}$ as the current node.
8: **Output:** Number of leaf nodes (less than or equal to K), and the corresponding cluster labels $\mathscr{L}$.

local minimums in S_{r_j} $(j = 1,\ldots,M)$, the one with the highest rank is the least favourite, and thus, we remove such local minimums iteratively until the left-over S_{r_j} are monotonically increasing or decreasing. Given monotonically increasing average silhouette values, the smallest rank after the largest increase in average silhouette value will be selected, since smaller ranks are preferred unless the increase in average silhouette value is large. On the other hand, if the average silhouette values are decreasing as rank increases, the smallest left-over rank will be selected. Details of this iterative selection approach are presented in Algorithm 5.

Once the chosen rank r is obtained from Algorithm 5, MrSHC generates a dendrogram (with any linkage of choice) based on the features in $\mathscr{C}_r$.

We revisit the example in Section 3.2. Suppose $q = 4$ is known, and we apply MrSHC with default reference number of clusters $K = 2$. The resulting C_r with its corresponding rank $r = 2$ contains the four true clustering features: V_1, V_2, V_3 and V_4. The resulting dendrogram is presented in Figure 3.3. The four clusters are separated correctly.

Algorithm 5 Iterative selection of rank

1: **Input:** $\boldsymbol{X}_{n\times p}$, $\mathscr{C}_{r_j}$ and $\mathscr{L}_{r_j}$, $j = 1,\ldots,M$.
2: **For** $j = 1,\ldots,M$; S_{r_j} := the average silhouette value calculated from $\mathscr{C}_{r_j}$, $\mathscr{L}_{r_j}$ and $\boldsymbol{X}$.
3: **Repeat** Step 4-5.
4: Among the local minimums in S_{r_j} $(j = 1,\ldots,M)$, remove the one with the highest rank.
5: **Break** if the left-over S_{r_j}, as r_j increases, are monotonically:
   - increasing: r := the smallest rank r_j after the biggest increase in the left-over S_{r_j}.
   - decreasing: r := the smallest left-over rank r_j.
6: **Output:** The chosen rank r.

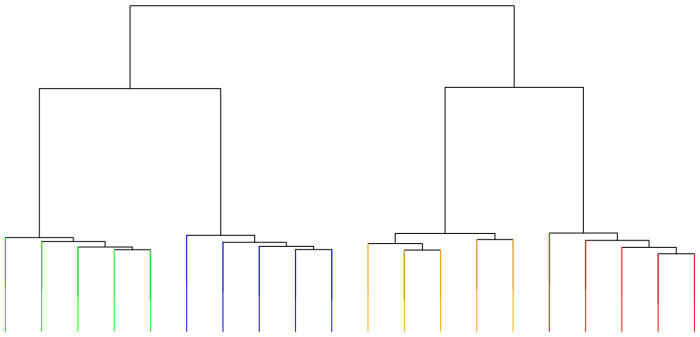

Figure 3.3: Dendrogram generated from MrSHC with known q

3.3.3 Case 3: Unknown q and r

Suppose q and r are both unknown. MrSHC considers a list of target numbers of chosen features $q_t, t = 1,\ldots,T$. For each of the candidate q_t, MrSHC chooses its corresponding feature set $\mathscr{C}_{q_t}$ $(|\mathscr{C}_{q_t}| = q_t)$ and average silhouette value S_{q_t} as described in Section 3.3.2. Higher silhouette values are preferred, and at the mean time, smaller target numbers of features are preferred for the sake of interpretation and exclusion of noise features. Therefore, MrSHC uses a similar iterative approach as in Algorithm 5 to choose q among q_t $(t = 1,\ldots,T)$. Details of this iterative approach are presented in Algorithm 6.

Algorithm 6 Iterative selection of number of features

1: **Input:** q_t and S_{q_t}, $t = 1, \ldots, T$.
2: **Repeat** Step 3-4.
3: Among the local minimums in S_{q_t} $(t = 1, \ldots, T)$, remove the one with the highest q_t.
4: **Break** if the left-over S_{q_t}, as q_t increases, are monotonically:

- increasing: $q :=$ the smallest q_t after the biggest increase in the left-over S_{q_t}.
- decreasing: $q :=$ the smallest left-over rank q_t.

5: **Output:** The chosen target number of chosen features q.

Once the chosen q is obtained from Algorithm 6, MrSHC generates a dendrogram (with any linkage of choice) based on the features in its corresponding $\mathscr{C}_q$.

Again, we revisit the example in Section 3.2. Suppose q and r are unknown, and we apply MrSHC with default reference number of clusters $K = 2$ and the list of q_t $\{2, 3, \ldots, 8\}$. MrSHC suggests $q = 3$ and its corresponding feature set $\{V1, V3, V4\}$. Although $q = 3$ is smaller than the true value 4, the four clusters can still be separated correctly (see Figure 3.4).

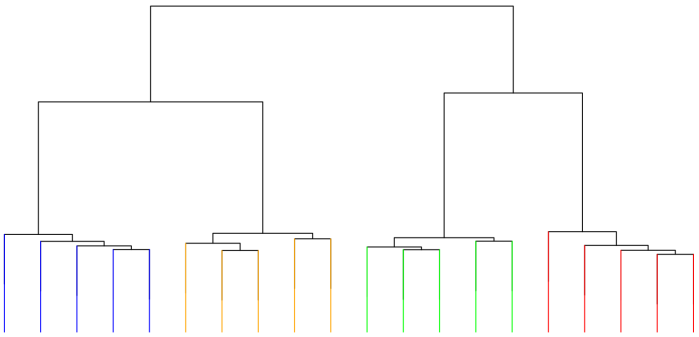

Figure 3.4: Dendrogram generated from MrSHC with unknown q

3.4 Simulation Study

We conduct simulation studies to compare the quality of dendrograms and the accuracy of feature selection of the following methods: HC, SHC (with known/unknown q) and MrSHC (with known/unknown q). We show the results for all the methods with complete linkage. Similar results (not shown here) are obtained with other linkages.

3.4.1 Simulation I

We generate data sets $\mathbf{X}$ with $n = 60$ observations and $p = 500$ features as follows. The observations are generated from three main underlying clusters C1, C2 and C3.

To be more specific, the clusters are determined by $q = 50$ features as follows:

$$X_{ij} = \begin{cases} \mu_i + \varepsilon_{ij} & j = 1, ..., 50 \\ \varepsilon_{ij} & j = 51, ..., 500 \end{cases}$$

where $\varepsilon_{ij} \sim_{i.i.d} N(0,1)$ and

$$\mu_i = \begin{cases} 0 & i = 1, \ldots, 20 \ (i \in C1) \\ \mu & i = 21, \ldots, 40 \ (i \in C2) \\ -\mu & i = 41, \ldots, 60 \ (i \in C3) \end{cases}$$

We show the results for $\mu = 1$. Similar conclusions are obtained for $\mu = 0.8$.

We generate 100 data sets and apply HC, SHC (with known/unknown q) and MrSHC (with known/unknown q) to each.

The quality of the resulting dendrograms is evaluated as follows. The dendrograms are cut at a level to obtain three clusters. CER is then used to assess clustering accuracy by comparing the resulting labels from three clusters against the underlying true labels (C1, C2, C3). The accuracy of feature selection is evaluated by RR.

Table 3.2 presents the average CER, average RR, and the corresponding average q. For SHC and MrSHC, the unknown q is chosen automatically. HC gives the

highest CER, while SHC achieves better accuracy due to the sparseness. MrSHC achieves the best accuracy among the three methods. When $q = 50$ is known, SHC and MrSHC give very similar average RR. When q is unknown, both methods give almost perfect RR, while MrSHC selects less features on average.

	HC		SHC			MrSHC		
	CER	q	CER	RR	q	CER	RR	q
Known ($q = 50$)	0.202	500	0.047	0.926	50	0.009	0.995	50
Unknown q	0.202	500	0.127	0.997	30.6	0.064	1.000	19.7

Table 3.2: Average CER, RR, and q for different clustering methods.

3.4.2 Simulation II

We generate data sets $\mathbf{X}$ with $n = 80$ observations and $p = 500$ features, i.e. $\boldsymbol{X}_{n\times p} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)^T$, where $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,p})^T$, $1 \leq i \leq n$. The observations are generated from four main underlying clusters C1, C2, C3 and C4. Let Y_i, $(i = 1, \ldots, n)$ denote the cluster memberships. Then x_{ij} $(i = 1, \ldots, n)$ is generated from $N(\mu_j(Y_i), 0.1)$ for $j = 1, \ldots, 50$, and $N(\mu_j(Y_i), 1)$ for $j = 51, \ldots, p$.

A sketch of $\boldsymbol{\mu}_j(Y_i)$ is presented in the table below:

Y_i	$\mu_1(Y_i)$	$\ldots$	$\mu_{25}(Y_i)$	$\mu_{26}(Y_i)$	$\ldots$	$\mu_{50}(Y_i)$	$\mu_{51}(Y_i) \ldots \mu_p(Y_i)$
1	μ	$\ldots$	μ	μ	$\ldots$	μ	0 $\ldots$ 0
2	-1.5μ	$\ldots$	-1.5μ	0	$\ldots$	0	0 $\ldots$ 0
3	0	$\ldots$	0	$-\mu$	$\ldots$	$-\mu$	0 $\ldots$ 0
4	0	$\ldots$	0	0	$\ldots$	0	0 $\ldots$ 0

We show the results for $\mu = 1$. Similar conclusions are obtained for other choices of μ.

Again, we generate 100 data sets and apply HC, SHC (with known/unknown q) and MrSHC (with known/unknown q) to each. Table 3.3 presents the average CER, average RR, and the corresponding average q. When $q = 50$ is known, HC and MrSHC produce the highest and lowest average CER, respectively. MrSHC produces more accurate feature selection and clustering than SHC due to the smaller average CER and larger average RR. When q is unknown, SHC produces the highest average CER with on average 19.5 chosen features, while MrSHC achieves the

smallest average CER with on average 31.5 chosen features.

	HC		SHC			MrSHC		
	CER	q	CER	RR	q	CER	RR	q
Known ($q = 50$)	0.172	500	0.129	0.875	50	0.041	0.977	50
Unknown q	0.172	500	0.202	1.000	19.5	0.057	0.992	31.5

Table 3.3: Average CER, RR, and q for different clustering methods.

3.4.3 Computational Times and Complexity

We investigate the computational times of MrSHC and SHC.
The `HierarchicalSparseCluster` function from the R-package `sparcl` is used to conduct SHC. MrSHC is implemented in `R`, where the `SPC` function from the R-package `PMA` is used to conduct the sparse PCA algorithm. We use default input parameters in MrSHC: the number of bootstrap samples $B = 50$, maximum rank $R = 5$, and default selection of the reference number of clusters K. The average computing times for Simulation I & II are presented in Table 3.4.

	HC		SHC		MrSHC	
Simulation	I	II	I	II	I	II
Known ($q = 50$)	0.006	0.012	0.746	1.075	7.754	11.276
Unknown q	0.006	0.012	10.496	15.518	73.254	103.752

Table 3.4: Average computing times (in seconds) for different clustering methods.

When n and p are relatively small, SHC takes less time to compute. However, since the framework of MrSHC is embarrassingly parallel, parallel computing functions such as `mcapply` from the R-package `multicore` can be easily used to speed up the computation. Moreover, when n and p get larger, MrSHC will become less time consuming (observed with $n = 320$ and $p = 2000$, results not shown). This is because the computational complexities of MrSHC and SHC are $O(n^3qB + np)$ and $O(n^3q + n^2p)$, respectively, and as a result, SHC will become more computationally demanding due to the n^2p term as n and p increase and $p >> n$.

3.5 Application to Microarray Data Sets

Three microarray data sets [1, 24, 35] are considered. We apply HC, SHC, MrSHC, and HC using features with the highest marginal variance (HC-HMV) to each data set. Dendrograms are created using the complete linkage. The default reference number of clusters is used in MrSHC, and the default list of candidate q (for both SHC and MrSHC) is $\{20, 40, 60, \ldots, 400\}$.

3.5.1 Lymphoma Data Set in Dettling [12]

The data set is provided by Dettling [12]. It contains 4026 gene expression levels (features) for 62 samples (observations). Three types of most prevalent adult lymphoid malignancies were studied: 42 cases of diffuse large B-cell lymphoma (DLBCL), 9 samples of follicular lymphoma (FL), and 11 observations of B-cell chronic lymphocytic leukemia (CLL). A specialized cDNA microarray was used to measure the gene expression levels. Following the pre-processing steps in Dudoit et al. [13], the data set is pre-processed by first setting a thresholding window $[100, 16000]$ and then excluding genes with $\max / \min \leq 5$ or $(\max - \min) \leq 500$. A logarithmic transformation and standardization are applied. Finally, a simple 5 nearest neighbor algorithm is employed to impute the missing values.

The dendrograms generated from HC, SHC, MrSHC and HC-HMV are shown in Figure 3.5. Colors are used to indicate the three tumor types. HC only misclassifies two red samples, while SHC gives mixed clusters with the automatically chosen $q = 20$. MrSHC chooses $q = 140$ and uses rank $r = 2$, with only two blue samples misclassified into the green cluster (notice that the blue and green clusters are closer to each other). HC-HMV with $q = 140$ mixes the blue and green clusters. Therefore, MrSHC achieves better clustering accuracy with a better chosen $q = 140$ features using rank $r = 2$.

We further investigate the effect of the rank selection by the dendrograms in Figure 3.6. Figure 3.6(a) shows the dendrogram generated from MrSHC with $q = 140$, but $r = 1$. Figure 3.6(b) shows the dendrogram generated from SHC with $q = 140$. Both dendrograms suggest mixed clusters. This confirms that the rank selection can be crucial for better feature selection and clustering accuracy.

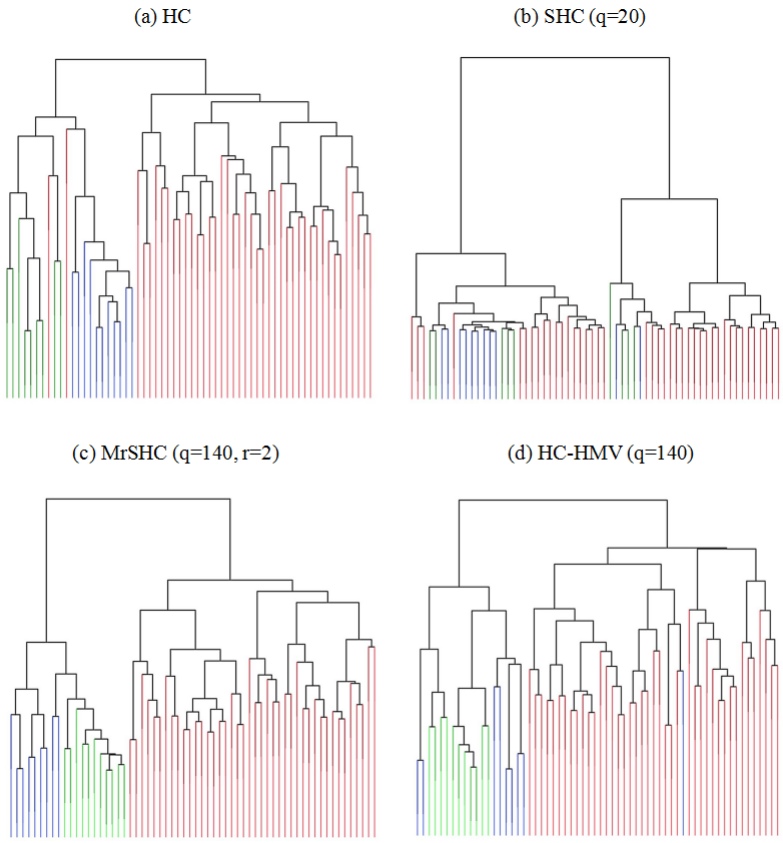

Figure 3.5: Dendrograms generated by HC, SHC, MrSHC and HC-HMV for Dettling [12] data ($n = 62$, $p = 4026$)

3.5.2 Breast Cancer Data Set in Perou et al. [24]

The data set was first published in Perou et al. [24] and later analyzed in Witten and Tibshirani [38]. It contains 1753 gene expression levels (features) for 62 samples (observations) to profile surgical specimens of human breast tumors. Perou et al. [24] categorized the 62 samples into four groups (clusters): basal-like, Erb-B2, normal breast-like, and ER+. Perou et al. [24] suggested that the four underlying clusters could be explained by only 496 of the 1753 features, which was confirmed by Witten and Tibshirani [38]. Two misclassified samples were identified by Witten

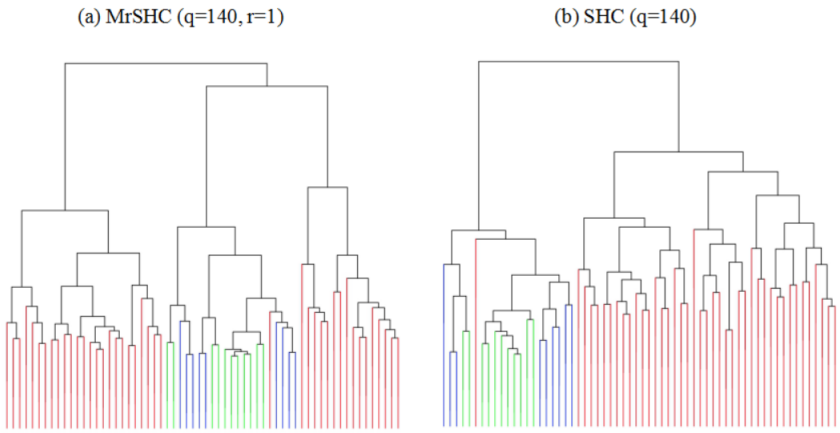

Figure 3.6: Dendrograms generated by MrSHC (rank 1) and SHC with $q = 140$ for Dettling [12] data ($n = 62$, $p = 4026$)

and Tibshirani [38]. The data set was pre-processed before being published. As such, there are no outliers in the data set.

Figure 3.7 shows the dendrograms generated from HC, SHC, MrSHC and HC-HMV. Colors are used to indicate the suggested four tumor groups. HC gives mixed clusters. SHC achieves better clustering – 5 misclassified samples, with the automatically chosen $q = 100$ (similar results – $q = 93$ features were automatically chosen and 5 samples are misclassfied – were obtained in Witten and Tibshirani [38]). MrSHC misclassifies only 2 samples by using the automatically selected $q = 60$ and rank $r = 1$. HC-HMV with $q = 60$ gives mixed clusters. Although MrSHC chooses $r = 1$ over higher ranks, it still provides better feature selection and more accurate clustering comparing to the other three methods.

3.5.3 Breast Cancer Data Set in van't Veer et al. [35]

The data set was presented and analyzed in van't Veer et al. [35]. It consists of 4751 gene expression levels for 77 primary breast tumor samples. A supervised classification technique was used in van't Veer et al. [35], revealing that only a subset of 70 out of the 4751 genes may help discriminating patients that have developed distant metastasis within five years.

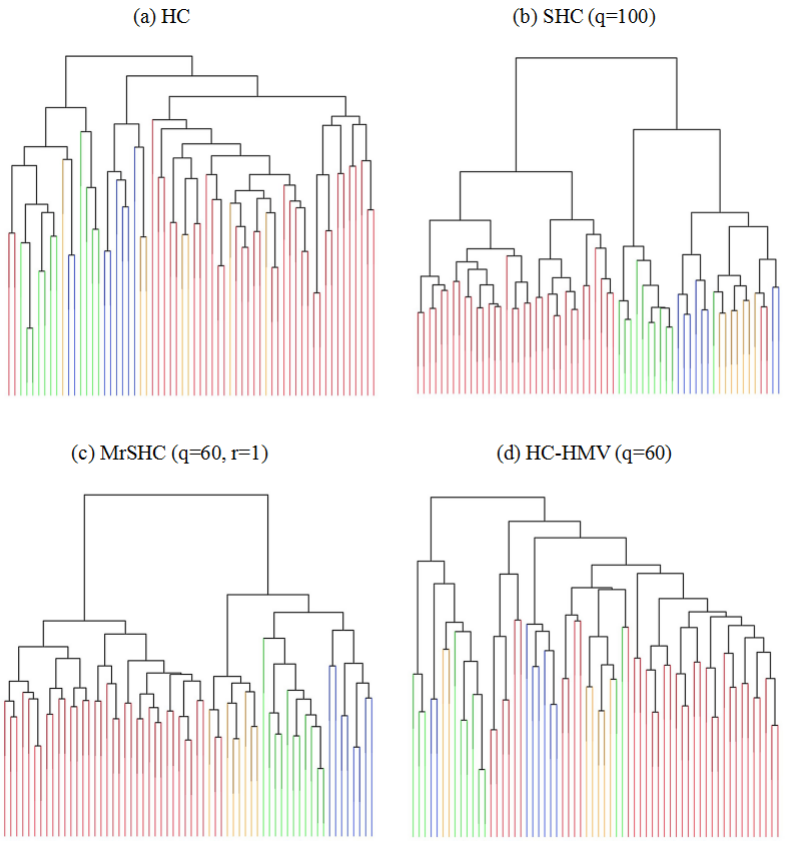

Figure 3.7: Dendrograms generated by HC, SHC, MrSHC and HC-HMV for Perou et al. [24] data ($n = 62$, $p = 1753$)

Figure 3.8 shows the resulting dendrograms generated from HC, SHC, MrSHC and HC-HMV. HC misclassifies 6 samples. SHC achieves slightly better accuracy – 5 misclassified samples, with the automatically chosen $q = 340$. MrSHC with $r = 2$ achieves the same accuracy with less ($q = 100$) features. HC-HMV with $q = 100$ features gives similar accuracy as MrSHC and SHC.

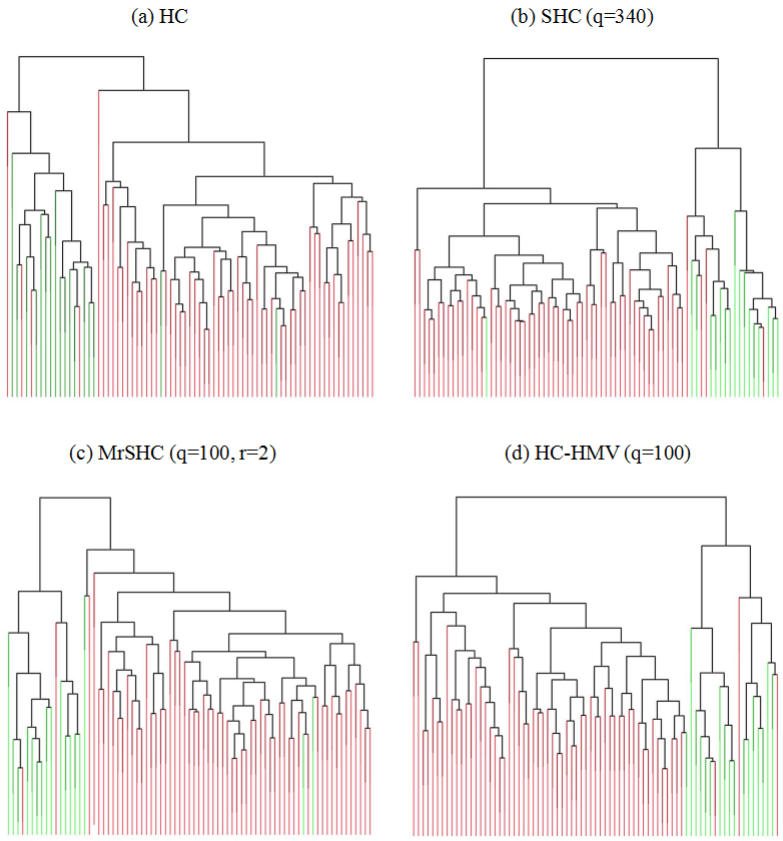

Figure 3.8: Dendrograms generated by HC, SHC, MrSHC and HC-HMV for van't Veer et al. [35]'s data. ($n = 77$, $p = 4751$)

3.6 Robustification of MrSHC

In this section, we focus on improving the robustness of MrSHC. We consider MrSHC to be relatively robust to outliers due to the use of indicator weights for the selected features – the effect of distortions on the feature weights generated from SPC is minimized. However, since SPC is not robust to outliers (see Chapter 2 for more details), we propose to further improve the robustness of MrSHC by using RSPC-τ instead of SPC in Algorithm 3. The robust version of Algorithm 3 is presented in Algorithm 7.

Algorithm 7 Robust feature set selection

1: **Input:** $\boldsymbol{X}_{n\times p}$, q, r, d (default $d = 0$).
2: Assign $\lambda^- = 1$ and $\lambda^+ = \sqrt{n}$ (λ^+ can be set smaller in practice).
3: **Repeat** Step 4-8.
4: Apply RSPC-τ to $\boldsymbol{X}$ with $\lambda = (\lambda_k^- + \lambda_k^+)/2$; obtain the first r robust sparse PCs.
5: $q^* :=$ the number of variables with non-zero loadings in any of the r robust sparse PCs.
6: $\mathscr{C}_r :=$ the set of q^* chosen variables.
7: **Break** if $q - d \le q^* \le q + d$.
8: If $q^* > q + d$, $\lambda^+ = \lambda^*$; if $q^* < q - d$, $\lambda^- = \lambda^*$.
9: **Output:** $\mathscr{C}_r$, q^*.

Replacing Algorithm 3 in place with Algorithm 7 results in a robust version of MrSHC – Robust-MrSHC. To compare the performance of Robust-MrSHC, MrSHC and SHC, we first revisit the simulation studies in Section 3.4.

We first generate data sets as described in Section 3.4.1. To illustrate the robustness of Robust-MrSHC, we add outliers in the simulated data set as follows. For each clusters, randomly select two entries in the noise features and two entries in the clustering features to be replaced by values from $N(0, 15^2)$. The results obtained from 100 simulation runs are presented in Table 3.5. We can see that Robust-MrSHC achieves the smallest CER and better RR than SHC.

	HC		SHC			MrSHC		
	CER	q	CER	RR	q	CER	RR	q
Known ($q = 50$)	0.176	500	0.425	0.299	50	0.021	0.839	50
Unknown q	0.176	500	0.437	0.364	28.1	0.060	0.812	39

Table 3.5: Average CER, RR, and q for different clustering methods.

We then generate data sets as described in Section 3.4.2. Again, for each clusters, we randomly select two entries in the noise features and two entries in the clustering features and replace them with values drawn from $N(0, 15^2)$. The results obtained from 100 simulation runs are presented in Table 3.6. We can see that Robust-MrSHC achieves the smallest CER and better RR than SHC.

Finally, we revisit the microarray examples in Section 3.5 and compare the performance of HC, SHC, MrSHC and Robust-MrSHC.

	HC		SHC			MrSHC		
	CER	q	CER	RR	q	CER	RR	q
Known ($q = 50$)	0.175	500	0.382	0.285	50	0.101	0.737	50
Unknown q	0.175	500	0.385	0.327	41.7	0.123	0.756	44.5

Table 3.6: Average CER, RR, and q for different clustering methods.

The default list of candidate q for SHC, MrSHC and Robust-MrSHC is reduced to $\{25, 50, 75, \ldots, 400\}$ from $\{20, 40, 60, \ldots, 400\}$ to accommodate the increased computational cost of Robust-MrSHC.

3.6.1 Lymphoma Data Set in Dettling [12]

We revisit the lymphoma data set to show the performance of Robust-MrSHC. To assess the robustness and performance of different methods in the presence of outliers, we randomly select five observations and ten features that were not chosen by any of the sparse clustering methods when applied to the clean data. For each chosen observation, the values of the ten noise features are replaced by values generated from N(15,1). Since the contaminations are in the noise features, a robust feature selection procedure will remain unchanged and avoid selecting those noise features, while a non-robust procedure tends to select those noise features due to contamination. We apply HC, SHC, MrSHC and Robust-MrSHC to the contaminated data set. The dendrograms generated from HC, SHC, MrSHC and Robust-MrSHC are shown in Figure 3.9. Colors are used to indicate the three tumor types.

HC only misclassifies two red samples, however, it is affected by outliers in the noise features and groups the outliers into a standalone cluster. SHC is also severely affected by the outliers and gives mixed clusters with the automatically chosen $q = 25$. The chosen 25 features include almost all the contaminated noise features which explains the cluster consisting of the outliers. MrSHC chooses $q = 175$ and uses rank $r = 4$, and it generates a cluster of outliers since some contaminated noise features are chosen by mistake and as a consequence, the green and blue clusters are not clearly separated. On the other hand, Robust-MrSHC chooses $q = 250$ features with rank $r = 4$ and it is not affected by the contaminations (the contaminated noise features are not chosen) while successfully separating the three

underlying clusters. It is clear that Robust-MrSHC is more robust and outperforms HC, SHC and MrSHC in this case.

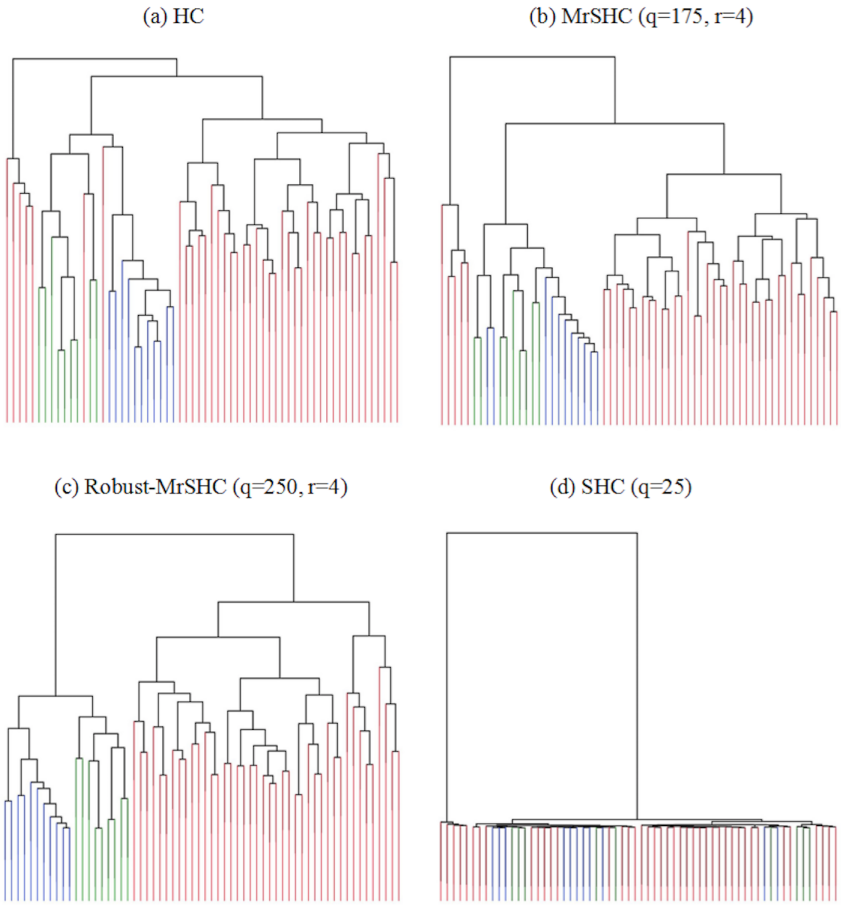

Figure 3.9: Dendrograms generated by HC, MrSHC, Robust-MrSHC and SHC for contaminated Dettling [12] data ($n = 62$, $p = 4026$)

3.6.2 Breast Cancer Data Set in Perou et al. [24]

We revisit the breast cancer data set in Perou et al. [24]. As mentioned in previous sections, this data set was pre-processed before being published. As such, there are no outliers in the data set. Again, we contaminate this data set with a few outliers.

We randomly select five observations and ten features that were not chosen by any of the sparse clustering methods on the clean data. For each chosen observation, the values of the ten noise features are replaced by values generated from N(5,1).

Figure 3.10 shows the dendrograms generated from HC, SHC, MrSHC and Robust-MrSHC. Colors are used to indicate the suggested four tumor groups. HC gives mixed clusters. SHC automatically chooses $q = 50$ features, including some contaminated noise features. Therefore, the outliers form a single cluster and SHC also has trouble separating the green and blue clusters. MrSHC also chooses $q = 50$ with rank $r = 1$. Although it is affected less by the outliers, it still gives mixed clusters in general. Robust-MrSHC automatically chooses $q = 300$ features with rank $r = 1$. It separates the green and blue clusters successfully while group the orange and red clusters closely together. This is coherent with the findings in Section 2.5. The dendrogram generated from SHC on the clean data found four main clusters with two red samples were misclassified as orange. This is counter-intuitive because the red and orange clusters are otherwise very well separated. The results from Robust-MrSHC is consistent with those from the FRSHC methods (see Section 2.5), which suggests that the red cluster may be further split into two subclusters, one of which is close to the orange cluster. In general, Robust-MrSHC is affected the least by the outliers comparing to the other three methods.

3.6.3 Breast Cancer Data Set in van't Veer et al. [35]

As pointed out in Section 2.5, this data set is suspected to contain a mild outlier. Therefore, we directly apply Robust-MrSHC and present its dendrogram in Figure 3.11.

Robust-MrSHC only misclassifies 4 samples with $q = 25$ features and rank $r = 4$. As presented in Section 3.5, the numbers of misclassified samples from HC, SHC and MrSHC are 6, 5 and 5 respectively. Therefore, Robust-MrSHC achieves better clustering accuracy with a smaller number of features on this data set with a mild outlier.

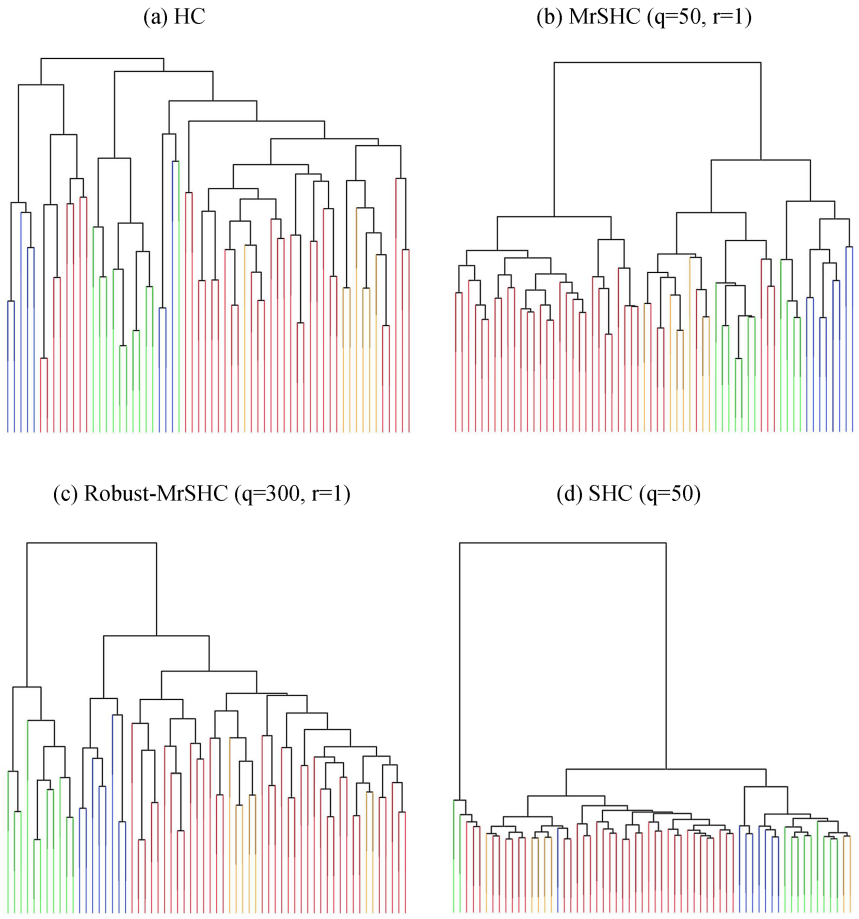

Figure 3.10: Dendrograms generated by HC, MrSHC, Robust-MrSHC and SHC for contaminated Perou et al. [24] data ($n = 62$, $p = 1753$)

3.7 Discussion

In this section, we propose the multi-rank sparse hierarchical clustering (MrSHC), which automatically selects clustering features with higher rank considerations and produces the corresponding sparse hierarchical clustering. As demonstrated in simulation studies and real data examples, MrSHC gives superior feature selection and clustering performance comparing with the classical hierarchical clustering and the

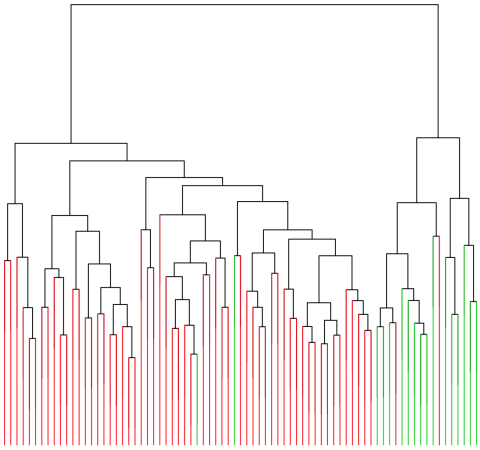

Figure 3.11: Dendrograms generated by Robust-MrSHC for van't Veer et al. [35]'s data. ($n = 77$, $p = 4751$)

sparse hierarchical clustering proposed by Witten and Tibshirani [38]. We also endow MrSHC with the capability of dealing with data contamination and show the robustness and performance of the resulting Robust-MrSHC with several real data examples.

Chapter 4

Regression Phalanxes

4.1 Introduction

Tomal et al. [34] introduced a novel approach for building diverse classification models, for the ensembling of classification models in the context of rare-class detection in two-class classification problems. They proposed an algorithm to divide the often large number of features (or explanatory variables) into subsets adaptively and build a base classifier (e.g. Random Forests) on each subset. The various classification models are then ensembled to produce one model, which ranks the new samples by their probabilities of belonging to the rare class of interest. The essence of the algorithm is to automatically choose the subset groups such that variables in the same group work well together for classification tasks; such groups are called phalanxes.

In this chapter, we propose a different class of phalanxes for application in general regression tasks. We define a "Regression Phalanx" – a subset of features that work well together for regression (or prediction). We then propose a novel algorithm, with hierarchical clustering of features at its core, that automatically builds Regression Phalanxes from high-dimensional data sets and builds a regression model for each phalanx for further ensembling.

In principle, any given regression method can be used as the base regression model. The goal is to improve the prediction accuracy of the base method. In this chapter, we mainly focus on two well-established regression models: Lasso

[32] and Random Forests [6]. These two methods are known to have superior performance in various regression and prediction applications. For each application in this chapter, we first compare the performances of Lasso and Random Forests (RF). The better performing method between the two is then chosen as the base regression model for building Regression Phalanxes.

The idea of ensembling Regression Phalanxes is promising because each Regression Phalanx is relatively low-dimensional. Thus, each variable makes a more significant contribution in the fitted model. Compared to training a full model where variables compete with each other in contributing to the final fit, more useful variables are likely to contribute to the ensembled regression model.

Our proposed phalanx-forming procedure resembles a hierarchical clustering of features (instead of samples), where "similarity" between a pair of features (or subsets of features) is defined by how well they work together in the same regression or prediction model. With properly defined similarity measures, features can be then hierarchically merged into different phalanxes.

The rest of chapter is organized as follows. Section 2 presents the details of our proposed algorithm for building Regression Phalanxes. Section 3 presents a simple illustrative example, which forms the basis for simulation studies in Section 4. In Section 5, we demonstrate the performance of Regression Phalanxes on four additional real data sets. Finally, we conclude with some remarks and discussion of future work.

4.2 Phalanx Formation Algorithm

In this section, the details of the Regression Phalanx formation algorithm are presented. The procedure is an agglomerative approach to build Regression Phalanxes, which is, in essence, a hierarchical clustering of variables. There are four key steps:

1. Initial grouping. Form d initial groups from the original D variables ($d \leq D$).

2. Screening of initial groups. Screen out the underperforming initial groups to obtain $s \leq d$ groups.

3. Hierarchical merging into phalanxes. Hierarchically merge the s screened groups into $e \leq s$ candidate phalanxes.

4. Screening of candidate phalanxes. Screen out the underperforming candidate phalanxes to obtain $h \leq e$ final phalanxes.

A sketch of the procedure is presented in Figure 4.1. Each step of the phalanx-forming procedure is explained in more details in the following sections.

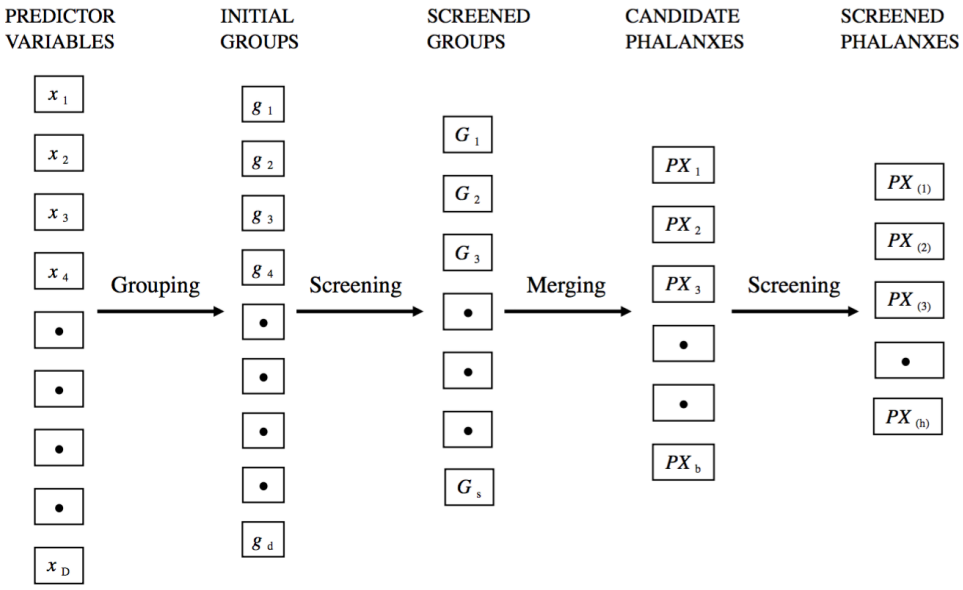

Figure 4.1: A sketch of the phalanx-formation procedure. D variables are partitioned into d initial groups, screened down to s groups, combined into e candidate phalanxes, and then screened down to h phalanxes in the final ensemble ($D \geq d \geq s \geq e \geq h$).

4.2.1 Initial Grouping

This is an optional step. If this step is omitted, each initial group contains a single individual feature and the number of initial groups equals the number of features. As a result, the following steps in the phalanx-formation steps become more computational intensive since the time complexity is quadratic in the number of groups. Thus, it is recommended that the features be grouped into fewer initial groups if they lend themselves to natural grouping (e.g. initial groups can be identified by

features with similar names). Also, if an initial group only contains a binary feature, its corresponding model is likely to be weak since it can only predict two possible values. On the other hand, an initial group with k binary features can produce up to 2^k different predictions. Thus, we recommend the grouping of the binary features into initial groups. If the data set contains a large number of features but no obvious hints for natural grouping, we can still use hierarchical clustering to obtain the initial groups. For binary features, Tomal et al. [34] proposed to use the Jaccard dissimilarity index, defined between binary features $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as

$$d_J(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 - \frac{\boldsymbol{x}_i \cap \boldsymbol{x}_j}{\boldsymbol{x}_i \cup \boldsymbol{x}_j}.$$

Here $\boldsymbol{x}_i \cap \boldsymbol{x}_j$ is the number of observations where the corresponding pair of entries of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ both take the value 1, and $\boldsymbol{x}_i \cup \boldsymbol{x}_j$ is the number of observations where at least one of the corresponding entries of $\boldsymbol{x}_i$ or $\boldsymbol{x}_j$ take the value 1. It is easy to see that $0 \leq d_J(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq 1$. For continuous features (or a mix of binary and continuous features), we propose to use the "1-Abs(Correlation)" dissimilarity measure. That is, the dissimilarity between variable x_i and x_j is calculated as

$$d_C(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 - |\text{corr}(\boldsymbol{x}_i, \boldsymbol{x}_j)|.$$

Notice that the data set needs to be transposed before clustering so that the features are clustered instead of the observations.

This step partitions the original D features into $d \leq D$ initial groups $g_1, g_2, \ldots, g_d$.

4.2.2 Screening of Initial Groups

High-dimensional data are likely to contain noise features which contribute little or even negatively to the prediction task. In such cases, initial groups need to be screened so that noisy initial groups do not participate in the following steps. We first introduce some notation and then we present two tests for the screening of the initial groups.

Notation

We first define some notations to be used in the screening procedure. Denote by c the assessment criterion of a given regression task. Typically c is defined as the mean squared error (MSE) of prediction

$$c = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{4.1}$$

where $\boldsymbol{y} = (y_1, \ldots, y_N)^T$ and $\hat{\boldsymbol{y}} = (\hat{y}_1, \ldots, \hat{y}_N)^T$ are the observed values and their predictions, respectively, of the response at N test points. The data available for the application in Section 4.5.2 allow separate test data, but usually the test points will be generated by cross validation or related methods using the training data.

To assess accuracy based on training data only, different strategies are used for the two candidate regression methods, Lasso and RF.

- Lasso.

  In Lasso, the predictions are produced by K-fold Cross-Validation (we choose $K = 5$ through out the chapter). More specifically, the data set $\boldsymbol{X}$ and the corresponding $\boldsymbol{y}$ are randomly grouped into K folds $(\boldsymbol{X}^{(1)}, \boldsymbol{y}^{(1)})$, …, $(\boldsymbol{X}^{(K)}, \boldsymbol{y}^{(K)})$ with $n_{(1)}, \ldots, n_{(K)}$ observations respectively ($\sum_{i=1}^{K} n_{(i)} = n$). Then the predictions for $\boldsymbol{y}^{(i)}$, namely $\hat{\boldsymbol{y}}^{(i)} = (\hat{y}_1^{(i)}, \ldots, \hat{y}_{n_{(i)}}^{(i)})^T$ is obtained by

  $$\hat{y}_j^{(i)} = \hat{\boldsymbol{f}}^{(-i)}(\boldsymbol{x}_j^{(i)}), \quad j = 1, \ldots, n_{(i)}$$

  where $\hat{\boldsymbol{f}}^{(-i)}(\cdot)$ is the Lasso model fit from the $(K-1)$ folds other than the i-th fold, and $\boldsymbol{x}_j^{(i)}$ is the corresponding feature vector of $y_j^{(i)}$. We denote the assessment criterion c for Lasso as the Cross-Validation MSE (CV-MSE).

- Random Forests (RF).

  In RF, $\hat{\boldsymbol{y}}$ can be obtained from the out-of-bag (OOB) predictions. The prediction $\hat{y}_i^{OOB}$ is obtained from only the trained trees that do not have y_i in their bootstrapped sample, hence, the prediction is called the out-of-bag prediction. We denote the assessment criterion c for RF as the out-of-bag MSE (OOB-MSE).

We further denote $\hat{\mathbf{y}}(g_i)$ as the vector of predictions from the base regression model (e.g. Lasso or RF) using only the variables in g_i. Let $c_i = c(\hat{\mathbf{y}}(g_i))$ denote the assessment measure. Denote by $\hat{\mathbf{y}}(g_i \cup g_j)$ the predictions when the model is fit using the variables in g_i and g_j $(i \neq j)$, and denote

$$c_{ij} = c(\hat{\mathbf{y}}(g_i \cup g_j)) \tag{4.2}$$

as the resulting performance. Similarly, let

$$\bar{c}_{ij} = c((\hat{\mathbf{y}}(g_i) + \hat{\mathbf{y}}(g_j))/2) \tag{4.3}$$

be the performance of the ensemble of the predictions from g_i and g_j.

Tests for Screening Initial Groups

A group survives the screening if it passes the two tests described as follows. A group g_i passes the first test if its own performance is "strong", i.e. high prediction accuracy. A group survives the second test if it produces "significant combining improvement", i.e., after combining with another group g_j, the model trained on the combined variables (from g_i and g_j) produces significantly better accuracy comparing to that from g_j.

We use a null permutation distribution to establish the baseline for strong individual performance and significant combining improvement. Denote $\tilde{\mathbf{y}}$ as the vector of permuted response values in which the original vector of response variable values $\mathbf{y}$ is randomly permuted relative to the features. Then the counterparts of c_i and c_{ij} are calculated with $\tilde{\mathbf{y}}$ as the response and denoted as $\tilde{c}_i$ and $\tilde{c}_{ij}$ respectively. Denote the α quantile of $\tilde{c}_i$ $(i = 1, \ldots, d)$ as $\tilde{p}_\alpha$ and the $1 - \alpha/(d-1)$ quantile of $\tilde{c}_i - \tilde{c}_{ij}$ $(i = 1, \ldots, d; j = 1, \ldots, d)$ by $\tilde{q}_{1-\alpha/(d-1)}$. Then a group g_i survives the screening if it passes both the following two tests:

1. g_i is strong itself:

$$c_i \leq \tilde{p}_\alpha. \tag{4.4}$$

   The rationale is that the strength of an individual initial group should be competitive with the α quantile of the strengths of initial groups with a randomly

permuted response.

2. g_i improves the strength of any other group g_j when g_i and g_j are combined to build a regressor:

$$\tilde{q}_{1-\alpha/(d-1)} \leq c_j - c_{ij} \tag{4.5}$$

The rationale is that the improvement from combining g_i and g_j should be competitive with the $1-\alpha/(d-1)$ quantile of combining improvements of initial groups with a randomly permuted response. The quantile is $1-\alpha/(d-1)$ to adjust for the $(d-1)$ tests for each initial group.

After the screening of initial groups, a list of surviving groups is relabeled as $\{G_1, G_2, \ldots, G_s\}$ for the next step.

4.2.3 Hierarchical Formation of Candidate Phalanxes

We use simple greedy hierarchical clustering techniques to merge Groups $\{G_1, G_2, \ldots, G_s\}$ into phalanxes, which proves to be effective in all of our applications. Each iteration merges the pair of groups G_i and G_j that minimizes

$$m_{ij} = c_{ij}/\bar{c}_{ij}. \tag{4.6}$$

Here $m_{ij} < 1$ indicates that combining G_i and G_j to build a single model provides more strength than ensembling two models built separately on G_i and G_j. The number of groups, s, decreases by 1 after each merge. The merging process stops when $m_{ij} \geq 1$ for all i, j, indicating that further merging damages the performance and the resulting groups, i.e. e candidate phalanxes $PX_1, PX_2, \ldots, PX_e$, should be now considered for ensembling.

4.2.4 Screening of Candidate Phalanxes

Searching for the best subset of candidate phalanxes for further ensembling is a combinatorial problem. In order to reduce the computational cost, we establish the search path P in a forward selection fashion. We initialize P with the candidate phalanx with the smallest value of the criterion c. At each stage all the remaining candidate phalanxes will be considered for ensembling with phalanxes in P one at

a time, and the one with the best ensembling performance with P will be added to P. The ensembling performance $\bar{c}_{(1,2,\ldots,h)}$ is calculated as follows. The predictions of y_i from $PX_1,\ldots,PX_h$ are denoted by $\hat{y}_{i;(PX_1)},\ldots,\hat{y}_{i;(PX_h)}$, and the ensembled prediction for y_i is calculated as

$$\hat{y}_{i;(1,\ldots,h)} = \sum_{j=1}^{h} \hat{y}_{i;(PX_j)}/h \tag{4.7}$$

The ensemble's performance is then calculated from the ensembled predictions.

For ease of description, we assume the order of entry to be $PX_1, PX_2, \ldots, PX_k$, and the corresponding ensembling performance as each candidate is added to be $c_{PX_{\{1\}}}, c_{PX_{\{1,2\}}}, \ldots, c_{PX_{\{1,\ldots,k\}}}$ respectively. Then the set of candidate phalanxes corresponding to the best ensembling performance, say $c_{PX_{1,\ldots,h}}$, will be selected, and the remaining candidates are screened out.

After the screening of weak phalanxes, the surviving h phalanxes are the final phalanxes, and they will be ensembled in the last step.

4.2.5 Ensembling of Regression Phalanxes

We fit a model for each of the h phalanxes of variables, and obtain predictions from them. (For RF, we can increase the number of trees and get better OOB predictions as final predictions.) For a test point, the h predictions from the ensemble of regression phalanxes (ERPX) are averaged to give the final prediction.

4.3 A Simple Illustrative Example: Octane

The octane data set [15] consists of NIR absorbance spectra over 226 wavelengths ranging from 1102 nm to 1552 nm with measurements every two nanometers. For each of the 39 production gasoline samples the octane number was measured.

It is known that the octane data set contains six outliers (cases 25, 26, 36–39) to which alcohol was added. We omit those outliers to obtain 33 clean samples.

We apply Lasso and RF separately on the data set, then we choose Lasso as the base regression model since it produces better results: the MSE of Lasso is about 0.085 compared with about 0.27 for RF.

The R package "glmnet" [16] is used for fitting Lasso models and the penalty

parameter λ is chosen by using "cv.glmnet" with method "lambda.1se". Since cross validation is used for choosing the tuning parameter λ for Lasso, the Phalanx formation procedure has inherited randomness. Therefore, we apply both ERPX and the original Lasso to the Octane data set three times each. For ERPX, we skip the initial grouping step since the number of features is relatively small and the features are all continuous. Different numbers of surviving groups and final phalanxes are obtained. The accuracies of both methods are assessed by CV-MSE. For each run, results for mean CV-MSE over 20 CV repetitions are presented in Table 4.1.

		Number of					
		Groups		Phalanxes		CV-MSE	
Run	Variables	Initial	Screened	Candidate	Screened	ERPX	Lasso
1	226	226	190	7	2	0.051	0.084
2	226	226	195	8	5	0.049	0.086
3	226	226	192	9	2	0.044	0.083

Table 4.1: Number of variables, initial groups, screened groups, candidate phalanxes, screened phalanxes and prediction accuracies for the octane data set.

We can see that the CV-MSE values from ERPX are much smaller than those obtained from the original Lasso models, which confirms that ERPX can boost the performance of the base regression model.

In the following section, we generate synthetic data sets based on the octane data. We simulate data favoring Lasso and RF respectively as the base regression model and show that we are able to improve the performance over the base regression model using the proposed ERPX.

4.4 Simulation Studies

In this section, we present several numerical experiments to demonstrate the performance of the proposed ERPX. We simulate data by first emulating the feature structure of the octane data. Then we generate response data as a function of the features in two different ways, to represent linear and nonlinear relationships with the features, respectively.

The data sets $\boldsymbol{X}_{(n\times p)}$ are generated based on the octane data as follows.

- $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p)$, $\boldsymbol{x}_j = (x_{1j}, x_{2j}, \ldots, x_{nj})^T (j = 1, \ldots, p)$, where $n = 33, p = 226$ as in the octane data set.
- $\boldsymbol{X}$ is sampled from multivariate normal distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ consists of p column means of the octane data.
- The covariance matrix $\boldsymbol{\Sigma}$ is equal to the sample covariance matrix as observed or from a perturbation of the data. Specifically, we add different levels of noise into the octane data, and for each noise level we take the resulting sample covariance matrix for $\boldsymbol{\Sigma}$. We consider three noise levels:
  - No noise: $\boldsymbol{\Sigma}$ is the sample covariance matrix of the octane data.
  - Medium noise: Random samples from $N(0, 3\sigma)$ are added to each element of the octane data, where σ is the minimum feature standard deviation among all the 226 features of the octane data. $\boldsymbol{\Sigma}$ is then the sample covariance matrix of the modified octane data.
  - High noise: Random samples from $N(0, 5\sigma)$ are added to each element of the octane data. $\boldsymbol{\Sigma}$ is then the sample covariance matrix of the modified octane data.

For each noise level we simulate the feature set $\boldsymbol{X}$ 100 times. The strategy for generating $\boldsymbol{y}$ depends on the choice of the base regression model.

4.4.1 Lasso as Base Regression Model

To favor Lasso, for each of the 100 simulated $\boldsymbol{X}$, we generate the response variable $\boldsymbol{y}$ in a linear pattern as follows.

- Randomly select 10 features $\boldsymbol{x}_{k_1}, \ldots, \boldsymbol{x}_{k_{10}}$ from the columns of $\boldsymbol{X}$.
- For each feature $\boldsymbol{x}_{k_j}$ $(j = 1, \ldots, 10)$, generate the corresponding coefficient β_j from a Uniform distribution $U(0, 1)$.
- Generate $\boldsymbol{y}_{init}$ as $\boldsymbol{y}_{init} = \sum_{j=1}^{10} \beta_j \boldsymbol{x}_{k_j}$.

- Generate $\boldsymbol{y}$ as $\boldsymbol{y} = \min(\boldsymbol{y}_{oct}) + a \cdot \boldsymbol{y}_{init} + \boldsymbol{\varepsilon}$, where a is a scale constant $(\max(\boldsymbol{y}_{oct}) - \min(\boldsymbol{y}_{oct}))/(\max(\boldsymbol{y}_{init}) - \min(\boldsymbol{y}_{init}))$, $\boldsymbol{y}_{oct}$ is the original response variable from the octane data set, and $\boldsymbol{\varepsilon}$ contains noises generated from $N(0,1)$.

For each set of simulated data $(\boldsymbol{X}, \boldsymbol{y})$, we apply ERPX with base regression models Lasso and RF respectively. Since the response variable is generated as a linear combination of the predictors, Lasso is expected to perform at least as good as RF. The performance is measured by 5-fold CV-MSE and OOB-MSE for Lasso and RF respectively.

		Average number of					
			Groups			MSE	
Base	Noise	Variables	Initial	Screened	Phalanxes	ERPX	Base
Lasso	No	226	226	172.32	6.01	0.96	1.43
	Medium	226	226	138.32	3.56	0.95	1.58
	High	226	226	88.75	2.78	1.07	1.71
RF	No	226	226	99.58	2.59	0.98	1.32
	Medium	226	226	62.93	2.67	0.96	1.40
	High	226	226	33.35	2.65	1.02	1.56

Table 4.2: Number of variables, initial groups, screened groups, candidate phalanxes, screened phalanxes and prediction accuracies of base regression models and ERPX for different noise levels when calculating sample covariance matrix $\boldsymbol{\Sigma}$.

We can see from Table 4.2, for all the simulation settings, regardless of the choice of the base regression model, ERPX produces more accurate predictions than the corresponding base regression model with large relative margins. Somewhat surprisingly, RF slightly outperforms Lasso in around 70% of the 300 simulated data sets (and also on average). This could be caused by the use of different metrics, OOB-MSE versus CV-MSE, when the sample size is small.

4.4.2 Random Forests as Base Regression Model

To generate the response variable $\boldsymbol{y}$ from highly nonlinear relationships favoring RF, we deploy the following strategy.

- Randomly select 10 features $\boldsymbol{x}_{k_1}, \ldots, \boldsymbol{x}_{k_{10}}$ from the columns of $\boldsymbol{X}$.

- Generate two sets of coefficients $\boldsymbol{\beta}_1 = (\beta_{11}, \ldots, \beta_{1,10})^T$ and $\boldsymbol{\beta}_2 = (\beta_{21}, \ldots, \beta_{2,10})^T$ from a Uniform distribution $U(0,1)$.

- Generate

$$\boldsymbol{y}_i^{init} = \begin{cases} \sum_{j=1}^{10} \beta_{1j} \boldsymbol{x}_{k_j} & x_{ik_1} < \text{median}(\boldsymbol{x}_{k_1}) \\ \sum_{j=1}^{10} \beta_{2j} \boldsymbol{x}_{k_j} & x_{ik_1} \geq \text{median}(\boldsymbol{x}_{k_1}) \end{cases}$$

  Since the initial values in $\boldsymbol{y}^{init}$ are generated from a mixture of two linear patterns, they cannot be easily modelled by linear methods such as Lasso.

- Generate $\boldsymbol{y} = \min(\boldsymbol{y}^{oct}) + a \cdot \boldsymbol{y}^{init} + \boldsymbol{\varepsilon}$, where $a = (\max(\boldsymbol{y}^{oct}) - \min(\boldsymbol{y}^{oct}))/(\max(\boldsymbol{y}^{init}) - \min(\boldsymbol{y}^{init}))$, a scale constant, and $\boldsymbol{\varepsilon}$ contains noises generated from $N(0,1)$.

For each set of simulated data $(\boldsymbol{X}, \boldsymbol{y})$, we again apply ERPX with either Lasso or RF as the base regression model. In this case, we expect RF to outperform Lasso. We can see from Table 4.2, for all the simulation settings, RF outperforms Lasso as anticipated given the simulation set up, and ERPX improves upon both base regression models by large relative margins.

		Average number of					
			Groups			MSE	
Base	Noise	Variables	Initial	Screened	Phalanxes	ERPX	Base
Lasso	No	226	226	150.10	5.29	2.75	3.83
	Medium	226	226	92.79	2.65	2.67	3.80
	High	226	226	48.83	2.48	2.52	3.71
RF	No	226	226	72.83	1.90	1.59	2.15
	Medium	226	226	37.52	2.04	1.66	2.49
	High	226	226	16.43	1.86	1.57	2.51

Table 4.3: Number of variables, initial groups, screened groups, candidate phalanxes, screened phalanxes and prediction accuracies of base regression models and ERPX for different noise levels when calculating sample covariance matrix $\boldsymbol{\Sigma}$.

4.5 Additional Real Data Examples

The prediction performance of ERPX is illustrated on the following data sets.

4.5.1 AID364 Data Set

Assay AID364 is a cytotoxicity assay conducted by the Scripps Research Institute Molecular Screening Center. There are 3,286 compounds used in our study, with their inhibition percentages recorded. Visit http://pubchem.ncbi.nlm.nih.gov/assay/assay.cgi?aid=364 for details. Because toxic reactions can occur in many different ways, this assay is expected to present modelling challenges. We consider five sets of descriptors for the assay, to make 5 data sets. The descriptor sets are the following: atom pairs (AP), with 380 variables; Burden numbers [BN, 8], with 24 variables; Carhart atom pairs [CAP, 9], with 1585 variables; fragment pairs (FP), with 580 variables; and pharmacophores fingerprints (PH), with 120 variables. The Burden numbers are continuous descriptors, and the other four are bit strings where each bit is set to "1" when a certain feature is present and "0" when it is not. See Liu et al. [21] and Hughes-Oliver et al. [18] for further explanation of the molecular properties captured by the descriptor sets.

The initial groups for the descriptor sets are determined by their features names. For example, related features for FP present similar names such as AR_01_AR, AR_02_AR, ..., AR_07_AR, and such features will form the initial groups. We perform our proposed ERPX on each of the five descriptor sets. The base regression model is chosen to be RF due to its superior performance over Lasso in this case (see Table A.1 in Appendix).

ERPX and the original RF are run on each of the five descriptor sets as well as the five descriptor sets combined as a whole set, three times each. The results are presented in Table 4.4. As we can see, ERPX provides superior prediction accuracy over the original RF, and the margin gets bigger with all five descriptor sets merged together. This is because ERPX can exploit the "richness" of features to improve prediction accuracy.

Due to the naming schema of the features, the descriptor sets lend themselves well to obtaining initial groups. However, we show that ERPX still produces comparable prediction accuracies with initial groups obtained from the hierarchical clustering approach described in Section 2.1. We choose the same number of initial groups as shown in Table 4.4 to facilitate comparison. Table 4.5 presents the new results for descriptor sets other than BN (since no initial grouping is needed).

Set	Run	Number of Variables	Groups Initial	Groups Screened	Phalanxes Candidate	Phalanxes Screened	OOB MSE ERPX	OOB MSE RF
BN	1	24	24	15	4	4	122.41	126.70
	2	24	24	14	5	5	123.81	126.97
	3	24	24	15	5	5	121.64	127.33
PH	1	120	21	15	3	2	131.59	135.70
	2	120	21	16	2	2	131.50	134.62
	3	120	21	17	2	2	131.67	135.57
FP	1	580	105	64	9	2	120.03	125.53
	2	580	105	52	9	3	120.26	126.15
	3	580	105	44	6	2	121.80	127.09
AP	1	380	78	35	4	3	124.40	132.07
	2	380	78	33	4	2	124.31	131.73
	3	380	78	31	5	3	124.92	131.69
CAP	1	1585	666	93	11	2	116.04	131.12
	2	1585	666	73	9	2	118.86	130.74
	3	1585	666	95	10	3	116.40	131.25
ALL5	1	2689	895	159	21	6	112.51	125.87
	2	2689	895	208	20	5	113.69	124.78
	3	2689	895	204	18	4	112.80	125.13

Table 4.4: Number of variables, initial groups, screened groups, candidate phalanxes, screened phalanxes and prediction accuracies for the AID 364 assay and five descriptor sets. Three runs of ERPX are presented.

As we can see, the prediction accuracies are comparable to those in Table 4.4 where the initial grouping is done according to feature names.

4.5.2 CLM Data Set

The community land model with carbon/nitrogen biogeochemistry (CLM-CN) is a state-of-the-art land surface model to make future climate projections. Sargsyan et al. [28] performed simulations using CLM-CN for a single plant functional type: temperate evergreen needleleaf forest. The outputs were 100-year-later projected values of several quantities of interest (QoIs). The simulator was run 10,000 times for different settings of 79 input parameters, out of which 9983 runs succeeded.

We make predictions for two QoIs, leaf area index (LAI) and total vegetation carbon (TOTVEGC), from the 79 input variables. The two QoIs are log scaled since their sample distributions are highly right-skewed. The 9983 runs contain 38% and 0.07% of zeros, respectively. Therefore, we add constants 10^{-10} and 10^{-13} to the two QoIs respectively before we apply the log scaling (these values are roughly equal to the minima of the respective non-zero values). Since the cli-

Set	Run	Number of Variables	Groups Initial	Groups Screened	Phalanxes Candidate	Phalanxes Screened	OOB MSE ERPX	OOB MSE RF
PH	1	120	21	14	3	2	132.06	135.01
	2	120	21	15	3	2	130.68	134.21
	3	120	21	15	3	2	131.47	134.78
FP	1	580	105	47	6	3	120.85	125.73
	2	580	105	44	9	2	122.85	125.53
	3	580	105	44	10	3	123.87	126.48
AP	1	380	78	20	4	2	125.23	132.35
	2	380	78	27	5	2	124.02	131.27
	3	380	78	20	5	2	123.95	132.77
CAP	1	1585	666	57	7	3	120.76	129.82
	2	1585	666	74	10	3	118.26	131.84
	3	1585	666	58	7	5	121.35	131.16
ALL5	1	2689	895	143	12	4	114.15	125.00
	2	2689	895	89	11	6	115.46	123.67
	3	2689	895	74	12	5	116.44	125.35

Table 4.5: Number of variables, initial groups, screened groups, candidate phalanxes, screened phalanxes and prediction accuracies for the AID 364 assay and four descriptor sets, with initial groups obtained from the hierarchical clustering. Three runs of ERPX are presented.

mate projections are affected by a number of uncertainties in CLM-CN, predicting the LAI and TOTVEGC is a challenging task. We choose RF as the base regression model due to its superior performance versus Lasso in this example (see Table A.2 in Appendix). We apply both ERPX and the original RF to demonstrate their performance. For ERPX, we skip the initial grouping step since there are only 79 variables.

The 9983 observations are randomly split into training and testing sets with 5000 and 4983 observations respectively. We repeat the random splitting 20 times to obtain 20 different pairs of training and testing sets. For each random split, both ERPX and RF are trained on the randomly sampled training set and applied to the corresponding test set. Therefore, we can generate boxplots of both the 20 OOB-MSEs from the training sets and the 20 test MSEs from the testing sets. The boxplots are shown in Figure 4.2. It is clear that ERPX provides more accurate predictions than RF, for both TOTVECG and LAI as response variables.

We also present the detailed results from the first three splits in Table 4.6. Since the number of phalanxes is always one in all the runs, the mainly difference between ERPX and RF is the screening of initial groups. By screening out most of

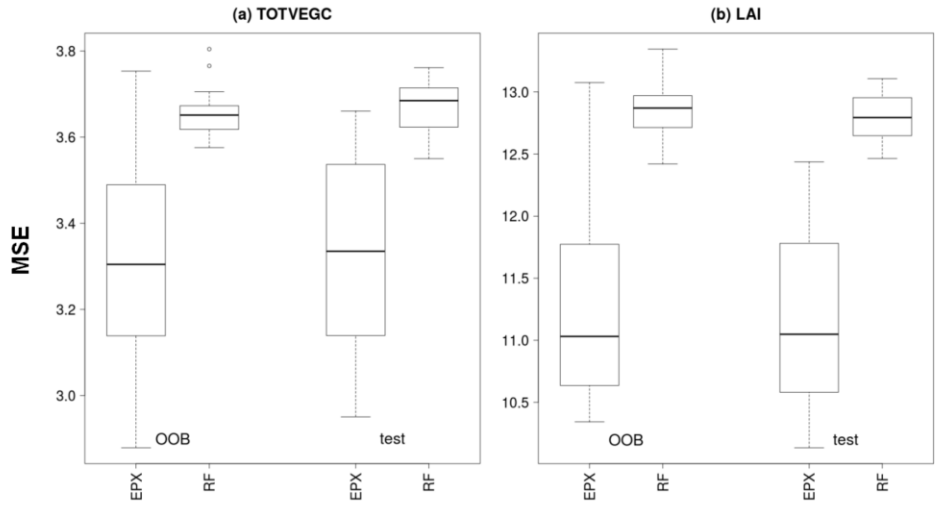

Figure 4.2: Boxplots of OOB MSE (from 5000 randomly selected training samples) and test errors (from 4983 corresponding testing samples) obtained from 20 random splits of the data into training and testing sets

the less important initial groups, ERPX is able to produce more accurate prediction than RF with only a small number of "strong" initial groups.

QoI	Run	Number of					OOB MSE	
		Variables	Groups		Phalanxes			
			Initial	Screened	Candidate	Screened	ERPX	RF
TOTVEGC	1	79	79	16	1	1	3.47	3.66
	2	79	79	14	1	1	3.57	3.67
	3	79	79	14	1	1	3.10	3.62
LAI	1	79	79	18	1	1	10.34	12.63
	2	79	79	20	1	1	11.02	12.94
	3	79	79	14	1	1	11.40	12.89

Table 4.6: Number of variables, initial groups, screened groups, candidate phalanxes, screened phalanxes and prediction accuracies for TOTVEGC and LAI from CLM-CN simulations. Three experiments based on random splitting of training and testing sets are presents.

4.5.3 Gene Expression Data Set

Scheetz et al. [29] conducted a study of mammalian eye diseases where the gene ex-

pressions of the eye tissues from 120 twelve-week-old male F2 rats were recorded. A gene coded as TRIM32 is of particular interest here since it is responsible for causing Bardet-Biedl syndrome.

According to Scheetz et al. [29], only 18976 probe sets exhibited sufficient signal for reliable analysis and at least 2-fold variation in expressions. The intensity values of these genes are evaluated on the logarithm scale and normalized using the method in Bolstad et al. [4]. It is believed from previous studies that TRIM32 is only linked to a small number of genes, so following Scheetz et al. [29] we concentrate mainly on the top 5000 genes with the highest marginal sample variance.

Again, we choose RF (MSE around 0.0128) as the base regression model over lasso (MSE around 0.0131). We apply ERPX and the original RF to this data set three times each. The results are presented in Table 4.7. For ERPX, we skip the initial grouping step.

	Number of						
		Groups		Phalanxes		OOB MSE	
Run	Variables	Initial	Screened	Candidate	Screened	ERPX	RF
1	5000	5000	659	4	2	0.0112	0.0130
2	5000	5000	578	4	3	0.0114	0.0125
3	5000	5000	513	5	4	0.0110	0.0128

Table 4.7: Number of variables, initial groups, screened groups, candidate phalanxes, screened phalanxes and prediction accuracies for the gene expression data set.

It is clear that ERPX is providing more accurate predictions than RF for this data set.

4.5.4 Glass Data Set

The glass data set [20] was obtained from an electron probe X-ray microanalysis (EPXMA) of archaeological glass samples. A total of 180 glass samples were analyzed and each glass sample has a spectrum with 640 wavelengths. The goal is to predict the concentrations of several major constituents of glass, namely, Na2O, SiO2, K2O and CaO, from the spectrum. For different responses, the choice of the base regression model varies, since neither RF nor Lasso performs uniformly better accross the response variables. We apply ERPX and the corresponding base

regression model to each of the six responses three times each. The results are presented in Table 4.8. As we can see, ERPX improves the prediction accuracy of the corresponding base regression method.

			Number of						
				Groups		Phalanxes		MSE	
	Base	Run	Variables	Initial	Screened	Candidate	Screened	ERPX	Base
Na2O	RF	1	640	640	224	3	3	0.78	0.94
		2	640	640	247	2	2	0.86	0.92
		3	640	640	245	2	2	0.85	0.96
SiO2	RF	1	640	640	86	1	1	0.98	1.23
		2	640	640	86	2	2	0.98	1.26
		3	640	640	89	1	1	0.97	1.22
K2O	Lasso	1	640	640	309	1	1	0.094	0.100
		2	640	640	425	1	1	0.093	0.093
		3	640	640	257	1	1	0.096	0.096
CaO	Lasso	1	640	640	302	1	1	0.109	0.111
		2	640	640	201	1	1	0.109	0.111
		3	640	640	265	1	1	0.111	0.112

Table 4.8: Number of variables, base regression model, initial groups, screened groups, candidate phalanxes, screened phalanxes and prediction accuracies (OOB-MSE for RF as base regression model, 5-fold CV-MSE for Lasso as base regression model) for four responses of the glass data set.

4.6 Conclusion

In this chapter, we propose a novel framework called ensemble of Regression Phalanxes (ERPX). We propose to divide a often large number of features into subsets called Regression Phalanxes. Separate predictive models are built using features in each Regression Phalanx and they are further ensembled.

The proposed approach is widely applicable. We have demonstrated it on a variety of applications spanning chemistry, drug discovery, climate-change ecology, and gene expression. The simulated examples and real applications demonstrate that ERPX can take advantage of the richness of the features and produce gains in prediction accuracy over effective base regression model such as Lasso and RF.

Chapter 5

Conclusion and Future Work

5.1 Summary

In this thesis, we focus on two use cases of hierarchical clustering – clustering observations for exploratory analysis and clustering features for adaptive feature grouping and ensembling.

Hierarchical clustering is widely used to cluster observations for exploratory analysis. Although hierarchical clustering is a well-established technique for this task, it may yield unsatisfactory clustering accuracy and interpretation in high-dimensional settings. We point out two main issues that may jeopardize the performance of the traditional hierarchical clustering – noise features and data contaminations. Noise features in high-dimensional data may introduce undesired perturbations that cover the true underlying clusters, while data contaminations such as outliers may deteriorate the feature selection quality.

Witten and Tibshirani [38] proposed a hierarchical clustering framework called Sparse Hierarchical Clustering (SHC) which adaptively chooses the important features for clustering from the noise features. However, its performance deteriorates significantly under data contaminations. Therefore, we propose a novel framework called Robust Sparse Hierarchical Clustering (RSHC) which handles noise features and data contaminations simultaneously. In RSHC, the important features are chosen using a novel sparse principal component analysis called Robust Sparse Principal Component with τ estimator (RSPC-τ). Using τ-estimator in place of the

non-robust cost function in the PCA optimization problem leads to a robust feature selection procedure. The features are selected if they correspond to non-zero loadings in the Robust Sparse PC obtained from applying RSPC-τ to the original data. We show that using the unweighted features, we can build desirable dendrograms even under data contaminations in both noise and important features. We also propose a way of automatically choosing the number of important features to select. Through extensive simulations and real data examples, we show that the proposed RSHC outperforms both the traditional hierarchical clustering as well as SHC in both clustering accuracy and interpretability.

However, as we further study the performance of RSHC, we notice that there are still room for improvements. In high-dimensional data, the important features that separate the underlying clusters may exhibit a complex structure. We show that RSHC and SHC are sometimes inefficient when selecting features with complex structures. Therefore, we propose another framework called Multi-rank Sparse Hierarchical Clustering (MrSHC) that conducts robust feature selection for hierarchical clustering, even if the features have complex structures. MrSHC incorporates different rank considerations in the feature selection process. Instead of choosing features based on one SPC, MrSHC considers features with non-zero loadings in multiple SPC generated from Sparse Principal Component (or RSPC-τ for better robustness). We also propose algorithms to automatically choose the number of rank and the number of important features. Using simulated and real data examples, we show that MrSHC and its robust version Robust-MrSHC yield better clustering accuracy and interpretability in high-dimensional settings even when the data is contaminated and of complex structure.

Using hierarchical clustering to cluster features for adaptive feature grouping and ensembling is a relatively new concept. It is first introduced in the context of rare-class classification problems by Tomal et al. [34]. We focus on regression tasks and propose a general ensemble framework called Regression Phalanxes. We form Regression Phalanxes by adaptively selecting subsets of features. Features in one Regression Phalanx work well together as predictors in a pre-defined regression procedure. Models built on different Regression Phalanxes are then ensembled to form a final regression model. We show that via simulation studies and real data examples, Regression Phalanxes can further boost the performance of a reasonable

regression model such as Lasso or Random Forests.

5.2 Future Work

5.2.1 Clustering Observations

For the purpose of clustering observations, the proposed RSHC and MrSHC can handle noise features and outliers simultaneously. Future work is still needed to address the following issues.

Missing Values

It is likely that there are missing values in the high-dimensional data. Neither RSHC nor MrSHC is able to handle missing values. We would like to endow our proposed frameworks with automatic imputation mechanisms. A potential approach would be replacing the τ estimator with a robust estimator that can handle missing values in RSPC-τ. The new RSPC method will be able to handle both outliers and missing values in the feature selection procedure. After we obtain the robust sparse PCs, we can then impute the missing values according to the low-rank approximations. The imputed data will be further used for hierarchical clustering.

Parallel Computing

The computational cost of MrSHC is relatively high. Fortunately, the feature and rank selection procedure is embarrassingly parallel. Therefore, we can make MrSHC scalable by paralleling computing. MrSHC is currently implemented in R, therefore, we plan to investigate packages such as *snow* or *foreach*. With a sufficient parallel computing package, the computational time needed for MrSHC would be reduced significantly.

Normalization of High-dimensional Data

All the real data examples used in this thesis are pre-normalized using subject-area knowledge before published publicly. However, we would like to extend our frameworks to data sets without pre-normalization. A natural way of normalizing the data is to standardize the features with their corresponding mean and standard deviation. Although this approach can eliminate the effect of measurement units (e.g. meters, milli-meters, etc), it may result in significantly deteriorated clustering accuracy since it may potentially downscale the effect (i.e. variance) of the

important features that contribute significantly to the separation of clusters. Therefore, we would like to explore other ways of normalization such that the resulting clustering accuracy and interpretability are satisfactory.

5.2.2 Regression Phalanxes

The proposed Regression Phalanxes are ensembled through unweighted average of their predictions. We can potentially increase the prediction accuracy of Regression Phalanxes by using weighted ensembling methods, where the weights for the phalanxes are chosen in an adaptive fashion according to their prediction accuracies. Instead of just considering the overall prediction accuracy, we may change the goal of Regression Phalanxes to ranking the predictions. For example, chemists may be interested in ranking the activity of the compounds. We will need to use objective functions such as hit rate instead of mean squared errors. We may also consider other types of base regression models other than Lasso and Random Forest to explore whether the boost in performance is model-dependent, e.g. tree-based methods tend to enjoy a larger boost in performance than linear methods. We also plan to investigate a parallel computing paradigm for the hierarchical clustering phase of Regression Phalanxes. There are very few developments around parallel computing strategy for hierarchical clustering, which I think could be an important topic to work on.

Bibliography

[1] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, 2000. → pages 49

[2] V. Batagelj. Generalized ward and related clustering problems. *Classification and related methods of data analysis*, pages 67–74, 1988. → pages 4

[3] G. Boente and M. Salibian-Barrera. S-estimators for functional principal component analysis. *Journal of the American Statistical Association*, 110 (511):1100–1111, 2015. → pages 21

[4] B. M. Bolstad, R. A. Irizarry, M. Åstrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003. → pages 76

[5] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996. → pages 14

[6] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. → pages 15, 16, 61

[7] L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and regression trees, the wadsworth statistics/probability series. 1984. *Wadsworth International Group, Belmont, CA*. → pages 14

[8] F. R. Burden. Molecular identification number for substructure searches. *Journal of Chemical Information and Computer Sciences*, 29(3):225–227, 1989. → pages 72

[9] R. E. Carhart, D. H. Smith, and R. Venkataraghavan. Atom pairs as molecular features in structure-activity studies: definition and applications.

Journal of Chemical Information and Computer Sciences, 25(2):64–73, 1985. → pages 72

[10] H. Chipman and R. Tibshirani. Hybrid hierarchical clustering with applications to microarray data. *Biostatistics*, 7(2):286–301, 2006. → pages 28

[11] C. Croux, P. Filzmoser, and H. Fritz. Robust sparse principal component analysis. *Technometrics*, 55(2):202–214, 2013. → pages 24

[12] M. Dettling. Bagboosting for tumor classification with gene expression data. *Bioinformatics*, 20(18):3583–3593, 2004. → pages vi, xii, 49, 50, 51, 55, 56

[13] S. Dudoit, J. Fridlyand, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American statistical association*, 97(457):77–87, 2002. → pages 49

[14] B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*. CRC press, 1994. → pages 14

[15] K. Esbensen, T. Midtgaard, and S. Schönkopf. *Multivariate Analysis in Practice: A Training Package*. Camo As, 1996. → pages 67

[16] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010. → pages 67

[17] J. H. Friedman and J. J. Meulman. Clustering objects on subsets of attributes (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(4):815–849, 2004. → pages 7

[18] J. M. Hughes-Oliver, A. D. Brooks, W. J. Welch, M. G. Khaledi, D. Hawkins, S. S. Young, K. Patil, G. W. Howell, R. T. Ng, and M. T. Chu. Chemmodlab: a web-based cheminformatics modeling laboratory. *In silico biology*, 11(1-2):61–81, 2010. → pages 72

[19] Y. Kondo, M. Salibian-Barrera, and R. Zamar. A robust and sparse k-means clustering algorithm. *arXiv preprint arXiv:1201.6082*, 2012. → pages 18

[20] P. Lemberge, I. De Raedt, and K. H. Janssens. Quantitative analysis of 16-17th century archaeological glass vessels using pls regression of epxma and mu-xrf data. *Journal of chemometrics*, 14(5):751–764, 2000. → pages 76

[21] K. Liu, J. Feng, and S. S. Young. Powermv: a software environment for molecular viewing, descriptor generation, data analysis and hit evaluation. *Journal of chemical information and modeling*, 45(2):515–522, 2005. → pages 72

[22] R. Maronna. Principal components and orthogonal regression based on robust scales. *Technometrics*, 47(3):264–273, 2005. → pages 22

[23] W. Pan and X. Shen. Penalized model-based clustering with application to variable selection. *The Journal of Machine Learning Research*, 8: 1145–1164, 2007. → pages 7

[24] C. M. Perou, T. Sørlie, M. B. Eisen, M. van de Rijn, S. S. Jeffrey, C. A. Rees, J. R. Pollack, D. T. Ross, H. Johnsen, L. A. Akslen, et al. Molecular portraits of human breast tumours. *Nature*, 406(6797):747–752, 2000. → pages vi, xi, xii, 31, 32, 33, 49, 50, 52, 56, 58

[25] A. E. Raftery and N. Dean. Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101(473):168–178, 2006. → pages 7

[26] P. Rousseeuw and V. Yohai. Robust regression means of S-estimators. In *Robust and Nonlinear Time Series Analysis. Lecture Notes in Statist.*, 26, pages 256–272, New York, 1984. Springer. → pages 86

[27] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987. → pages 42

[28] K. Sargsyan, C. Safta, H. N. Najm, B. J. Debusschere, D. Ricciuto, and P. Thornton. Dimensionality reduction for complex models via bayesian compressive sensing. *International Journal for Uncertainty Quantification*, 4(1), 2014. → pages 73

[29] T. E. Scheetz, K.-Y. A. Kim, R. E. Swiderski, A. R. Philp, T. A. Braun, K. L. Knudtson, A. M. Dorrance, G. F. DiBona, J. Huang, T. L. Casavant, et al. Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, 103(39): 14429–14434, 2006. → pages 75, 76

[30] T. Strauss and M. J. von Maltitz. Generalising ward?s method for use with manhattan distances. *PloS one*, 12(1):e0168288, 2017. → pages 4

[31] W. Sun, J. Wang, Y. Fang, et al. Regularized k-means clustering of high-dimensional data and its asymptotic consistency. *Electronic Journal of Statistics*, 6:148–167, 2012. → pages 7

[32] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. → pages 11, 61

[33] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001. → pages 42

[34] J. H. Tomal, W. J. Welch, R. H. Zamar, et al. Ensembling classification models based on phalanxes of variables with applications in drug discovery. *The Annals of Applied Statistics*, 9(1):69–93, 2015. → pages 60, 63, 79

[35] L. J. van't Veer, H. Dai, M. J. Van De Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, et al. Gene expression profiling predicts clinical outcome of breast cancer. *nature*, 415 (6871):530–536, 2002. → pages vi, xi, xii, 31, 34, 36, 49, 51, 53, 57, 59

[36] W. N. Venables and B. D. Ripley. *Modern applied statistics with S-PLUS*. Springer Science & Business Media, 2013. → pages 14

[37] S. Wang and J. Zhu. Variable selection for model-based high-dimensional clustering and its application to microarray data. *Biometrics*, 64(2):440–448, 2008. → pages 7

[38] D. M. Witten and R. Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490): 713–726, 2010. → pages vi, 7, 8, 9, 10, 18, 31, 34, 38, 39, 50, 51, 59, 78

[39] D. M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009. → pages 9, 20

[40] B. Xie, W. Pan, and X. Shen. Penalized model-based clustering with cluster-specific diagonal covariance matrices and grouped variables. *Electronic journal of statistics*, 2:168, 2008. → pages 7

[41] V. Yohai and R. Zamar. High breakdown-point estimates of regression by means of the minimization of an efficient scale. *Journal of the American Statistical Association*, 83(402):406–413, 1988. → pages 22, 87

Appendix A

Supporting Materials

Algorithm to Compute RSPC-τ

Following the notations introduced in Section 2.3.1, we present an iterative approximation approach for computing RSPC-τ. Recall that the objective function for RSPC-τ is (2.3):

$$\min_{\boldsymbol{a},\boldsymbol{b},\boldsymbol{\mu}} \left\{ \sum_{j=1}^{p} \widehat{\sigma}_j^2 + \lambda ||\boldsymbol{b}||_1 \right\}, \quad \text{subject to } ||\boldsymbol{b}||_2^2 = 1,$$

where λ is a tuning parameter that controls the sparseness of the feature weights, $\boldsymbol{b}$.

Since τ-estimator of scale is used for $\widehat{\sigma}_j$ for $j = 1, ..., p$, we replace the notation $\widehat{\sigma}_j$ by $\widehat{\tau}_j$ and re-write (2.3) as

$$\min_{\boldsymbol{a},\boldsymbol{b},\boldsymbol{\mu}} \left\{ \sum_{j=1}^{p} \widehat{\tau}_j^2 + \lambda ||\boldsymbol{b}||_1 \right\}, \quad \text{subject to } ||\boldsymbol{b}||_2^2 = 1, \tag{A.1}$$

Given $\boldsymbol{a}, \boldsymbol{b}$ and $\boldsymbol{\mu}$, let $\widehat{\sigma}_j^2$ be the M-estimate of scale [26] based on loss function ρ_{c_1}, which satisfies

$$\frac{1}{n} \sum_{i=1}^{n} \rho_{c_1} \left(\frac{r_{ij}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\mu})}{\widehat{\sigma}_j} \right) = b_1. \tag{A.2}$$

Then the τ-scale $\widehat{\tau}_j = \widehat{\tau}_j(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\mu})$ is defined by

$$\widehat{\tau}_j^2 = \frac{\widehat{\sigma}_j^2}{b_2} \cdot \frac{1}{n} \sum_{i=1}^{n} \rho_{c_2} \left(\frac{r_{ij}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\mu})}{\widehat{\sigma}_j} \right), \tag{A.3}$$

with where $\rho_{c_1}(t) = \rho(t/c_1)$, $\rho_{c_2}(t) = \rho(t/c_2)$ are assumed to be symmetric, continuously differentiable, bounded, strictly increasing on $[0, c_j]$ and constant on $[c_j, \infty)$, with $0 < c_j < \infty$, $j = 1, 2$. Here we consider Yohai-Zamar loss function

$$\rho(t) = \begin{cases} 1.38 \left(\frac{t}{c}\right)^2 & \left|\frac{t}{c}\right| \leq \frac{2}{3} \\ 0.55 - 2.69 \left(\frac{t}{c}\right)^2 + 10.76 \left(\frac{t}{c}\right)^4 - 11.66 \left(\frac{t}{c}\right)^6 + 4.04 \left(\frac{t}{c}\right)^8 & \frac{2}{3} < \left|\frac{t}{c}\right| \leq 1 \\ 1 & \left|\frac{t}{c}\right| > 1. \end{cases} \tag{A.4}$$

To obtain consistency and 50% breakdown point, we choose the tuning parameters $c_1 = 1.214$ and $b_1 = 0.5$ for ρ_{c_1}. The choices $c_2 = 3.270$ and $b_2 = 0.128$ for ρ_{c_2} yield a τ-estimator with 95% efficiency when the errors are normally distributed (see Yohai and Zamar [41]).

Given $\boldsymbol{b}$, the estimating equations for a_i and μ_j are given by the following formulas which are obtained by fixing $\boldsymbol{b}$ and solving the subgradient equations in (A.1) with respect to a_i and μ_j:

$$a_i = \frac{\sum_{j=1}^{p} w_{ij}(Z_{ij} - \mu_j) b_j}{\sum_{j=1}^{p} w_{ij} b_j^2}, \quad 1 \leq i \leq n, \tag{A.5}$$

$$\mu_j = \frac{\sum_{i=1}^{n} w_{ij}(Z_{ij} - a_i b_j)}{\sum_{i=1}^{n} w_{ij}}, \quad 1 \leq j \leq p, \tag{A.6}$$

with

$$w_{ij} = \left[d_j h_j^{-1} \rho'_{c_1} \left(\frac{r_{ij}}{\widehat{\tau}_j} \right) + \rho'_{c_2} \left(\frac{r_{ij}}{\widehat{\tau}_j} \right) \widehat{\tau}_j \right] r_{ij}^{-1}, \tag{A.7}$$

$$h_j = \sum_{i=1}^{n} \rho'_{c_1} \left(\frac{r_{ij}}{\widehat{\tau}_j} \right) \frac{r_{ij}}{\widehat{\tau}_j},$$

and

$$d_j = 2\widehat{\tau}_j \left[\sum_{i=1}^{n} \rho_{c_2} \left(\frac{r_{ij}}{\widehat{\tau}_k} \right) \right] - \sum_{i=1}^{n} \rho'_{c_2} \left(\frac{r_{ij}}{\widehat{\tau}_k} \right) r_{ij}.$$

The minimization over $\boldsymbol{b}$ for fixed $\boldsymbol{a}$ and $\boldsymbol{\mu}$ is more involved due to the L1-penalty. As such we use the following fast and approximated iterative approach to obtain an estimating equation for $\boldsymbol{b}$. First we re-write (A.1) as

$$\min_{\boldsymbol{a},\boldsymbol{b},\boldsymbol{\mu}} \left\{ \sum_{j=1}^{p} \widehat{\tau}_j^2 + \lambda \sum_{j=1}^{p} g(b_j) \right\}, \quad \text{subject to } ||\boldsymbol{b}||_2^2 = 1. \tag{A.8}$$

where

$$g(t) = \begin{cases} \frac{t^2}{|t|} & \text{if } t \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

At this point, we make the following approximation to simplify the optimization procedure. If $|b_j|(j = 1, \ldots, p)$ in the denominator of $g(b_j)$ are non-zero, then we consider them constants $b_j^{(const)}(j = 1, \ldots, p)$, and Equation A.8 is approximated as follows.

$$\min_{\boldsymbol{b}} \left\{ \sum_{j=1}^{p} \widehat{\tau}_j^2 + \lambda \sum_{j=1}^{p} \frac{b_j^2}{|b_j^{(const)}|} \right\}, \quad \text{subject to } ||\boldsymbol{b}||_2^2 = 1. \tag{A.9}$$

By taking the derivative of the above objective function with respect to b_j (ignoring the side constraints $||\boldsymbol{b}||_2^2 = 1$ since it is just for ensuring the uniqueness of the solution) and setting it to 0, we have the closed form solution for b_j:

$$b_j = \frac{\sum_{i=1}^{n} w_{ij}(Z_{ij} - \mu_j)a_i}{\sum_{i=1}^{n} w_{ij}(a_i)^2 + 2\lambda/|b_j^{(const)}|}. \tag{A.10}$$

Since $b_j^{(const)}$ is unknown, we initialize it with an non-zero value and obtain b_j using the following iteration:

$$b_j^{(k+1)} = \frac{\sum_{i=1}^{n} w_{ij}(Z_{ij} - \mu_j)a_i}{\sum_{i=1}^{n} w_{ij}(a_i)^2 + 2\lambda/|b_j^{(k)}|}. \tag{A.11}$$

The iteration steps in Equation A.11 will continue until convergence or the value of $b_j^{(k+1)}$ is smaller than a pre-defined threshold, say 10^{-20}, in which case the value of b_j is set to 0.

Note that the constraint on $\boldsymbol{b}$ (i.e. $||\boldsymbol{b}||_2^2 = 1$) is just to ensure the uniqueness of

the solution, we normalize $\boldsymbol{b}$ after the iteration terminates for all the $b_j (j = 1, \ldots, p)$ in $\boldsymbol{b}$.

We use the L1-median of the input matrix as the initial values of $\boldsymbol{\mu}$. For the vector $\boldsymbol{b}$, we generate R random starts by assigning $\boldsymbol{b}^{(0)}$ either by one of the $R-1$ randomly selected rows of the original matrix or the SVD solution obtained from the original matrix, i.e. $\boldsymbol{b}^{svd}$. The columns of the matrix A are the sparse scores of each observation given $\boldsymbol{b}^{(0)}$. For each set of the initial values we run max_iter iterations, or until a tolerance level is achieved.

Algorithm 8 gives a detailed description of the algorithm used for computing RSPCA-τ. We observe that, in Algorithm 8, for set of initial values, the value of the objective function does not always decrease during the iterations. Therefore, we cache the smallest value of the objective function and its corresponding $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{\mu}$ among all the runs with different initial values to be returned as the final solution. Although we do not provide theoretical guarantee on the convergence of the iteration process, we observe convergence within a pre-specified number of max_iter (say 500) in all of our simulations and real examples.

Set	MSE	
	Lasso	RF
BN	152.53	127.39
PH	145.29	135.01
FP	143.56	125.98
AP	148.45	131.76
CAP	146.57	131.67

Table A.1: MSEs of Lasso (5-fold CV-MSEs) and RF (OOB-MSEs) for five descriptor sets of the AID 364 assay

Response	MSE	
	Lasso	RF
LAI	12.37	11.58
TOTVEGC	3.47	3.29

Table A.2: MSEs of Lasso (CV-MSEs) and RF (OOB-MSEs) for two responses of the CLM data

Algorithm 8 RSPCA-τ

Let $L_S(\boldsymbol{a},\boldsymbol{b},\boldsymbol{\mu}) = \sum_{j=1}^{p} \widehat{\tau}_j^2 + \lambda||\boldsymbol{b}||_1$ subject to $||\boldsymbol{b}||_2 = 1$
Input: $\mathbf{Z} = (\mathbf{Z}_1, \cdots, \mathbf{Z}_p) \in \mathbb{R}^{n\times p}$, $M \in \mathbb{N}$, tuning parameter $\lambda > 0$, convergence threshold ω, default 10^{-10}, max number of iteration *max_iter*, default 500, number of initial value set R
$L_S^* = +\infty$
Generate $\boldsymbol{B}$ including R random starts for $\boldsymbol{b}$ by randomly selection $R-1$ rows from $\mathbf{Z}$, and $\boldsymbol{b}^{SVD}$.
for $\boldsymbol{b}^{(0)}$ in $\boldsymbol{B}$
    $\boldsymbol{a}^{(0)} \leftarrow \mathbf{Z}\boldsymbol{b}^{(0)}$, $\boldsymbol{\mu}^{(0)} \leftarrow (\text{median}(\mathbf{Z}_1), ..., \text{median}(\mathbf{Z}_p))^T$.
    $N\_iter = 1$
    while $N\_iter < max\_iter$
        compute $w_{ij}, i = 1, ..., n, j = 1, ..., p$ in (A.7) using $\boldsymbol{a}^{(0)}, \boldsymbol{b}^{(0)}, \boldsymbol{\mu}^{(0)}$.
        $a_i^{(1)} \leftarrow (\sum_{j=1}^{p} w_{ij}(Z_{ij} - \mu_j^{(0)})b_j^{(0)})/(\sum_{j=1}^{p} w_{ij}(b_j^{(0)})^2), i = 1, ..., n$
        for $j = 1, \cdots, p$:
            if $b_j^{(0)} = 0$
                $b_j^{(1)} \leftarrow 0$
            else
                $b_j^{(0*)} \leftarrow b_j^{(0)}$
                $n\_iter = 1$
                while $n\_iter < max\_iter$
                    $b_j^{(1*)} \leftarrow \left(\sum_{i=1}^{n} w_{ij}(Z_{ij} - \mu_j^{(0)})a_i^{(0)}\right) / \left(\sum_{i=1}^{n} w_{ij}(a_i^{(0)})^2 + 2\lambda/|b_j^{(0*)}|\right)$
                    if $|b_j^{(1*)} - b_j^{(0*)}| < \omega$ or $|b_j^{(1*)}| < \omega^2$
                        break
                    $b_j^{(0*)} \leftarrow b_j^{(1*)}$
                    n_iter++
                end while
                $b_j^{(1)} \leftarrow b_j^{(1*)}$ if $|b^{(1*)}| > \omega^2$, otherwise $b_j^{(1)} \leftarrow 0$
        end for
        Normalize $b_j^{(1)} (j = 1, \ldots, p)$ so that $\sum_j (b_j^{(1)})^2 = 1$
        $\mu_j^{(1)} \leftarrow (\sum_{i=1}^{n} w_{ij}(Z_{ij} - a_i^{(1)} b_j^{(1)}))/(\sum_{i=1}^{n} w_{ij})$
        if $L_S(\boldsymbol{a}^{(1)}, \boldsymbol{b}^{(1)}, \boldsymbol{\mu}^{(1)}) < L_S^*$
            $L_S^* = L_S(\boldsymbol{a}^{(1)}, \boldsymbol{b}^{(1)}, \boldsymbol{\mu}^{(1)})$, $\boldsymbol{a}^* = \boldsymbol{a}^{(1)}$, $\boldsymbol{b}^* = \boldsymbol{b}^{(1)}$, $\boldsymbol{\mu}^* = \boldsymbol{\mu}^{(1)}$
        if $|L_S(\boldsymbol{a}^{(1)}, \boldsymbol{b}^{(1)}, \boldsymbol{\mu}^{(1)}) - L_S(\boldsymbol{a}^{(0)}, \boldsymbol{b}^{(0)}, \boldsymbol{\mu}^{(0)})| < \varepsilon$, say 10^{-5}
            break
        $\boldsymbol{a}^{(0)} \leftarrow \boldsymbol{a}^{(1)}, \boldsymbol{b}^{(0)} \leftarrow \boldsymbol{b}^{(1)}, \boldsymbol{\mu}^{(0)} \leftarrow \boldsymbol{\mu}^{(1)}$
        N_iter++
    end while
end for
Output: $\boldsymbol{a}^*$, $\boldsymbol{b}^*$, $\boldsymbol{\mu}^*$ with L_S^*.