NEW STATISTICAL LEARNING APPROACHES WITH APPLICATIONS TO RNA-SEQ DATA

Patrick K. Kimes

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research.

Chapel Hill
2015

Approved by:

Yufeng Liu

J. S. Marron

D. Neil Hayes

Jan Hannig

Kai Zhang

ABSTRACT

Patrick K. Kimes: New Statistical Learning Approaches with Applications to RNA-seq Data
(Under the direction of Yufeng Liu, J. S. Marron, D. Neil Hayes)

This dissertation examines statistical learning problems in both the supervised and unsupervised settings. The dissertation is composed of three major parts. In the first two, we address the important question of significance of clustering, and in the third, we describe a novel framework for unifying hard and soft classification through a spectrum of binary learning problems.

In the unsupervised task of clustering, determining whether the identified clusters represent important underlying structure, or are artifacts of natural sampling variation, has been a critical and challenging question. In this dissertation, we introduce two new methods for addressing this question using statistical significance. In the first part of the dissertation, we describe SigFuge, an approach for identifying genomic loci exhibiting differential transcription patterns across many RNA-seq samples. In the second part of this dissertation, we describe statistical Significance of Hierarchical Clustering (SHC), a Monte Carlo based approach for testing significance in hierarchical clustering, and demonstrate the power of the method to identify significant clustering using two cancer gene expression datasets. Both methods were implemented and made available as open source packages in `R`.

In the final part of this dissertation, we propose a spectrum of supervised learning problems which spans the hard and soft classification tasks based on fitting multiple decision rules to a dataset. By doing so, we reveal a novel collection of binary supervised learning problems. We study the problems using the framework of large-margin classification and a class of piecewise linear surrogate losses, for which we derive statistical properties. We evaluate our approach using simulations and a magnetic resonance imaging (MRI) dataset from the Alzheimer's Disease Neuroimaging Initiative (ADNI) study.

To お母さん and お父さん.
To mom and dad.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
Introduction

With advances in computing and data collection, fields such as genomics are producing larger and increasingly more complex data. As such, a need for newer and more powerful tools for data analysis is rapidly growing. The field of statistical machine learning encompasses an expanding collection of computational methods for uncovering patterns in data, many of which were developed to address the new and challenging problems encountered in practice.

In statistical machine learning, a distinction is traditionally made between approaches for supervised and unsupervised learning. Supervised tasks involve learning a pattern given both an outcome and a set of covariates. Typical supervised tasks include the standard regression and classification problems. In contrast, unsupervised tasks involve learning a pattern from a set of covariates in the absence of an outcome. Popular unsupervised problems included clustering and dimension reduction.

In this dissertation, we investigate topics in classification and clustering, two widely popular tasks of statistical learning. Much of this work is motivated by problems arising in modern genomic studies. To help frame the completed work, in this chapter we provide a review of classification (Section 1.1), clustering (Section 1.2) and high-throughput mRNA sequencing (RNA-seq; Section 1.3). We conclude this chapter by summarizing the major ideas presented in the remainder of this dissertation (Section 1.4).

1.1 Classification

Classification is one of the most widely applied and well studied problems in supervised learning. Given a training set of observed covariates and outcomes, the goal of classification is to build a prediction model. While similar to the usual regression problem (with continuous response), classification describes the particular setting where the outcome is a discrete class label. In binary classification, the label takes one of two possible values, typically denoted by -1 and $+1$. While

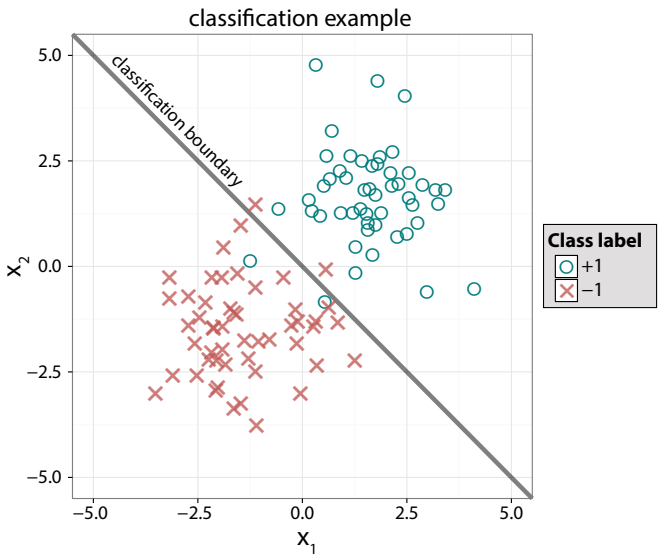

Figure 1.1: A simple example of classification is shown where covariates (x_1, x_2) were observed for 50 (-1) class and 50 $(+1)$ class instances. Observations from the -1 and $+1$ classes are denoted by x's and o's, respectively. The classification boundary separating the two classes may be used to predict the labels of future observations.

generalizations to more than two classes exist, in this dissertation we focus only on the binary problem.

In Figure 1.1, we illustrate binary classification using a simple example where the covariates x_1 and x_2 were measured for a training set of 100 observations, 50 from class -1 and 50 from class $+1$. Given the set of points, we want to estimate a rule for predicting the class of an unlabeled (x_1^*, x_2^*) pair. Consider the line $x_1 + x_2 = 0$ passing through the center of the plot. One possible rule is to predict labels by checking whether an observation falls above or below the line. More formally, this can be expressed as predicting the class of (x_1^*, x_2^*) to be the sign of $(x_1^* + x_2^*)$. The affine hyperplane partitioning the covariate space according to the predicted label is commonly called a separating hyperplane or classification boundary.

A closely related problem to classification is that of *conditional class probability estimation*. While both problems take covariates and a discrete label as inputs, the two differ by the modeled output. Rather than simply predict the most likely class, in the probability estimation task we seek to estimate the probability of each class conditional on an observed covariate value. This is particularly useful in settings where classification certainty is of interest. Consider again the example in Figure 1.1 and suppose the class labels -1 and $+1$ correspond to the status of a severe

disease. In this case, mild evidence of belonging to the disease positive class, +1 class may still warrant further follow-up. This type of inference is not possible using standard classification. The relationship between conditional class probability estimation and class prediction is often studied as soft vs. hard classification or generative vs. discriminative learning, and will be discussed in more detail later in this section. First, we briefly introduce some popular approaches to classification, giving particular attention to margin-based classifiers. For a more thorough treatment of these and other approaches, we refer the reader to chapters 13 and 14 of Hastie et al. (2011).

Formalizing the description given above, let $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ denote a training set of n i.i.d. covariate–label pairs drawn from an unknown distribution $\mathcal{P}(\boldsymbol{X}, Y)$ defined over $\mathcal{X} \times \mathcal{Y}$. Here, $\mathcal{X}$ is used to denote the p-dimensional covariate space and $\mathcal{Y} = \{-1, +1\}$ the binary label space. In classification, we estimate some rule $\widehat{Y} : \mathcal{X} \to \mathcal{Y}$ to predict the label for an unlabeled $\boldsymbol{X}$. A natural criterion for evaluating a classification rule is the corresponding expected prediction error, also known as the expected *0−1 loss*: $\mathbb{E}_{\boldsymbol{X},Y}\mathbf{I}\{\widehat{Y}(\boldsymbol{X}) \neq Y\}$, where $\mathbf{I}\{\cdot\}$ is used to denote the indicator function.

The classification rule minimizing the expected prediction error is referred to as the Bayes optimal rule and can be shown to equal: $Y^*(\boldsymbol{X}) = \text{sign}\{p(\boldsymbol{X}) - \frac{1}{2}\}$ where $p(\boldsymbol{X}) = \mathbb{P}(Y = +1|\boldsymbol{X})$ denotes the conditional probability of belonging to class +1 given the observed $\boldsymbol{X}$. Note that since $\mathbb{P}(Y = -1|\boldsymbol{X}) = 1 - \mathbb{P}(Y = +1|\boldsymbol{X})$, the conditional class probability at $\boldsymbol{X}$ is completely characterized by $p(\boldsymbol{X})$. Simple manipulation of the Bayes rule reveals the following equivalent form:

$$\begin{aligned} Y^*(\boldsymbol{X}) &= \text{sign}\{p(\boldsymbol{X}) - \tfrac{1}{2}\} \\ &\propto \text{sign}\{\mathbb{P}(Y = +1|\boldsymbol{X}) - \mathbb{P}(Y = -1|\boldsymbol{X})\} \\ &= \underset{y}{\text{argmax}}\, \mathbb{P}(Y = y|\boldsymbol{x}). \end{aligned} \tag{1.1}$$

Thus, the Bayes optimal rule intuitively corresponds to predicting the class with greater theoretical conditional probability.

Various approaches have been proposed for approximating the Bayes rule in classification including likelihood, prototype, and margin-based methods. *Likelihood-based* approaches include the classical Fisher's linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA; Fisher, 1936; Mardia et al., 1979; Rao, 1973). In LDA, the underlying class conditional distributions

$\mathcal{P}(X|Y=+1)$ and $\mathcal{P}(X|Y=-1)$ are assumed to be Gaussian with equal covariance. Maximum likelihood is used to estimate the class conditional means, common covariance and marginal class probabilities. Then, applying Bayes' theorem, the class with larger estimated conditional probability is predicted by LDA. QDA generalizes the LDA approach by allowing the covariance matrices to differ between the two classes. The two methods are named for the shapes of their resulting decision boundaries. While introduced using a Gaussian formulation, we note that similar to least squares, LDA may also be derived non-parametrically, i.e. without the Gaussianity assumption (Mai et al., 2012; Hastie et al., 2011).

Prototype and related methods, such as k-nearest-neighbor (k-NN) and K-means classification approximate the underlying conditional distributions by groups of prototypes (Fix and Hodges, 1989; Cover and Hart, 1967; Duda et al., 2000). These prototypes may be the set of all training observations as in 1-NN or the $2K$ cluster centroids generated by applying K-means to each class. The label of a new observation is predicted by the class of the nearest prototype. Similar to likelihood based methods, prototype methods approximate the underlying class conditional densities, $\mathcal{P}(\boldsymbol{X}|Y=y)$, to estimate a prediction rule.

In contrast to likelihood and prototype based methods, margin-based classifiers directly estimate partially or fully $\mathcal{P}(Y|\boldsymbol{X}=\boldsymbol{x})$. *Margin-based* classification rules take the form of a function, $f:\mathcal{X}\to\mathbb{R}$ from which a class label is predicted based on the sign of f. That is, $\widehat{Y}(\boldsymbol{x})=+1$ is predicted if $f(\boldsymbol{x})>0$, and $\widehat{Y}(\boldsymbol{x})=-1$ is predicted if $f(\boldsymbol{x})<0$. The discriminant boundary shown in Figure 1.1 is an example of a margin-based rule, with $f(\boldsymbol{x})=x_1+x_2$. Typically, we assume $f(\boldsymbol{x})\neq 0$ almost surely and arbitrary let $\widehat{Y}(\boldsymbol{x})=+1$ when $f(\boldsymbol{x})=0$. Commonly, $f(\boldsymbol{x})$ is referred to as the *margin function*.

By definition, the product term $yf(\boldsymbol{x})$, called the functional margin, is such that $yf(\boldsymbol{x})>0$ and $yf(\boldsymbol{x})<0$ correspond respectively to correct and incorrect classification. Furthermore, the functional margin may be interpreted as a rough measure of classification accuracy. Using the functional margin, minimization of the empirical prediction error may be written:

$$\min_{f\in\mathcal{F}} \quad \frac{1}{n}\sum_{i=1}^{n}\mathbf{I}\big\{y_i f(\boldsymbol{x}_i)<0\big\},$$

where $\mathcal{F}$ is a space of margin functions, and the use of $\mathbf{I}\{\cdot\}$ corresponds to $0-1$ loss. However, optimization with respect to $0-1$ loss is typically NP-hard. As such, a *surrogate loss*, $yf(\boldsymbol{x}) \mapsto L$, is commonly used in place. Typically, a regularization term $J : \mathcal{F} \to \mathbb{R}$ is also added to control the complexity of f, with corresponding tuning parameter $\lambda > 0$. Combining the two, a margin-based classifier solves the following optimization problem of the *loss* + *penalty* form:

$$\min_{f \in \mathcal{F}} \ \frac{1}{n} \sum_{i=1}^{n} \underbrace{L\big(y_i f(\boldsymbol{x}_i)\big)}_{loss} \ + \ \underbrace{\lambda J(f)}_{penalty}. \tag{1.2}$$

Some of the most popular methods for classification, including penalized logistic regression (PLR; Lee and Silvapulle, 1988; Le Cessie and Van Houwelingen, 1992), the support vector machine (SVM; Cortes and Vapnik, 1995; Wahba, 1999), Adaboost (Freund and Schapire, 1997; Friedman et al., 2000), and distance-weighted discrimination (DWD; Marron et al., 2007) may be formulated as margin-based problems of the form (1.2). Some common loss functions, shown in Figure 1.2, are given as follows:

$$\begin{aligned}
\text{SVM (hinge)}: &\quad L(z) = \max\{0, 1-z\}, \\
\text{PLR}: &\quad L(z) = \log(1+e^{-z}), \\
\text{squared hinge}: &\quad L(z) = (\max\{0, 1-z\})^2, \\
\text{Adaboost (exponential)}: &\quad L(z) = e^{-z}, \\
\text{DWD}: &\quad L(z) = \begin{cases} \frac{1}{4z} & \text{if } z \geq \frac{1}{2} \\ 1-z & \text{if } z < \frac{1}{2} \end{cases}.
\end{aligned}$$

Margin-based classifiers enjoy substantial popularity in practice for their classification accuracy in both high and low dimensional settings. The theoretical properties of these methods have been studied in depth, shedding light on the reason for their success, and further, the relative advantages of different loss functions (Steinwart and Scovel, 2007; Blanchard et al., 2008; Bartlett et al., 2006; Cristianini and Shawe-Taylor, 2000). We next discuss the important distinction between hard and soft margin classifiers.

When considering loss functions for margin-based classification, an important theoretical property is Fisher consistency (Lin, 2004). A loss function $L(\cdot)$ is called Fisher consistent if

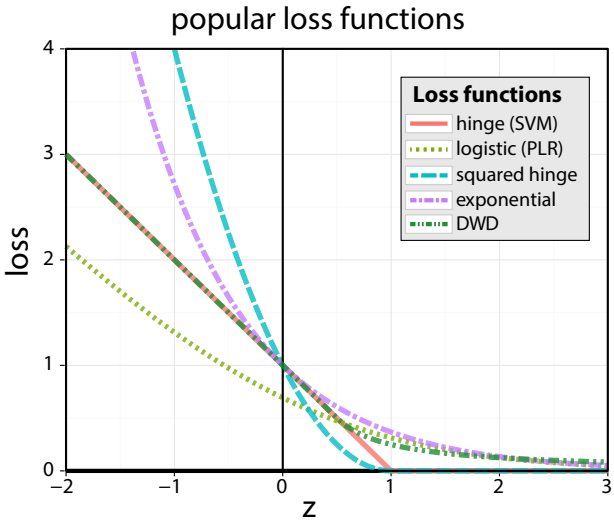

Figure 1.2: Popular choices for the loss function in (1.2) are plotted as a function of z. Typically, for binary classification the functional margin, yf, is used for z. Plotted loss functions include: SVM (red), logistic (light green), squared hinge (blue), exponential (purple), and DWD (dark green) – corresponding line types are given in the figure legend.

$\text{sign}\{f_L^*(\boldsymbol{X})\} = Y^*(\boldsymbol{X})$, where Y^* is the Bayes rule given in (1.1) and

$$f_L^*(\boldsymbol{X}) = \underset{f}{\text{argmin}}\, \mathbb{E}_{Y|\boldsymbol{X}} L\big(Y f(\boldsymbol{X})\big).$$

By the form of the Bayes rule, Fisher consistency only requires identifying a rule which correctly estimates whether the conditional class probability, $p(\boldsymbol{x})$, is greater than or less than $\frac{1}{2}$ at any point $\boldsymbol{x} \in \mathcal{X}$. Some loss functions, when optimized, identify rules which provide direct estimation of $p(\boldsymbol{x})$. The corresponding classifiers defined by these loss functions which directly target the conditional class probability estimation problem, are often called *soft classifiers*. Popular soft classifiers include PLR and Adaboost. In contrast, loss functions which only produce estimates of $p(\boldsymbol{x})$ at $\frac{1}{2}$, i.e. $\text{sign}(p(\boldsymbol{x}) - \frac{1}{2})$, such as SVM, are referred to as *hard classifiers* (Wahba, 1999, 2002). Classification consistency as it relates to hard and soft classification is discussed in greater detail in Subsection 4.2.2.

Despite the intuitive difference between the two approaches, it is not immediately obvious how soft and hard classifiers differ in practice. Recently, Liu et al. (2011) introduced the family of Large-margin Unified Machines (LUM) connecting several popular hard and soft classifiers, including DWD, SVM, Adaboost and PLR. Through their unified framework, the authors provide some insight on the relative advantages of partial and full estimation of $p(\boldsymbol{x})$. In Chapter 4, we

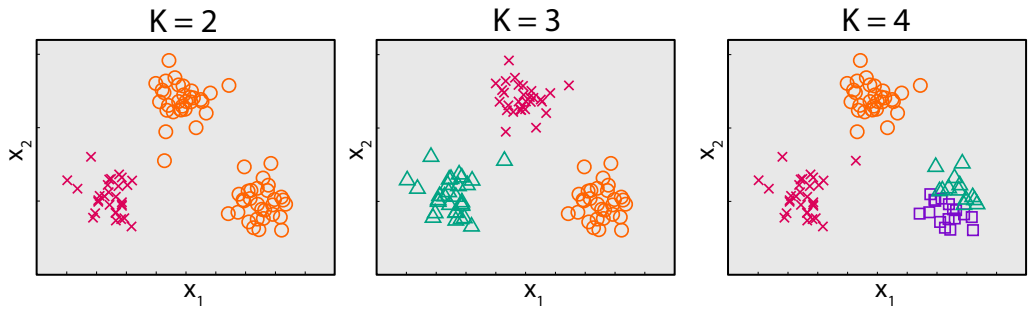

Figure 1.3: The popular K-means clustering algorithm is used to separate the same set of 90 points into 2, 3 and 4 clusters. Color and shape are used to denote cluster labels in each panel. Clustering algorithms may be used to identify any number of clusters from a given data set. In this example, $K = 3$ appears to identify the most natural set of clusters in the data.

present an alternative framework for connecting hard and soft classification through a range of supervised learning problems.

1.2 Clustering

We next provide some background on *clustering*, a popular task in unsupervised learning. In contrast to the classification setting described above, an outcome variable Y is no longer the focus of the analysis. Instead, the aim in unsupervised learning is to gain insight with only the n instances of a p-dimensional variable, $\boldsymbol{X}$.

Clustering is the specific unsupervised task of partitioning a dataset into subsets, called clusters, of similar objects. Figure 1.3 shows an example of clustering applied to a single dataset to obtain $K = 2$, 3 and 4 clusters. Cluster assignments are denoted by colors and symbols. The clusters identified by $K = 3$ appear to capture the most natural structure in the data. In contrast, $K = 2$ and $K = 4$ make unnatural and unintuitive splits in the data. The K-means clustering algorithm used to determine the partitions in Figure 1.3 will be described later in this section.

Fundamental to clustering is the definition of pairwise object *dissimilarity*. Dissimilarity may be thought of as a relaxed notion of distance between two points. Occasionally, all pairwise dissimilarities are explicitly defined for a collection of points through a *proximity matrix*. However, more commonly, dissimilarity is calculated for a set of points using a symmetric dissimilarity function, denoted by $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$. Assuming $\boldsymbol{X} \in \mathbb{R}^p$ is a continuous random variable, some popular choices of this function are:

- *Squared Euclidean distance* (L_2^2) : $d(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \sum_j (x_{ij} - x_{i'j})^2$,
- *Euclidean distance* (L_2) : $d(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \left(\sum_j (x_{ij} - x_{i'j})^2 \right)^{1/2}$,
- *Manhattan distance* (L_1) : $d(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \sum_j |x_{ij} - x_{i'j}|$,
- 1− *Pearson correlation* ($1 - \rho$) : $d(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = 1 - \frac{\sum_j (x_{ij} - \bar{x}_{i\cdot})(x_{i'j} - \bar{x}_{i'\cdot})}{\sqrt{\sum_j (x_{ij} - \bar{x}_{i\cdot})^2 \sum_j (x_{i'j} - \bar{x}_{i'\cdot})^2}}$.

Using a specified dissimilarity function, clusters can be identified by finding the partition which minimizes the total within-cluster dissimilarity in the data. However, since the total within-cluster dissimilarity is highly dependent on the number of clusters, K, optimization must be carried out conditional on a fixed choice of K. Selection of the optimal K is a difficult but important problem in cluster analysis. Some popular methods for choosing K include the gap statistic (Tibshirani et al., 2001) and consensus clustering (Monti et al., 2003; Wilkerson and Hayes, 2010). Briefly, the gap statistic is a heuristic approach for determining the optimal K based on the decrease in within-cluster dissimilarity as K increases. The approach chooses the optimal number of clusters to be the largest K for which a substantial decrease in the total within-cluster dissimilarity is observed. Consensus clustering is similarly a heuristic approach which was proposed in the microarray analysis literature as a way of addressing the sensitivity of clustering algorithms to individual samples. The approach aims to identify robust, or stable, clusters in a dataset by aggregating the results of clustering on a large number of random subsamples drawn from the data. A closely related problem to choosing K is that of assessing the significance of the resulting clusters. Intuitively, the K which produces the most statistically significant set of clusters may be interpreted as the optimal number of clusters. Existing approaches for assessing significance of clusters include statistical Significance of Clustering (SigClust; Liu et al., 2008), a bootstrapping approach by Maitra et al. (2012), and `pvclust` (Suzuki and Shimodaira, 2006). These approaches, as well as new contributions to the area of significance of clustering, are described in more detail in Chapter 3.

Clustering algorithms carried out independently for each value of K will be referred to as *non-nested* approaches to clustering. Non-nested algorithms enforce no structure between the clusters produced by different values of K, and thus provide no intuitive way of studying relationships among clusters. In addition to non-nested approaches, there also exist hierarchical approaches to clustering. Unlike non-nested clustering, hierarchical clustering methods produce a nested hierarchy of clusters through the entire range of $K = 1, \ldots, n$. We next briefly review some non-nested algorithms and

describe the general framework for hierarchical clustering. A more complete treatment of these approaches can be found in Kaufman and Rousseeuw (2009) and Hartigan (1975).

1.2.1 Non-Nested Clustering

For the sake of brevity, we only focus on combinatorial approaches for non-nested clustering. Other examples of non-nested clustering include mixture modeling and mode seeking approaches. Combinatorial methods seek to find an explicit partition of the n observations to K clusters $C_1, \ldots, C_K$ optimal to an appropriate criterion, where C_k is the set of observation indices in cluster k. Letting $\mathcal{I} = \{1, \ldots, n\}$ denote the complete set of observation indices and $|\cdot|$ denote the cardinality of a set, for all $k \neq k'$, the clusters must satisfy:

- $C_k \subset \mathcal{I}$: they form a subset of the data,
- $\cup_k C_k = \mathcal{I}$: they jointly cover all observations,
- $C_k \cap C_{k'} = \emptyset$: they are disjoint,
- $|C_k| > 0$: they are non-empty.

Since combinatorial approaches specify neither a probabilistic model nor estimable parameters, solutions are given precisely by the partition of $\mathcal{I}$ to $C_1, \ldots, C_K$. These approaches are aptly named for the combinatorial nature of the resulting solution space, the size of which grows rapidly in n and K. As an example, the total number of partitions of 19 observations to 4 clusters is of the order 10^{10} (Section 14.3.5, Hastie et al., 2011). As such, exhaustive search for the optimal partition quickly becomes infeasible. To handle this problem, many algorithms rely on iterative greedy descent which only requires searching a small subset of the solution space. However, these algorithms can only guarantee local minima. It is therefore common practice to consider multiple starting points of the algorithm to avoid highly suboptimal solutions.

Given an appropriately chosen dissimilarity measure and partition $C_1, \ldots, C_K$, the *total dissimilarity*, $T(C)$, can be decomposed as the sum of the within-cluster dissimilarity and the between-cluster dissimilarity, denoted $W(C)$ and $B(C)$, as follows:

$$T(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{i=1}^{n} \sum_{i'=1}^{n} d(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$$

$$= \underbrace{\frac{1}{2}\sum_{k=1}^{K}\sum_{i\in C_k}\sum_{i'\in C_k} d(\boldsymbol{x}_i, \boldsymbol{x}_{i'})}_{W(C)} \quad + \quad \underbrace{\frac{1}{2}\sum_{k=1}^{K}\sum_{i\in C_k}\sum_{i'\notin C_k} d(\boldsymbol{x}_i, \boldsymbol{x}_{i'})}_{B(C)}. \tag{1.3}$$

Since $T(C)$ is independent of the cluster assignments, minimizing within-cluster dissimilarity and maximizing between-cluster dissimilarity are exactly equivalent. Many combinatorial approaches, including K-means and K-mediods, minimize some variant of the within-cluster dissimilarity $W(C)$ (MacQueen et al., 1967; Kaufman and Rousseeuw, 2009).

Briefly, the K-means algorithm seeks to minimize a cluster size-adjusted variant of $W(C)$ using squared Euclidean dissimilarity. Precisely, the K-means objective is given by:

$$W'(C) = \sum_{k=1}^{K}\sum_{i\in C_k} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k\|_2^2,$$

where $\bar{\boldsymbol{x}}_k = \frac{1}{n_k}\sum_{C_k} \boldsymbol{x}_i$ is the k-th cluster mean. Given a starting partition, the algorithm solves for an locally optimal partition by alternating between the following two steps until the clusters stabilize:

1. solve for $\bar{\boldsymbol{x}}_k$ to minimize $W'(C)$ given current partitions C_k,
2. solve for C_k to minimize $W'(C)$ given current means $\bar{\boldsymbol{x}}_k$.

At Step 2, new clusters are obtained by reassigning samples to the cluster with the closest mean. The K-mediods algorithm simply generalizes K-means to dissimilarity measures other than squared Euclidean distance. Although K-mediods is useful in many situations, the algorithm requires substantially more computational time than K-means.

1.2.2 Hierarchical Clustering

In contrast to non-nested approaches, hierarchical clustering does not require specifying K. Instead, the approach estimates all $K = 1, \ldots, N$ partitions of the data through a sequential optimization procedure. The sequence of steps can be implemented as either an agglomerative (bottom-up) or divisive (top-down) approach to produce the nested hierarchy of clusters. Agglomerative clustering begins with each observation belonging to one of n disjoint singleton clusters. Then, at each step, the two most similar clusters are joined to form a single cluster, until after $(n-1)$ steps all observations belong to a single cluster of size n. Divisive clustering proceeds in a similar, but

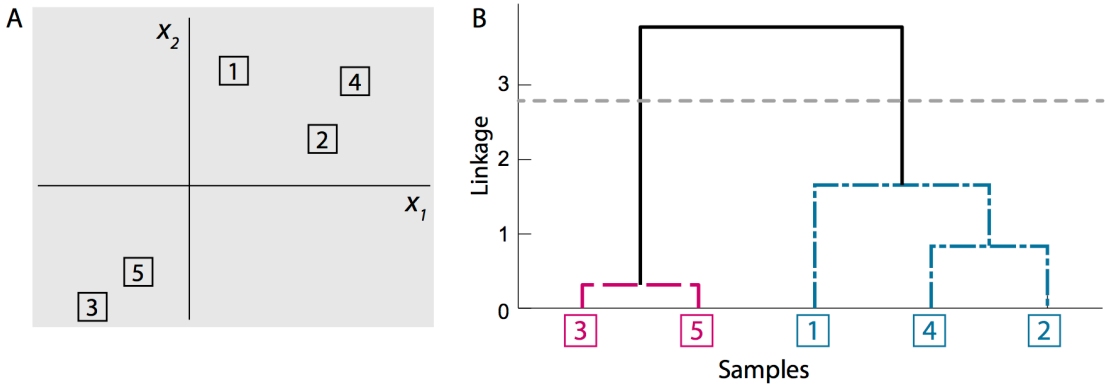

Figure 1.4: Hierarchical clustering with squared Euclidean dissimilarity and average linkage is applied to 5 observations of the bivariate variable, $\mathbf{x}$. (A) The actual values of each observation are shown in $\mathbb{R}^2$. (B) The resulting dendrogram is shown with sample indices placed along the horizontal axis and sequentially connected until all observations are joined at the top of the tree. The vertical axis corresponds to cluster similarity, such that clusters joined lower along the dendrogram are most similar.

reversed manner, in which a single cluster containing all n observations is sequentially split until after $(n-1)$ steps, each observation belongs to a separate singleton cluster.

To determine which clusters to join at each step of the agglomerative procedure, a linkage function is used to extend the definition of pairwise object dissimilarity to clusters. Let $C_k \subset \mathcal{I} = \{1, \ldots, N\}$ denote the set of observation indices belonging to cluster k. Then, given a dissimilarity function $d : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}^+$, we similarly denote the linkage function by $d : \mathcal{I} \times \mathcal{I} \to \mathbb{R}^+$. Often, linkage is defined as some function of the pairwise dissimilarities of observations belonging to the two clusters. Examples of linkage functions include:

- *Ward's*: $d_W(C_k, C_{k'}) = \frac{2|C_k||C_{k'}|}{|C_k|+|C_{k'}|}\|\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}}_{k'}\|^2$,
- *single*: $d_S(C_k, C_{k'}) = \min\{d(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) : i \in C_k, i' \in C_{k'}\}$,
- *complete*: $d_C(C_k, C_{k'}) = \max\{d(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) : i \in C_k, i' \in C_{k'}\}$,
- *average*: $d_A(C_k, C_{k'}) = \frac{\sum_{i \in C_k}\sum_{i' \in C_{k'}} d(\boldsymbol{x}_i, \boldsymbol{x}_{i'})}{|C_k||C_{k'}|}$.

Note that Ward's linkage is specifically defined using squared Euclidean dissimilarity while all other linkages are defined for general dissimilarity functions (Ward, 1963). As with the choice of dissimilarity measure, the resulting clusters greatly depend on the chosen linkage function.

Since the work of Eisen et al. (1998), agglomerative hierarchical clustering algorithms have enjoyed substantial popularity in the analysis of microarray expression data. In several landmark papers that followed, these methods were successfully used to identify clinically relevant expression

subtypes of lymphoma, breast, and other types of cancer (Alizadeh et al., 2000; Perou et al., 2000; Bhattacharjee et al., 2001). The popularity of hierarchical clustering in practice may be largely attributed to *dendrograms*, a highly useful and informative visualization of clustering results. Specifically, dendrograms represent the results of hierarchical clustering as a binary tree where clusters are connected at a height corresponding to the value of the objective function at which the joining or splitting occurred. We next give a simple example of a dendrogram and describe how agglomerative and divisive approaches determine which clusters to join or split at each step.

In Figure 1.4B we show the dendrogram for five realizations of the bivariate random variable, $\mathbf{X}$, clustered using squared Euclidean dissimilarity and average linkage. The actual values for the five observations lying in $\mathbb{R}^2$ are given in Figure 1.4A. In Figure 1.4B, the observation indices are placed evenly along the horizontal axis, such that no two branches of the dendrogram cross. Note that as a result, several equivalent arrangements of the same dendrogram are possible by flipping the orientation of branches. The sequential clustering procedure is shown by the joining of clusters at their respective linkage value, represented by the vertical axis of Figure 1.4B. As such, the most similar clusters and observations are connected near the bottom of the tree. The spectrum of clustering solutions can be recovered from the dendrogram by cutting the tree at an appropriate height, and taking the resulting subtrees as the clustering solution. For example, the corresponding $K = 2$ solution is obtained by cutting the dendrogram at the gray horizontal line in Figure 1.4B, resulting in the red dashed and blue dot-dashed subtrees.

While less popular, divisive approaches to clustering also exist. Unlike agglomerative approaches, they do not require a linkage function. Instead, they require explicitly defining a splitting rule, which is then recursively applied to obtain a complete partition of the data. While rules specific to divisive clustering have been proposed, e.g. Macnaughton-Smith et al. (1964), combinatorial algorithms, such as K-means for two clusters, may also be used. At each step, a single cluster is chosen based on a heuristic, such as largest average within-cluster dissimilarity, and split to return two smaller clusters.

Figure 1.5: The central dogma of molecular biology.

1.3 RNA-seq Data

In this section, we provide a brief introduction to the analysis of RNA-seq data. We first present an overview of RNA-seq technologies, and then describe some popular statistical methods which have been developed for expression analysis using RNA-seq data.

Over the past two decades, microarrays have been the workhorses for data collection in cancer genomics. However, recent advances in sequencing technology have given rise to high-throughput *second-generation* methods, including RNA-seq (Wang et al., 2009; Metzker, 2010). In contrast to microarrays, which typically measure relative expression levels at the per-gene level, sequencing data provide integer counts which quantify expression at each base position. For this reason, RNA-seq is often referred to as a *digital* measurement of expression. With RNA-seq data, it is now possible to study the collection of all mRNAs in a sample at higher accuracy and resolution than before, fundamentally transforming genomic research in cancer as well as other diseases (Meyerson et al., 2010).

To help contextualize RNA-seq, we first introduce the *central dogma of molecular biology*, which summarizes the general process of gene expression within a cell (Figure 1.5). The human genome contains approximately 3 billion base pairs (bp) of DNA, often represented as a directed sequence of 'A', 'T', 'C', and 'G's. During transcription, a relatively short segment of DNA is copied into pre-cursor messenger RNA (pre-mRNA). Following transcription, pre-mRNA molecules are processed to produce mature mRNAs for translation. Each pre-mRNA molecule is comprised of protein coding and non-coding regions, called exons and introns, respectively. During the processing step, introns are removed from the pre-mRNA molecule, and the remaining exons are selective joined, or *spliced*, together to form a variety of mRNA sequences (Figure 1.5A). The process which results in several distinct mRNAs, or *isoforms*, from a single DNA template is called *alternative splicing*, and is a major source of protein diversity in vertebrates (Maniatis and Tasic, 2002). Three examples of alternative splicing are shown in Figure 1.5A, including: cassette exon inclusion/exclusion, alter-

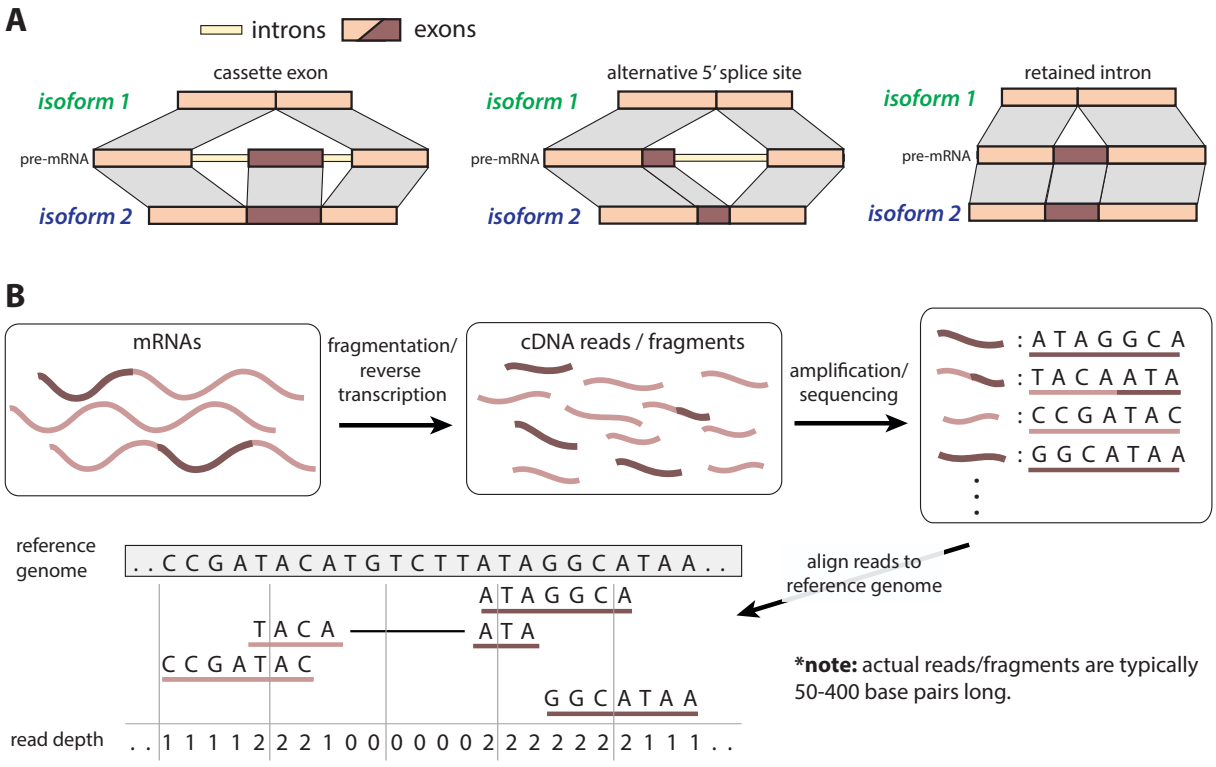

Figure 1.6: (A) Three modes of mRNA alternative splicing leading to multiple isoforms from a single gene locus: cassette alternative exon, alternative 5′ splice site, and intron retention. (B) A typical RNA-seq experiment given a collection of mRNAs from a biological sample, e.g. tumor sample.

native 5′ splice-site selection, and intron retention (Feng et al., 2012). Finally, during translation, proteins are synthesized from the mRNA strands by ribosomes within the cell. These proteins then proceed to serve various roles within the body. As such, it is often of interest to quantify the total production, or expression, of these proteins within different biological samples or states. However, since the direct quantification of protein products can be difficult, more commonly, the abundance of mRNA sequences which encode the proteins are measured as a surrogate to quantify gene or isoform level expression. Microarrays, and more recently, RNA-seq, are examples of technologies which have been developed to accomplish this task.

The workflow for a typical RNA-seq experiment is shown in Figure 1.6B (Li et al., 2012). First, mRNAs collected from a biological sample are randomly fragmented into short *reads* or *fragments*, which are then reverse-transcribed to complementary DNA (cDNA). Next, the collection of cDNA reads/fragments are PCR amplified and sequenced using high-throughput methods. Popular sequencing platforms include Illumina, Roche 454 and Life Technologies. The sequenced reads or

fragments are collected and typically stored as FASTA or FASTQ files. At this point, the RNA-seq data generation and collection process has been completed. However, prior to statistical analysis, the data is usually further processed using bioinformatics tools and methods. Most often, this involves mapping the sequenced short reads back to a reference genome using an alignment algorithm such as TopHat2 (Kim et al., 2013) or MapSplice (Wang et al., 2010). Then, the number of aligned reads at any position may be used to infer the expression at that locus for further analysis. Note that in Figure 1.6B, a single read ('TACAATA') is aligned with space separating the first four and final three positions. As a result of pre-mRNA splicing, reads which span exon-exon junctions, called *junction reads*, map to the genome with a gap corresponding to a spliced intronic region of the genome. Junction reads are particularly important, as they provide direct insight in to the alternative splicing patterns present in a sample. Additionally, while for illustrative purposes each read in Figure 1.6B was only 7bp, in practice, reads are typically 50bp to several hundred bp long.

The development of RNA-seq has made it possible to not only quantify gene expression at higher accuracy, but also qualify the variety of isoforms being expressed in a sample (Ozsolak and Milos, 2011). As a result, recent genomic studies using RNA-seq are beginning to shed light on the sheer prevalence of post-transcriptional events, such as alternative splicing, across the human genome (Nilsen and Graveley, 2010).

A large number of statistical methods have been developed for the analysis of RNA-seq data. However, in this review, we only focus on the subset of approaches implemented for differential expression analysis and expression-based clustering. First, the goal of differential expression methods is to identify genomic regions at which expression is significantly associated with an outcome or known stratification across a collection of samples. Some early examples of gene level differential expression methods developed for RNA-seq data include DEseq (Anders and Huber, 2010) and `edgeR` (Robinson et al., 2010). Growing evidence supporting the importance and abundance of alternative splicing has lead to the development of more complex differential expression methods for RNA-seq data. Examples of these methods include approaches for detecting differential expression at the exon and isoform level, such as DEXSeq (Anders et al., 2012) and Cuffdiff2 (Trapnell et al., 2013). Since expression in RNA-seq data is approximated using the number of reads aligned at a given position, many of these methods use Poisson or Negative-Binomial distributions to model gene, isoform or exon level expression.

In addition to the differential expression approaches described above, some work has been done to develop unsupervised approaches for RNA-seq data. Due to the highly heterogeneous nature of cancers, there has been great interest in the identification of subtypes of cancers using genomic data for improving prognoses and developing targeted therapies. As mentioned in Section 1.2, over the past decade, hierarchical clustering has been successfully used to identify clinically relevant subtypes of cancers from microarray data. Notably, Witten (2011) proposed an extension of the hierarchical approaches used in microarray analyses for RNA-seq data. In contrast to the vast literature on supervised approaches for studying isoform and exon level differences in RNA-seq data, the literature on unsupervised methods for problems such as subclass discovery is sparse, leaving much room for development. The help fill this gap, in Chapter 2 we introduce SigFuge, a method for identifying clusters of samples with differential isoform usage.

1.4 Outline

In this dissertation we consider problems in both supervised and unsupervised learning with particular interest in applications to next-generation sequencing analysis. These include a method for clustering in RNA-seq data, an extension of the SigClust methodology to hierarchical clustering, and a unified framework for hard and soft classification.

First, in Chapter 2 we consider the problem of unsupervised learning in RNA-seq data. We introduce SigFuge, an approach to clustering RNA-seq samples using per-base expression. The effectiveness of our approach is shown through simulations and applications of SigFuge to two cancer datasets obtained from The Cancer Genome Atlas (TCGA) Research Network. An `R` implementation of SigFuge has been made available through Bioconductor.

In Chapter 3 we extend the SigClust methodology to hierarchical clustering. In the original SigClust manuscript, Liu et al. (2008) proposed a divisive approach to applying SigClust based on iterative 2-means K-means splits. In this chapter, we describe a more general extension of the SigClust approach for testing along a dendrogram. The approach is implemented as a sequential testing procedure guaranteeing control of the family-wise error rate. Theoretical justification is provided for our approach, and its power to detect true clustering structure is illustrated through several simulation studies and applications to two cancer gene expression datasets. Additionally,

we describe a modification of the current approach for estimating the parameters of the SigClust null distribution.

Finally, in Chapter 4 we propose a novel framework for connecting soft and hard classification. Other frameworks, such as the LUM family (Liu et al., 2011) have helped to shed light on the different behavior of hard and soft classifiers. However, these approaches relate the two problems through specific classes of loss functions. We argue that a more natural approach is to relate soft and hard classification as particular cases of a family of binary learning problems. We study the problems using the framework of large-margin classifiers and propose a class of piecewise linear convex surrogates for which we derive statistical properties and a corresponding sub-gradient descent algorithm.

CHAPTER 2
SigFuge for Discovery in RNA-seq Data

2.1 Introduction

Today, massively parallel next-generation sequencing platforms offer unbiased analysis of transcriptomes at higher accuracy and resolution than microarrays (Marioni et al., 2008). Beyond measuring expression levels, transcriptome sequencing (RNA-seq) can be used to discover novel transcriptional events such as splicing patterns (Sultan et al., 2008), alternative untranslated region (UTR) usage (Ramsköld et al., 2009), and gene fusions (Maher et al., 2009). With the rise of platforms capable of producing large-scale genomic datasets, unsupervised methods have played an increasingly major role in the analysis of such data. Arguably, among unsupervised approaches, clustering methods have had the most visible impact on the field. In past studies, hierarchical clustering has been applied to microarray expression data to identify clinically relevant subclasses of cancers and other diseases (Eisen et al., 1998; Perou et al., 2000; Hayes et al., 2006). As such, extensions of these approaches to modern sequencing platforms could potentially be used to identify unrecognized structure with applications to a variety of problems.

An emerging area of genomic research is the identification of alternative splicing events, i.e. when pre-mRNAs are spliced in different ways to produce distinct isoforms, ultimately encoding for different proteins (Maniatis and Tasic, 2002). Recent estimates suggest that most human genes are alternatively spliced, with most alternative exons showing tissue-specific regulation (Wang and Cooper, 2007). Further, alternative splicing and isoform selection have been implicated as determinants of cell type and specificity (Pan et al., 2008). Within individual samples, multiple isoforms are often simultaneously expressed at a single gene. Therefore, identifying differential isoform usage, where multiple isoforms of a single gene are expressed, but at different proportions between groups of samples, may provide insight into the functional consequences of a disease.

Throughout this chapter, we will refer to a region of the genome to which a single gene has been annotated as a *locus*. Using RNA-seq, expression at a single gene, or locus, can now be measured at each base-position along the length of the transcript, making the technology sensitive to isoform level changes in expression. Thus, genome-wide discovery of alternate isoform usage is an opportunity afforded by RNA-seq, beyond what was possible using gene expression arrays. The SigFuge (SIGnificant Forms Using per-base Gene Expression) approach introduced in this chapter is motivated by the desire to realize the full potential of RNA-seq data.

Several methods have been suggested for the detection of alternative splicing or isoform differences in supervised settings, e.g. in a tumor vs. normal comparison, including Cuffdiff2 (Trapnell et al., 2013), DEXSeq (Anders et al., 2012), and DiffSplice (Hu et al., 2013). However, differences in isoform usage may not always correspond to known class labels, e.g. differential usage may exist between subsets of a single tissue type. As an example, the significant expression of a novel *CDH3* splice variant was reported in only a subset (8/20) of adenocarcinoma tumors relative to normal (Xi et al., 2008). In this case, the differential signal may become lost within the larger tumor vs. normal comparison, and further, the subtype behavior completely missed. Using existing approaches, it is not clear how to identify differential isoform usage when the appropriate stratification of samples is unknown.

To address these problems, unsupervised approaches, including clustering, have complemented supervised analyses in genomics. Earlier on, approaches to whole-genome clustering, i.e. clustering by gene expression across all loci, were proposed for RNA-seq data (Witten, 2011). More recently, SIBER (Tong et al., 2013) and DEXUS (Klambauer et al., 2013) have been proposed for clustering samples at the single gene level, i.e. clustering at each gene separately, to discover novel subpopulations exhibiting differential expression at individual loci. However, these methods were not specifically designed to detect differences in isoform usage as they only consider gene-level expression.

In order to detect subsets, or clusters, of RNA-seq samples with alternative forms or patterns of isoform usage, we have developed SigFuge. SigFuge aims to identify clusters which express isoforms from a single gene locus at differing proportions. That is, we seek to identify clusters with differing isoform preferences at the level of single genes. This is possible because SigFuge uses expression levels at each base-position across a gene locus. Briefly, for each locus, the approach

first requires filtering out lowly expressed samples. Then, among the remaining samples, SigFuge normalizes expression at the base-pair level. This normalization allows SigFuge to emphasize expression differences occurring throughout a segment of the gene, e.g. exon-level differences, while ignoring differences occurring across the entire gene, e.g. whole gene gain/loss, which methods such as SIBER and DEXUS aim to identify. Next, the samples are clustered into two subpopulations by the normalized base-pair level expression, and finally, a significance test is performed to quantify the strength of evidence supporting a difference in isoform usage between the two subpopulations. SigFuge is available as an `R` package through Bioconductor.

In Section 2.2, we first describe SigFuge using a simple toy example. We then compare the performance of the method against the closest competing approaches, DEXUS and SIBER, through an extensive simulation study in Section 2.3. In Section 2.4, we apply the method to collections of lung squamous cell carcinoma (LUSC) and head and neck squamous cell carcinoma (HNSC) RNA-seq samples from The Cancer Genome Atlas (TCGA). We show that SigFuge identifies important transcriptional alterations including alternative splicing of the tumor suppressor gene *CDKN2A*. Finally, we conclude with a discussion in Section 2.5.

2.2 Methodology

We describe the SigFuge method in three major parts: data extraction, processing and analysis. A pipeline of the complete approach is given in Figure 2.1A, with blue boxes used to distinguish the three parts. In the next subsections, we describe each part in detail, motivating our approach using a hypothetical *Gene A* across a cohort of 60 RNA-seq samples. To replicate true variation observed in RNA-seq data, the toy dataset was generated using counts obtained from 60 of the LUSC samples along a subset of the bases within the *FAM64A* locus.

2.2.1 Data Extraction

Consider *Gene A* having two known isoforms differing by a single cassette (middle) exon (Figure 2.1B). We first study three samples which represent important modes of expression in the larger cohort of 60 samples:

1. low expression across the entire gene,

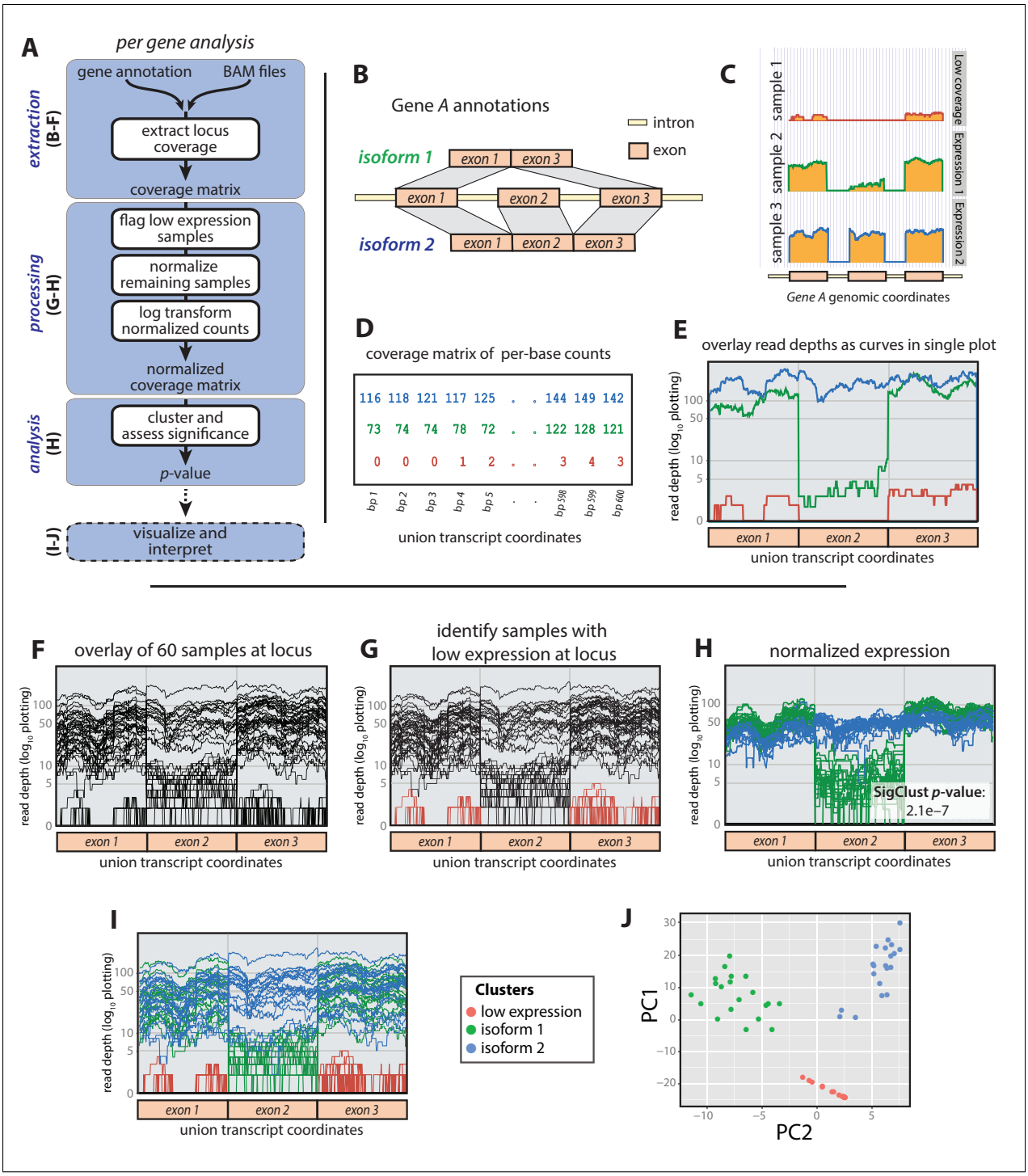

Figure 2.1: The SigFuge approach is illustrated through a hypothetical example *Gene A* with two true isoforms differing by a single cassette exon. (A) A general outline is given for the complete SigFuge pipeline. (B) The gene model includes two isoforms. (C) Read count pile-ups for samples 1, 2 and 3 show low expression (top, red), expression of isoform 1 (middle, green) and expression of isoform 2 (bottom, blue), respectively. (D) The expression curves are analyzed as a (sample $\times$ base) matrix representation of read counts. (E) The SigFuge approach studies these expression profiles as curves along the union gene model of the locus. (F) We consider a collection of 60 unlabeled samples at *Gene A*. (G) To study expression patterns, lowly expressed samples (colored red) are first excluded. (H) The remaining samples are normalized, log-transformed and clustered using K-means clustering. The clusters are clearly visible in the log-transformed raw curve space (I) as well as the principal components space (J).

2. primary expression of isoform 1,

3. primary expression of isoform 2.

The differences in mRNA product are clearly reflected in the corresponding per-base read depths, plotted on the log-scale (Figure 2.1C). From this, we propose characterizing gene expression at the per-base resolution to study differential isoform usage. More specifically, we define a sample expression profile at a given gene to be the vector of per-base read depths across the exons of the union gene. In our toy example, the union gene, formed by combining all isoforms, is simply isoform 2 (Figure 2.1B). The resulting data structure is an expression count matrix with rows corresponding to samples, and columns corresponding to positions across the gene model (Figure 2.1D). This count matrix serves as the input to the computational steps of SigFuge, and can be obtained, for example, from BAM files using the `samtools` command line package (Li et al., 2009). While read depth is commonly plotted using separate panels for each sample (Figure 2.1C), we prefer a more compact visualization where expression profiles are overlaid as curves in a single figure (Figure 2.1E). Note that the empty regions of Figure 2.1C corresponding to introns are excluded from the expression profiles of Figure 2.1E.

While exon annotations are not explicitly required for SigFuge, as each base-position is treated with equal weight, their use leads to more naturally interpretable results by restricting attention to expressed regions of the locus, e.g. in obtaining Figure 2.1E from Figure 2.1C. Similarly, our approach does not require information about the possible isoforms composing a gene model. This is a major strength of the method, as the precise structure of isoforms may be unknown.

2.2.2 Data Processing

In Figure 2.1F-J, we consider the complete collection of 60 expression curves (samples) along *Gene A*. The goal of SigFuge is to determine whether the sample-set contains subgroups, or clusters, exhibiting different isoform usage. An example of differential isoform usage would be a subset of samples which only express isoform 1 while all remaining samples express both isoforms 1 and 2 in equal proportions. Since we are not interested in whole-gene changes in expression, we first perform a normalization of the sample curves to make identifying clusters of differential isoform usage easier. The normalization procedure is broken into the three following steps:

1. **Filtering**: removing low-expression samples,

2. **Count Normalizing**: scaling expression curves to have equal total coverage,

3. **Log-Transformation**: mapping count data, which vary by orders of magnitude, to a more natural scale.

Since our interest is in differential isoform usage, samples not expressing any isoforms of the gene are first removed from the analysis, forming a separate cluster of low-expression samples. Specifically, at each locus we exclude samples with over 90% of base positions having zero coverage, as well as samples with median coverage of less than 5 reads across covered positions. In the toy example, low-expression samples are colored red in Figure 2.1G and completely removed from Figure 2.1H. Next, count normalization is used to remove differences in overall gene expression between samples. To do this, each remaining sample is scaled by its total expression across the gene, i.e. by the corresponding row sum of the count matrix. While several approaches have been proposed for the normalization of RNA-seq expression data (Mortazavi et al., 2008; Bullard et al., 2010; Robinson and Oshlack, 2010), these approaches were developed for genome-wide normalization, with the goal of identifying differentially expressed genes. In contrast, we aim to identify isoform imbalances at individual gene loci by identifying curves exhibiting different shapes, regardless of overall expression. Therefore, rather than employing normalization procedures for genome-wide differences (e.g. library sizes), we instead use a simple per-gene approach. Note that this normalization procedure assumes that each locus contains a single gene and may not be appropriate for loci containing multiple genes. Lastly, the scaled count data are log-transformed (Figure 2.1H). Log-transformation is used to study counts on the scale of relative expression, and is often applied when data vary over several orders of magnitude, as with read counts. To ensure the log is always well defined, i.e. to handle zero counts, all scaled values are increased by 1 prior to transformation. Note that zero counts remain at zero after transformation. As shown in Figure 2.1H, normalization reduces sample variability across most of the gene (exons 1 and 3), and highlights regions with non-uniform usage across samples (exon 2).

2.2.3 Data Analysis

Following normalization, K-means clustering (MacQueen et al., 1967) for two clusters ($K = 2$) using Euclidean distance is applied to the normalized samples to identify clusters corresponding to differential isoform usage. In Figure 2.1H these clusters, colored blue and green, differ noticeably by their use of exon 2. Here, SigFuge accurately captures clusters of samples with differing preferences for isoforms 1 and 2.

Translating the cluster labels to the original expression profiles, i.e. coloring the data by clusters, verifies that the identified clusters indeed correspond to clear differential patterns across *Gene A* (Figure 2.1I). To emphasize the notion of isoform clusters, we visualize the toy example using principal component analysis (PCA; Jolliffe, 2002), an exploratory analysis tool for identifying low-dimensional structure in high-dimensional data (Figure 2.1J). The log-transformed raw data are projected along the first two principal component directions and colored according to the results of SigFuge. The plot clearly reveals three distinct clusters, showing the protocol accurately captures the main modes of variation among the samples.

This toy example was generated such that the clusters represent clearly differential patterns of expression. However, often loci considered in practice will only possess a single expression pattern. This may correspond to loci with a single dominant isoform or expression of multiple isoforms in similar proportions across all samples. As an exploratory tool, K-means identifies clusters regardless of whether they represent true underlying structure. An important, yet difficult task in cluster analysis is to distinguish natural clustering from artificial clustering generated by the chosen algorithm. In the present context, this corresponds to identifying the small subset of genes with clusters exhibiting true differential isoform usage within the large number of genes across the entire transcriptome. To address this issue, SigFuge calculates a p-value quantifying the statistical significance of clustering at each locus. The SigFuge p-values can then be used to order a large set of genes to identify a subset of loci most likely to exhibit true differential isoform usage.

The p-value calculation is carried out using SigClust (Liu et al., 2008). A thorough description of the approach is provided in Section 3.2.2. SigClust is implemented using the sample covariance matrix estimate of the null Gaussian distribution. Extensive simulation study has shown that among several proposed SigClust null covariance estimators, the sample covariance approach

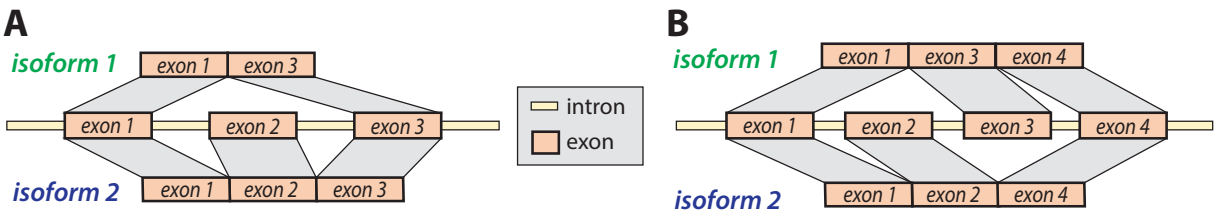

Figure 2.2: Two gene models used in the simulation settings. (A) Three exon gene model containing a cassette exon (exon 2) spliced out from isoform 1 and only retained in isoform 2. (B) Four exon gene model containing mutually exclusive cassette exons (exons 2, 3).

produces consistently conservative p-values (Huang et al., 2014). In our genome-wide analysis we restrict the number of simulations to 100 for each gene. We then fit a Gaussian distribution to the 100 observed null 2-means CIs and report the lower tail probability of this fitted Gaussian as our approximate p-value, as described in (Liu et al., 2008). Although these p-values are not exact, they give a good sense of the relative significance of genes which otherwise report equivalent empirical p-values of 0. While SigFuge may be used to test for the statistical significance of clusters obtained using any algorithm, K-means clustering is used as it has favorable properties for the SigClust testing procedure, as noted in (Liu et al., 2008). For the clustering shown in Figure 2.1H, SigFuge reports a significant p-value of 2.1×10^{-7}. A more in-depth discussion of the SigClust assumptions as they pertain to our application can be found in Supplementary Methods S1 and Supplementary Figure S1. Finally, our analysis is restricted to $K = 2$ clusters as the SigClust methodology is currently only capable of testing for statistical significance with two subgroups. However, we note that other existing unsupervised approaches, such as SIBER and DEXUS, also seek a binary partition of observations, the later separating between a single major condition and all remaining minor conditions.

When applying SigFuge to a large number of genes, we suggest using an appropriate statistical procedure for controlling either the family-wise error rate (FWER) or false discovery rate (FDR). We use the Benjamini-Hochberg step-up procedure to control the FDR (Benjamini and Hochberg, 1995).

2.3 Simulations

An extensive simulation study was carried out at the level of single gene loci for varying experimental conditions. Datasets were simulated with several values of sample size, depth, dispersion and underlying isoform structure. For each simulation, per-base expression profiles were generated from an underlying gene model encoding two isoforms. Toy diagrams for the two gene models used in the simulations are shown in Figure 2.2. These include a three-exon gene model containing a cassette exon (Figure 2.2A), and a four-exon gene model containing alternate cassette exons (Figure 2.2B). Expression profiles along a single gene locus were generated from two subpopulations differing only by their isoform preferences. That is, samples from the two subpopulations were simulated with the same expected gene-level expression, but with differing expected isoform-level expressions. The null setting with no differential behavior was also considered by setting the isoform preferences to be equal between the two subpopulations.

More explicitly, let k denote the subpopulation index. Then, for $k \in \{1, 2\}$ let $\boldsymbol{\psi}_k = (\psi_{k1}, \psi_{k2})$ denote the corresponding isoform preference of samples in that subpopulation. Without loss of generality, assume $\psi_{k1}, \psi_{k2} \geq 0$ and $\psi_{k1} + \psi_{k2} = 1$. For example, if $\boldsymbol{\psi}_1 = (0, 1)$, then samples from subpopulation 1 will only express isoform 2, and if $\boldsymbol{\psi}_2 = (.5, .5)$, then samples from subpopulation 2 will express isoforms 1 and 2 equally. Five combinations of $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2$ are given in Table 2.1. Setting 1 corresponds to the null setting in which isoform preferences are equal between the two subpopulations ($\boldsymbol{\psi}_1 = \boldsymbol{\psi}_2$).

Under each of the settings described in Table 2.1, various simulations were performed to assess the performance of the methods across different levels of mean read depth ($\mu \in \{50, 100, 500\}$), dispersion ($\phi \in \{0.087, 0.179, 0.369\}$), gene length ($d \in \{1200, 2400\}$), and subpopulation sizes ($(n_1, n_2) \in \{(10, 10), (50, 50), (50, 1), (75, 25), (100, 100)\}$). Candidate values for ϕ were chosen

Table 2.1: Differential expression settings used in SigFuge simulation study.

Setting	**Gene Model**	$\boldsymbol{\psi}_1$	$\boldsymbol{\psi}_2$
1	Three-exon	(0.50, 0.50)	(0.50, 0.50)
2	Three-exon	(0.25, 0.75)	(0.75, 0.25)
3	Four-exon	(0.25, 0.75)	(0.75, 0.25)
4	Three-exon	(0.67, 0.33)	(0.33, 0.67)
5	Four-exon	(0.67, 0.33)	(0.33, 0.67)

as the quartiles of a lognormal distribution estimated in Wu et al. (2013) for the Gilad dataset (Blekhman et al., 2010).

For a fixed simulation setting and values of μ, ϕ, d, n_1, n_2, a single gene dataset is simulated as follows. First, to generate subpopulation 1, $2 \cdot n_1$ isoform level expression values are simulated from Negative Binomial distributions with means $\mu \cdot \psi_{11}$ and $\mu \cdot \psi_{12}$, with dispersion parameter ϕ. Similarly, $2 \cdot n_2$ isoform level expression values are simulated from Negative Binomial distributions with means $\mu \cdot \psi_{21}$ and $\mu \cdot \psi_{22}$, with dispersion parameter ϕ. Then, to mimic the short-fragment read sequencing process of RNA-seq data, for each sample and each isoform, 50bp "reads" are generated randomly (uniformly) across the corresponding isoform model to achieve the necessary isoform level expression. Finally, at each position along the gene, the number of aligned reads is counted to produce the per-base level expression profiles passed to SigFuge. Examples of datasets simulated for the 5 simulation settings are shown in Figure 2.3. Each dataset was simulated with parameters: $\mu = 100$, $\phi = 0.179$, $d = 1200$, $n_1 = 50$, $n_2 = 50$. The resulting coverage profiles appear similar to what is observed in real RNA-seq data.

Each simulated single gene dataset was analyzed using SigFuge, DEXUS and SIBER. While DEXUS and SIBER were originally described for gene-level analysis, to make the approaches more comparable to SigFuge, both methods were applied to read counts at the following three levels of aggregation: (1) whole gene, (2) exon, and (3) disjoint 100bp windows. DEXUS results were called significant according to the default informative/noninformative (I/NI) value threshold of the accompanying `R` implementation. Significance for the results of SIBER were determined based on a bimodality index (BI) cutoff described in Table 1 of Tong et al. (2013) for controlling FDR at 0.05. Results for SIBER are not reported for total sample sizes less than 50, as BI cutoffs were only provided for sample sizes of 50, 100, 200 and 300. Since no clear approach exists for aggregating across multiple tests with the respective I/NI and BI output of DEXUS and SIBER, for the exon and 100bp window implementations of these two methods, loci were determined to be significant if any exon or window was called significant with no correction for multiple testing. SigFuge results were called significant at a p-value cutoff of 0.05.

In addition to single gene simulations, a joint simulation of 10,000 genes was also considered, including 9,000 null genes with no subpopulation behavior and 1,000 non-null genes with varying levels of differential usage across 100 samples. Of the 9,000 null genes, 4,500 were simulated

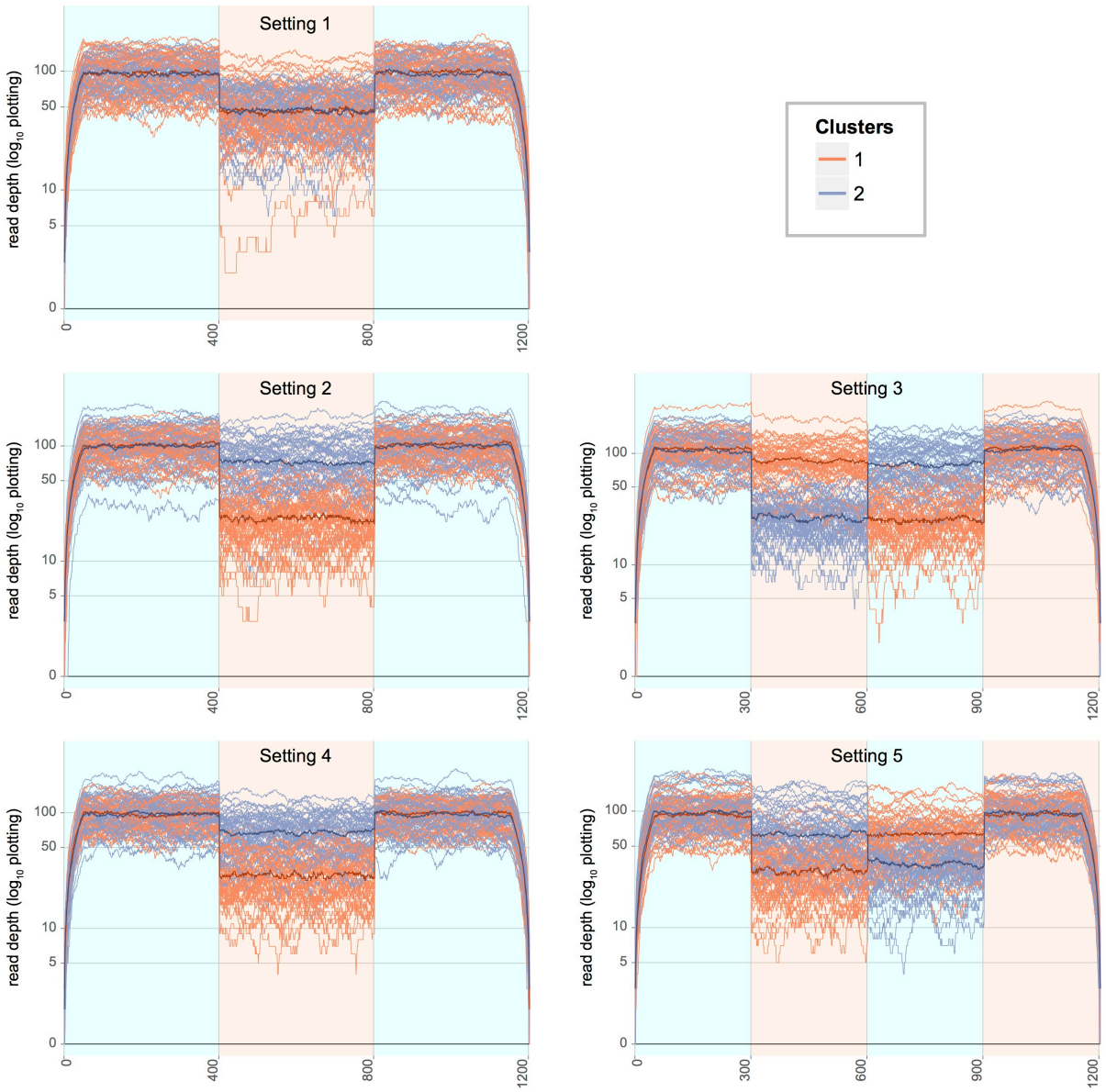

Figure 2.3: Log-transformed expression plots for 5 simulation settings. For each simulated gene locus, the per-base coverages were simulated from two underlying populations exhibiting differential isoform usage, denoted used by red and blue. Each dataset plotted was simulated with mean coverage (μ) 100, over-dispersion (ϕ) 0.179, gene length (d) 1200bp, and (n_1, n_2) 50 samples in each class. Population medians are shown in darker lines.

Table 2.2: Results for select simulation settings, with parameters: n_1, n_2 (subpopulation sample sizes), d (gene length), μ (gene-level read depth), ϕ (isoform-level dispersion). Parameter values $n_1, n_2 = 50, 50$; $d = 1200$; $\mu = 100$; $\phi = 0.179$ are treated as baseline, and deviations are marked by underlined values. For each setting, the numbers of significant calls out of 100 replications are reported for the default implementation of SigFuge, and DEXUS and SIBER at the gene, exon, and 100bp levels of aggregation. The mean (standard deviation) runtimes for single replications are reported in seconds for SigFuge, and DEXUS and SIBER at the 100bp level. Occasionally, NAs were reported in the output of SIBER. In this case, we mark the output with an asterisk (^{*m}) and report the number of significant calls out of $m < 100$ simulations. For non-null simulations (settings 2, 3), the method with highest sensitivity is highlighted in bold.

simulation parameters					**SigFuge**		**DEXUS**				**SIBER**			
setting	n_1, n_2	d	$\boldsymbol{\mu}$	$\boldsymbol{\phi}$	**bp**	*runtime*	**gene**	**exon**	**100bp**	*runtime*	**gene**	**exon**	**100bp**	*runtime*
1	100	1200	100	0.179	2	6.82	0	0	1	0.58	5	7	14	1.46
1	100	1200	500	0.369	7	6.59	0	2	2	0.58	2	5	9	1.55
1	20	1200	100	0.179	1	3.29	17	48	79	0.16	–	–	–	–
2	50, 50	1200	100	0.179	**89**	6.77	0	13	21	0.60	1	11	25	1.51
2	75, 25	1200	100	0.179	**98**	6.86	0	10	17	0.60	2	16	21	1.47
2	10, 10	1200	100	0.179	38	3.24	23	76	**86**	0.17	–	–	–	–
2	100, 100	1200	100	0.179	**98**	11.5	0	0	0	1.16	0	9	18	2.51
2	50, 50	1200	100	0.087	**100**	6.71	0	47	52	0.41	3	86	91	1.08
2	50, 50	1200	100	0.369	**62**	6.66	0	10	16	0.40	2	3	4^{*99}	0.94
3	50, 50	1200	100	0.179	**99**	6.82	0	23	33	0.62	3	26	38	1.50
3	75, 25	1200	100	0.179	60	6.80	0	56	**72**	0.61	1	13	15	1.40
3	10, 10	1200	100	0.179	29	3.25	14	93	**96**	0.17	–	–	–	–

according to Setting 1, with the remaining 4,500 as in Setting 1, except with a four-exon gene model. The 1,000 non-null genes were simulated from Settings 2, 3, 4 or 5 with equal probability. For each gene, (n_1, n_2) was randomly selected form $\{(10, 90), (75, 25), (50, 50)\}$. The remaining gene-level parameters (μ, ϕ, d, n_1, n_2) were randomly selected for each gene from the candidate values presented above. Each simulated gene was again analyzed using SigFuge, as well as DEXUS and SIBER applied at the three levels of aggregation. The SigFuge p-value, DEXUS I/NI and SIBER BI output were recorded for each simulated gene. As above, for the exon and 100bp window implementations of DEXUS and SIBER, the maximum I/NI and BI output were used to aggregate across the multiple tests.

Results of representative simulations from the single gene study are presented in Table 2.2. The table includes results for three simulation settings: (1) no differential behavior, (2) differential usage with a three-exon gene model, and (3) differential usage with a four-exon gene model. For each setting and combination of simulation parameters we report the number of significant calls over 100 replications of a single gene dataset.

2.3.1 Setting 1

We first considered the null setting with no subpopulation differences. For all parameter values considered, SigFuge made only the expected number of false positive calls at the 0.05 significance level. Similarly, SIBER and DEXUS make few false positive calls for the larger sample size ($n = 100$). However, when the sample size was decreased ($n = 20$), DEXUS produced a large number of false significant calls across all levels of aggregation. With both DEXUS and SIBER, more false significant calls were observed at finer levels of aggregation, i.e. using exon and 100bp windows, as no correction was made for multiple testing at these levels.

2.3.2 Setting 2

This setting features differential usage across a three-exon gene model encoding for two isoforms, with samples drawn from subpopulations 1 and 2 expressing the isoforms at proportions 1:3 and 3:1, respectively. Notably, SigFuge consistently provided high sensitivity, with the exception of when sample size was decreased ($n_1, n_2 = 10, 10$). Furthermore, we observed expected trends across all methods, with sensitivity decreasing with increasing dispersion (ϕ), and increasing with greater sample size (n_1, n_2). In most settings other than low dispersion ($\phi = 0.087$), DEXUS and SIBER showed low sensitivity across all levels of aggregation, with the exception of DEXUS showing high sensitivity with lower sample size. However, care is needed in interpreting this because of the poor specificity shown above for DEXUS in this context.

2.3.3 Setting 3

In this setting, we considered similar differential usage as in Setting 2, using a four-exon gene model. Samples were again drawn from two subpopulations expressing two isoforms at proportions 1:3 and 3:1. Similar results were observed as in Setting 2, with the exception of increased sensitivity by DEXUS and decreased sensitivity by SigFuge in the unbalanced sample size setting ($n_1, n_2 = 75, 25$). In general, sensitivity for both DEXUS and SIBER were higher in Setting 3, likely due to the regions of differential usage comprising a larger proportion of the entire gene.

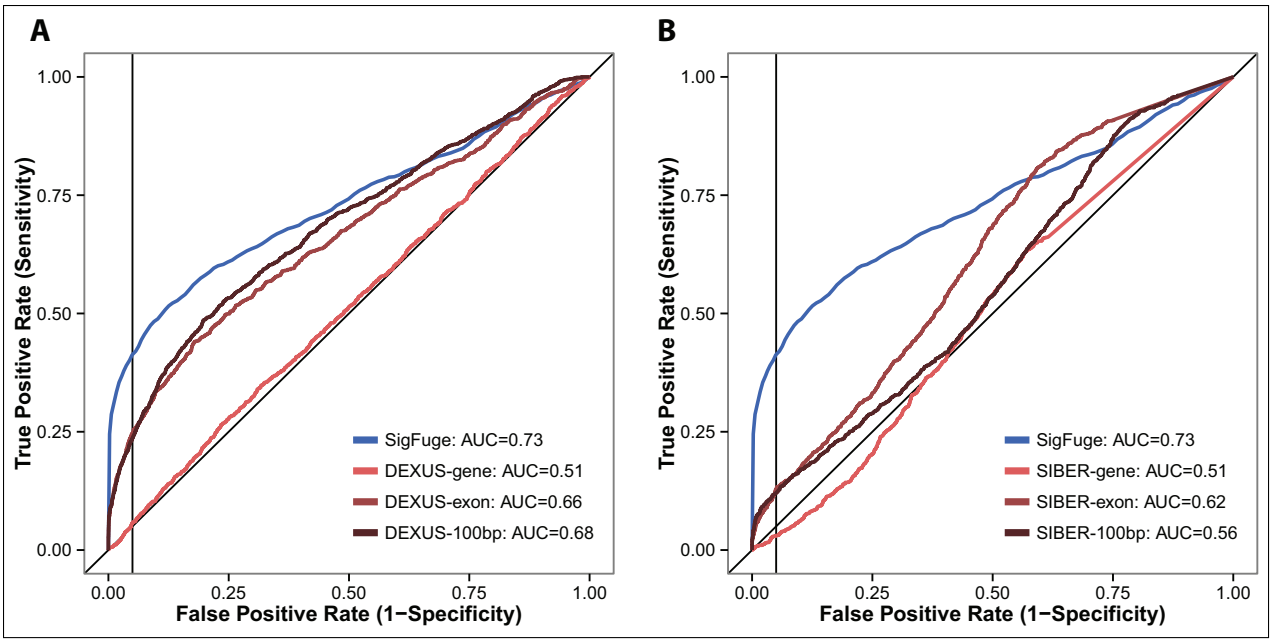

Figure 2.4: ROC curves are shown for SigFuge and 6 competing approaches applied to 10,000 simulated gene loci. The corresponding AUC is reported for each method in the legends, and vertical black lines are used to denote the 95% specificity cutoff. Comparison of ROC curves for SigFuge p-values and (A) DEXUS I/NI indices and (B) SIBER BI indices at the gene, exon and 100bp window levels.

2.3.4 Joint Setting

A joint simulation of 10,000 genes, including 9,000 null and 1,000 non-null genes was also performed to further evaluate the sensitivity and specificity of SigFuge, DEXUS and SIBER. The resulting receiver operating characteristic (ROC) curves for each method are given in Figure 2.4, and corresponding summary statistics are reported in Table 2.3, including area under the ROC curve (AUC), sensitivity, and the F1-measure (the harmonic mean of precision and recall; Powers, 2011). Across all metrics, SigFuge performs the best, with DEXUS consistently outperforming SIBER. Although the exon and 100bp window implementations of DEXUS achieve nearly the same AUC as SigFuge, the DEXUS-based approaches achieve substantially lower sensitivity when specificity is constrained to be above 90 or 95 percent.

In general, SigFuge was found to produce higher sensitivity and lower false positive calls than either DEXUS or SIBER. Additionally, in both Settings 2 and 3, DEXUS and SIBER benefited substantially from aggregating at the exon and 100bp window level. Finally, we note that for all settings, SigFuge required the most computational time, with DEXUS requiring the least. This is a consequence of the simulation based p-value calculation implemented by SigFuge and the

underlying SigClust algorithm. However, as computational times differ with available hardware, it may be more appropriate to interpret these results as a relative, rather than absolute, comparison of computational cost across the evaluated methods. Furthermore, since SigFuge is applied at each locus separately, the method may be easily parallelized on a cluster to reduce the total computing time for larger scale analyses.

2.4 Real Data Analysis

To illustrate the power of our approach in real data, SigFuge was applied to two cancer datasets, consisting of 177 LUSC samples and 279 HNSC samples obtained from the TCGA Research Network. The datasets were processed as described in The Cancer Genome Atlas Research Network (2012). The `samtools depth` function was used to obtain per-base read counts. Union gene models and corresponding composite exon boundaries for 20,500 genes were obtained from the TCGA generic annotation file v2.1 (https://tcga-data.nci.nih.gov/docs/GAF/GAF.hg19.June2011.bundle/outputs/) based on the December 2009 version of the UCSC Gene annotations. Methylation, mutation, and copy number calls for the LUSC dataset at the *CDKN2A* locus were also obtained from the supplementary data for The Cancer Genome Atlas Research Network (2012).

2.4.1 Lung Squamous Cell Carcinoma (LUSC)

Of the 20,500 genes considered, 3,547 genes having less than 10 samples passing the expression threshold were removed from the analysis. Genes of this type were empirically considered to be expressed at such low levels in so few samples that clustering results would be of little interest. The distribution of the remaining 16,953 p-values is shown in Figure 2.5.

Table 2.3: Summary statistics for the joint simulation, including AUC, sensitivity at 90% and 95% specificity (TPR_{90}, TPR_{95}), and the F1-measure at 90% and 95% specificity ($F1_{90}$, $F1_{95}$).

Method	**AUC**	**TPR_{95}**	**TPR_{90}**	**$F1_{95}$**	**$F1_{90}$**
SigFuge-bp	**0.73**	**0.41**	**0.49**	**0.44**	**0.41**
DEXUS-gene	0.51	0.06	0.11	0.08	0.11
DEXUS-exon	0.66	0.25	0.34	0.29	0.30
DEXUS-100bp	0.68	0.24	0.34	0.28	0.30
SIBER-gene	0.51	0.03	0.06	0.04	0.06
SIBER-exon	0.62	0.13	0.17	0.16	0.17
SIBER-100bp	0.56	0.12	0.17	0.15	0.16

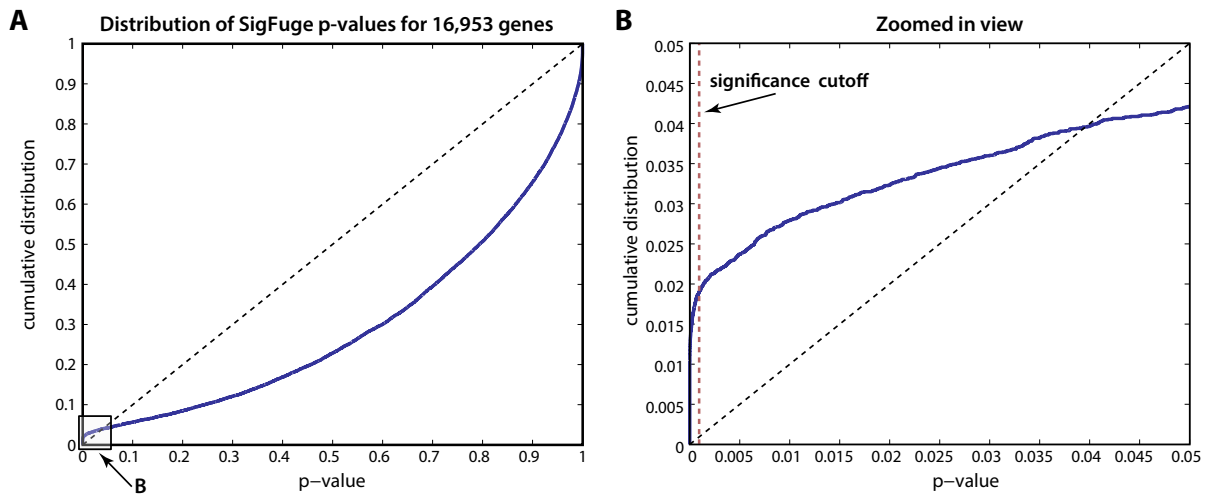

Figure 2.5: Distribution of SigFuge p-values for 16,953 genes with more than 10 highly expressed samples. A dashed red line is used to denote the p-value significance cutoff for controlling FDR at 5%. (A) Empirical cumulative distribution function (CDF) of p-values and (B) zoomed in view of the empirical CDF for the range of p-values < 0.05.

Controlling FDR at 5%, 322 genes were identified as showing significant differential patterns of expression. Manual review of the expression at these genes suggested that SigFuge identified a limited number of recurring patterns. Thus, the set of 322 genes was separated into 6 categories by visually inspecting the corresponding expression plot at each locus (Table 2.4). Genes placed in the same category were determined to exhibit similar patterns of differential isoform usage. While these categories do not necessarily correspond to unique regulatory events, they help summarize the various types of differences detectable by SigFuge.

The first five categories, containing potentially biologically meaningful behavior, include: (1) skipping of a cassette exon, (2) outlier behavior, i.e. differential usage in less than 5 samples, (3) differential use of the 5′- end, (4) differential use of the 3′-end, and (5) alternative start sites.

Table 2.4: Six consistent patterns of differential isoform usage were identified across the set of 322 significant LUSC genes.

Cat.	Name	Count	Representative Genes
1	cassette exon	27	*CDKN2A*, *KLK12*, *FAM64A*
2	outliers	67	*APRT*, *RABAC1*, *TSPO*
3	diff. use of 5′- exons	50	*SPATA21*, *SMN1*, *CKMT1A*
4	diff. use of 3′- exons	15	*RPL22L1*, *CRHR1*, *ECE2*
5	alternative start sites	53	*RPS8*, *RPL7A*, *RPL35A*
6	likely mapping artifacts	110	*S100A7*, *HLA-DRB1*, *RPL27*

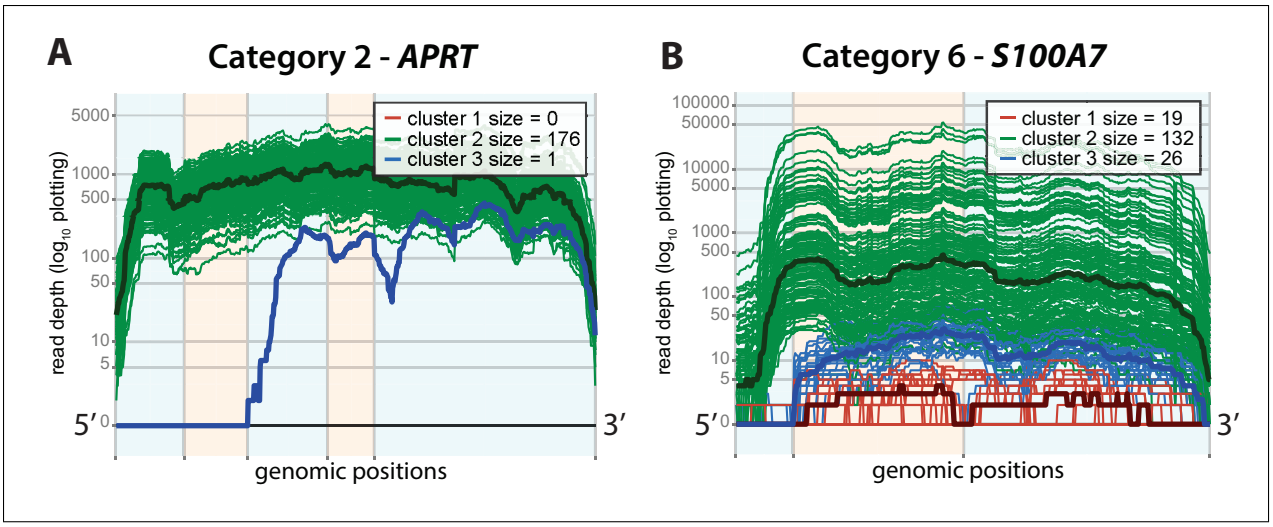

Figure 2.6: Representative genes for two common patterns of differential expression are shown: (A) a single outlier sample (Cat. 2), *APRT* and (B) an unmapped short starting exon (Cat. 6), *S100A7*. Alternating orange and blue are used to denote annotated exon boundaries. Red, blue and green represent clusters of low expression, isoform usage 1 and isoform usage 2. Bold lines denote cluster median expression.

In Figure 2.7, we show the expression plots for three genes with differential usage of a cassette exon (Cat. 1), each described in detail later in this section. In each plot, the region of differential expression along the transcript is highlighted in purple. Additionally, in Figure 2.6A, we show *APRT*, an example of a gene with one clear outlier sample (Cat. 2). We primarily focus on the set of genes in Category 1, as their functional impact is most directly predicted.

Events deemed likely to be artifacts of current RNA-seq technologies and alignment algorithms, such as short unmapped exons (*S100A7*, Figure 2.6B), are included in Category 6. Previous studies have shown that many split-read alignment algorithms have difficulty aligning reads to short exons, especially when overall gene expression is low (Cabanski et al., 2013). Thus, it is highly likely that the identified clusters at *S100A7* simply correspond to samples falling above and below the threshold for properly aligning reads to the first exon.

The set of Category 1 genes include 27 loci identified based on apparent gain or loss of a middle exon. We will now describe in detail three notable genes from this category for which differential isoform usage may play a role in tumor development and growth: *CDKN2A*, *FAM64A*, and *KLK12*.

First, consider *CDKN2A*, a tumor suppressor gene known to code for two proteins, $p16^{INK4a}$ and $p14^{ARF}$. Recently, *CDKN2A* was identified as one of the most highly altered genes in LUSC (The Cancer Genome Atlas Research Network, 2012). In the union gene model shown in Fig-

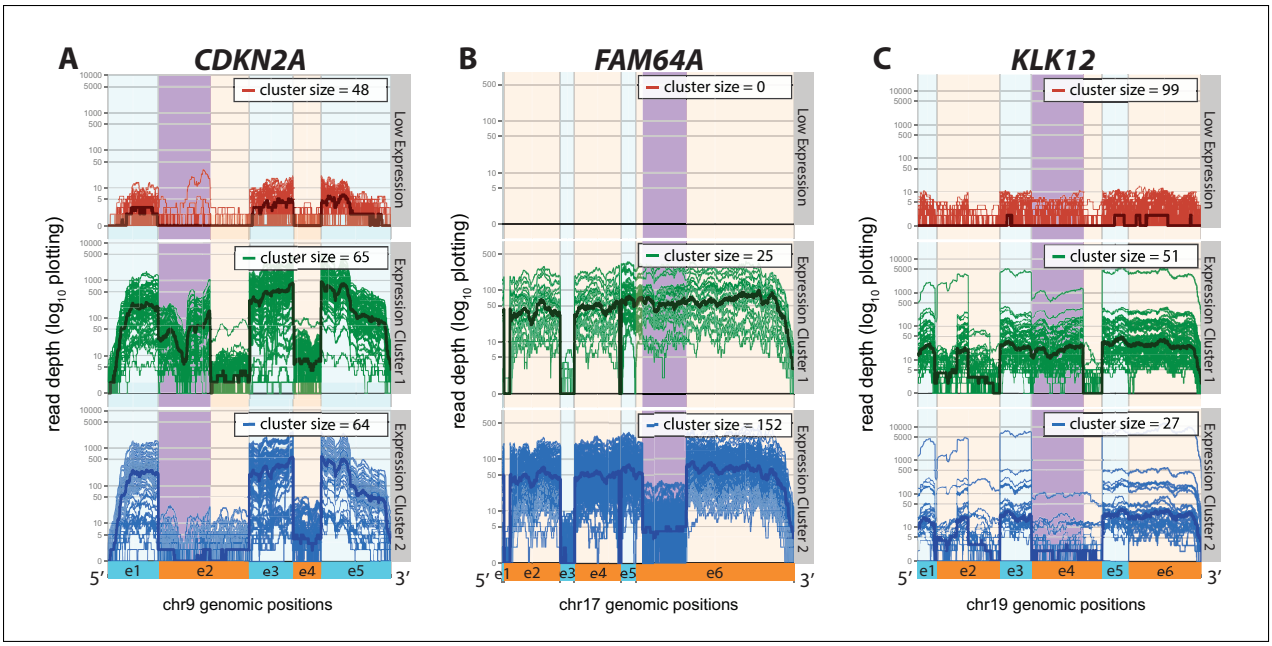

Figure 2.7: Three genes identified as being significant by SigFuge with differential usage of middle exons (Cat. 1). For each gene, the SigFuge clusters are shown using separate colors and panels. Each red, green and blue curve represents an individual sample and darker bold curves are used to denote the cluster medians. Regions of differential usage, identified by visual inspection, are highlighted in the figure for each gene. Alternating orange and blue are used to denote annotated exon boundaries.

ure 2.7A, expression of exons 1−3−5 encodes for $p14^{ARF}$, and expression of exons 2−3−5 encodes for $p16^{INK4a}$. Thus, the SigFuge classes correspond to expression of neither protein due to low expression (red class), expression of $p16^{INK4a}$ and $p14^{ARF}$ (green class) and expression of $p14^{ARF}$ only (blue class).

Table 2.5 compares the SigFuge clusters against the three major modes of *CDKN2A* inactivation identified by the TCGA integrative analysis: homozygous deletion, epigenetic silencing by methylation, and inactivation by point mutation. As can be seen by the clear diagonal structure of the table, the SigFuge clusters approximately capture the three classes of alterations. Notably, 64% of samples in the blue class were methylated, including all but one methylated sample in the cohort. Furthermore, 96% of samples identified as low expression (red class) were homozygous deleted, comprising 82% of homozygous deleted samples in all clusters, confirming the validity of our proposed filtering scheme. Pearson's chi-square (χ^2) tests were applied to each row of Table 2.5 and the entire table. The highly significant p-values further confirm the strong association between our SigFuge clusters and the previously identified alterations.

SigFuge also identified *FAM64A*, a gene that has been implicated in the regulation of cell proliferation, suggesting a possible role in cancer (Archangelo et al., 2008). *FAM64A* has been shown to be highly expressed in leukemia, lymphoma and other tumor cell lines (Archangelo et al., 2006). The plot of *FAM64A* expression shows clustering based on an unannotated splice junction, resulting in lower expression for a large proportion of the final exon (Figure 2.7B). Although the event has been previously reported as a retained intron (Coulombe-Huntington et al., 2009), the implication of the isoform difference has yet to be described. This supports our use of per-base expression, as analysis based on aggregation along exon or whole gene boundaries would have likely missed this event.

The final Category 1 gene which we focus on, *KLK12*, is part of a family of 15 kallikrein-related peptidases (*KLK* genes) encoding secreted serine proteases. *KLK* splice variants are receiving increased attention as potential biomarkers in cancer, and have been studied in epithelial ovarian, prostate, and lung cancers (Dong et al., 2003, 2005; Planque et al., 2010). The *KLK12* locus is known to produce multiple isoforms, largely differing by the use of a cassette exon, exon 4 (Figure 2.7C exon 4). A recent study has shown expression of the exon 4 skipping isoform to be clinically relevant in breast cancer (Talieri et al., 2012). The identified *KLK12* expression clusters capture evidence of similar differential usage in our cohort of 177 samples. These results support the potential of *KLK12* and other *KLK* splice variants as markers in LUSC.

To confirm our identified clusters were not an artifact of sequencing, we performed PCR at the *KLK12* gene locus on representative samples from each of the green and blue classes in Figure 2.7C. The results of PCR were visualized by agarose gel electrophoresis (Figure 2.8). The coverage plot shown in Figure 2.8A suggests clear differential expression at exon 4 between the two representative

Table 2.5: SigFuge label and genomic alteration agreement at *CDKN2A*

	SigFuge Label				
	Red	**Green**	**Blue**	**Total**	χ^2 p-value
Homozygous Deleted	46	0	8	54	3×10^{-15}
Mutated	0	29	3	32	4×10^{-11}
Methylated	1	0	41	42	≈ 0
None	1	36	12	49	3×10^{-9}
Total	48	65	64	177	≈ 0

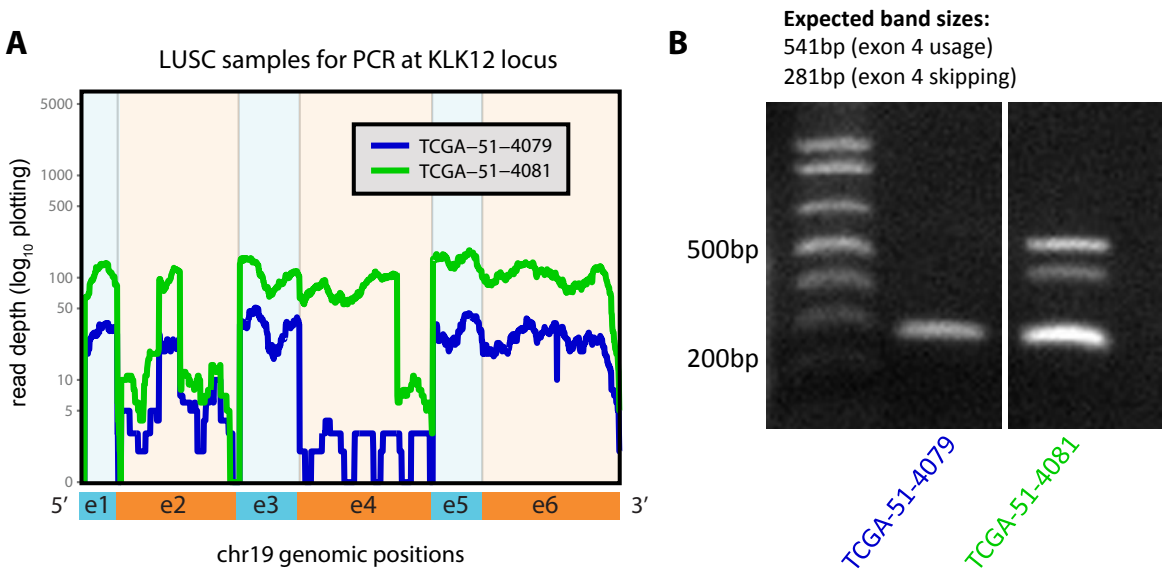

Figure 2.8: (A) The expression curves are shown for two samples selected for validation. Sample TCGA-51-4079 shows a drop in expression at exon 4. (B) The inclusion or exclusion of exon 4 within *KLK12* gene transcripts was assessed by PCR and results were visualized by agarose gel electrophoresis.

samples. Primers for PCR were chosen such that isoforms which include exon 4 produce 541bp fragments, and isoforms which splice out exon 4 produce 261bp fragments. Thus, the absence of a band at the 541bp mark for the sample from the blue cluster (TCGA-51-4079) in Figure 2.8B validates the hypothesis that samples of this cluster do not express *KLK12* isoforms which include exon 4. However, note that both samples show expression of the exon 4 skipping isoforms, as observed by the bands at 281bp.

2.4.2 Head and Neck Squamous Cell Carcinoma (HNSC)

As an attempt to confirm the results identified in LUSC, SigFuge was also applied to an independent set of 279 HNSC samples, a biologically similar tumor type. Controlling FDR at 5%, 335 genes were identified as exhibiting significant differential usage. Notably, similar clusters of differential isoform usage were identified at the *CDKN2A*, *KLK12* and *FAM64A* loci. The clustered expression plots for the 279 HNSC samples at these genes are shown in Figure 2.9. Of the three, *KLK12* and *FAM64A* were included in the set of 335 significant HNSC genes with p-values $1.99e-15$, and $6.76e-6$. While not included in the top 335 genes, *CDKN2A* was also found to exhibit strong evidence of differential isoform usage (p-value 0.0021, 381st most significant). Furthermore, of the

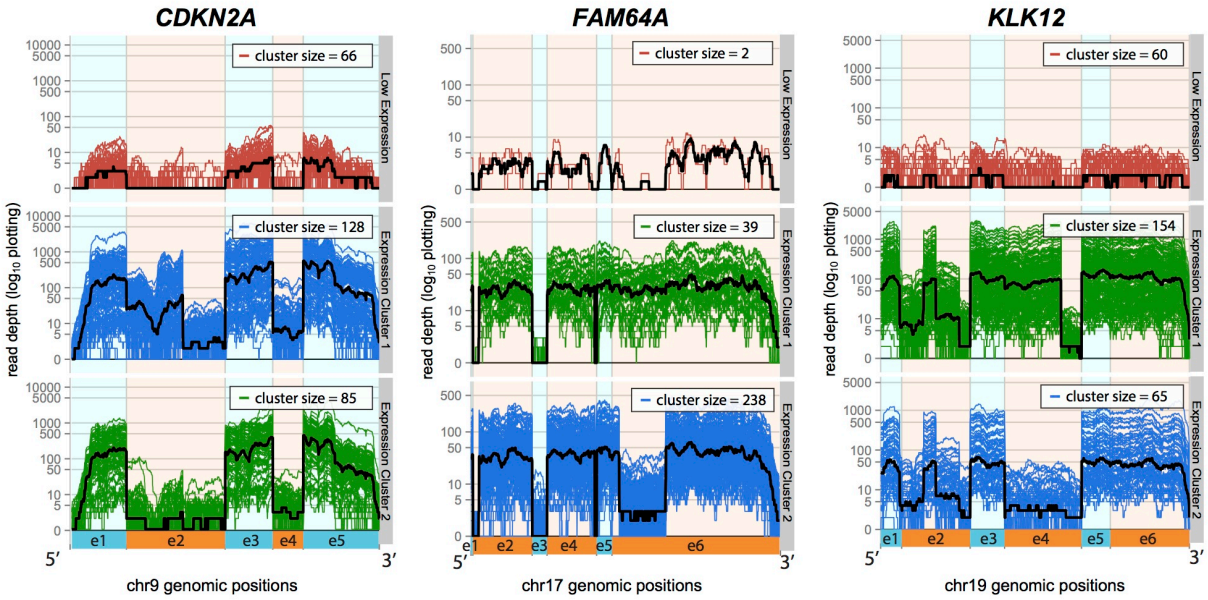

Figure 2.9: SigFuge clusters in a cohort of HNSC samples for three genes identified as highly significant in the LUSC samples. All three genes were found to be highly significant HNSC, with clustering patterns strongly resembling those identified in LUSC (Figure 2.7). Alternating orange and blue are used to denote annotated exon boundaries.

27 Category 1 genes identified in LUSC, 21 (78%) were also identified as significant in HNSC, suggesting the reproducibility of most interesting events across different datasets.

2.5 Discussion

The introduction of RNA-seq has fundamentally transformed genomic research in cancer by making it possible to study transcriptomes at the resolution of base positions. Concurrently, the importance of studying isoform regulatory behavior beyond whole gene events has become increasingly clear. SigFuge is presented as a novel method capable of unsupervised discovery of differential isoform events in RNA-seq. Our approach to studying gene expression as per-base expression curves along transcriptome coordinates makes it possible to identify differential events without strictly constraining our analysis to proposed exon or transcript boundaries.

Through simulation study, we have shown that SigFuge is often capable of detecting true differential isoform usage with higher sensitivity than DEXUS and SIBER across various experimental conditions. This may be attributed to the unique multivariate approach taken by SigFuge, in which all base positions and exons are considered simultaneously to detect and assess clustering. In

contrast, DEXUS and SIBER cluster marginally at individual genes or exons, thus rendering the approaches less sensitive to isoform differences which occur non-uniformly, but in concert, across the entire gene. In addition to having high sensitivity, we have shown that in the absence of sub-population differences, SigFuge does not make more than the expected number of false positive calls.

Applying SigFuge to a cohort of LUSC samples, we identified *CDKN2A*, a tumor suppressor gene known to be highly altered in LUSC, and *KLK12*, a gene recently shown to have differential isoform usage in breast cancer. To our knowledge, SigFuge is the only unsupervised approach for identifying loci with significant differential isoform usage. All other genome-wide methods for identifying genes with differential isoform usage require *a priori* knowledge of the differential class labels, and therefore could not be used to identify these events. The biological relevance of our *CDKN2A* clusters was validated by observed high concordance with homozygous deletion, methylation and point mutation events at the locus. Further, the predicted isoforms of *KLK12* were confirmed by PCR as a validation of the method. Additionally, many of the clusters identified in LUSC were found to reproduce in an independent analysis of 279 HNSC samples, suggesting that our discoveries relate to biologically meaningful events.

The importance of alternative splicing in the development of diseases, including cancer, is well recognized (Faustino and Cooper, 2003; Venables, 2004). SigFuge shows promise as a tool for identifying biologically relevant cases of aberrant isoform usage. Given clinical outcomes, testing clusters of differential isoform usage for significant associations with survival could potentially reveal novel therapeutic targets.

A major benefit of SigFuge is the calculation of a p-value to quantify significance of clustering. Using the p-value, it becomes possible to screen a large set of genes to identify a small subset of potentially biologically interesting loci. However, our post-hoc analysis makes it clear that identifying truly interesting events is not simply a statistical endeavor, i.e. finding significant SigFuge p-values. That is, some loci identified by SigFuge as statistically significant, may on manual review appear to be artifacts introduced by sequencing and mapping challenges beyond our control. Therefore, manual review of statistically significant results is strongly recommended. To this end, our approach to visualizing expression profiles makes it possible to quickly gain intuition at each locus to determine the nature of the underlying event.

Recent genomics studies using RNA-seq are beginning to shed light on the sheer prevalence and importance of post-transcriptional events across the human genome. In this chapter, we have proposed the first approach for the unsupervised discovery of differential isoform usage in RNA-seq data. By taking the novel approach of clustering by per-base expression, we believe SigFuge is a step in the right direction for realizing the full potential of RNA-seq for understanding the genomic complexity of diseases.

CHAPTER 3

Statistical Significance for Hierarchical Clustering (SHC)

3.1 Introduction

Clustering describes the unsupervised learning task of partitioning observations into homogenous subsets to uncover subpopulation structure in a dataset. As an unsupervised learning task, cluster analysis makes no use of label or outcome data. A large number of methods have been proposed for clustering, several of which were described in detail in Section 1.2, including both hierarchical and non-nested approaches. Since the work of Eisen et al. (1998), hierarchical clustering algorithms have enjoyed substantial popularity for the exploratory analysis of gene expression data. In several landmark papers that followed, these methods were successfully used to identify clinically relevant expression subtypes in lymphoma, breast, and other types of cancer (Perou et al., 2000; Bhattacharjee et al., 2001).

While non-nested clustering algorithms typically require pre-specifying the number of clusters of interest, K, hierarchical algorithms do not. Instead, hierarchical approaches produce a single nested hierarchy of clusters from which a partition can be obtained for any possible choice of K. As a result, hierarchical clustering provides an intuitive way to study relationships among clusters not possible using non-nested approaches. The popularity of hierarchical clustering in practice may also be largely attributed to dendrograms (Figure 1.4B), a highly informative visualization of the clustering as a binary tree.

As described in Section 1.2, while dendrograms provide an intuitive representation for studying the results of hierarchical clustering, the researcher is still ultimately left to decide which partitions along the tree to interpret as biologically important subpopulation differences. Often, in genomic studies, the determination and assessment of subpopulations are left to heuristic or *ad hoc* methods (Verhaak et al., 2010; Wilkerson et al., 2010; Bastien et al., 2012). To provide a statistically sound alternative to these methods, in this chapter we introduce statistical Significance of Hier-

archical Clustering (SHC), a Monte Carlo based approach for assessing the statistical significance of clustering along a hierarchical partition. The approach makes use of the ordered and nested structure in the output of hierarchical clustering to reduce the problem to a sequence of hypothesis tests descending the tree. Each test is formulated such that the procedure may be applied even in the high-dimension low-sample size (HDLSS) setting, where the number of variables is much greater than the number of observations. This is of particular importance, as the number of measured variables in genomic studies continues to grow with advances in high-throughput sequencing technologies, such as RNA-seq (Marioni et al., 2008; Wang et al., 2009). A stopping rule along the sequence of tests is also provided to control the family-wise error rate (FWER) of the entire procedure.

Several approaches have been proposed to address the question of statistical significance in the non-nested setting. The SigClust hypothesis test mentioned in Chapters 1 and 2 was introduced by Liu et al. (2008) for assessing the significance of clustering in HDLSS settings using a Monte Carlo procedure. While well-suited for detecting the presence of more than a single cluster in a dataset, the approach was not developed with the intention of testing in hierarchical or multi-cluster settings. This approach is described in greater detail in Section 3.2. More recently, Maitra et al. (2012) proposed a bootstrap based approach capable of testing for any number of clusters in a dataset. However, in addition to not directly addressing the hierarchical problem, their approach has not been evaluated in the important HDLSS setting. As such, neither approach provides a solution for handling the structure and multiplicity of nested tests unique to hierarchical clustering.

For assessing statistical significance in the hierarchical setting, Suzuki and Shimodaira (2006) developed the `R` package `pvclust`. The hypothesis tests used in `pvclust` are based on bootstrapping procedures originally proposed for significance testing in the context of phylogenetic tree estimation (Efron et al., 1996; Shimodaira, 2004). Since the procedure is based on a nonparamateric bootstrapping of the covariates, while `pvclust` can be used in the HDLSS setting, it cannot be implemented when the dataset is of low-dimension. In contrast, SHC may be used in either setting. The overall approach of `pvclust` differs fundamentally from that of SHC and is discussed briefly in Section 3.6. To our knowledge, no other approaches have been proposed for assessing the statistical significance of hierarchical clustering.

The remainder of this chapter is organized as follows. In Section 3.2 we first review hierarchical clustering and describe the SigClust hypothesis test of Liu et al. (2008). Then, in Section 3.3, we introduce our proposed SHC approach. In Section 3.4, we present theoretical justifications for our method under the HDLSS asymptotic setting. In Section 3.5, we describe a simple improvement to the null estimation procedures of SigClust and SHC. We then evaluate the performance of our SHC method as well as our improved null estimation procedure under various simulation settings in Section 3.6. In Section 3.7, we apply our method to two cancer gene expression datasets. Finally, we conclude with a discussion in Section 3.8. All technical proofs are included in Section 3.9, and complete simulation results are presented in Section 3.10. The SHC procedure is implemented in `R`, and is available at `http://github.com/pkimes/`.

3.2 Clustering and Significance

We begin this section by first providing a brief review of hierarchical clustering. We then describe the K-means based SigClust approach of Liu et al. (2008) for assessing significance of clustering in HDLSS data.

3.2.1 Hierarchical Clustering

Given a collection of N unlabeled observations, $\mathbb{X} = \{\boldsymbol{x}_i, \ldots, \boldsymbol{x}_N\}$, algorithms for hierarchical clustering estimate all $K = 1, \ldots, N$ partitions of the data through a sequential optimization procedure. The sequence of steps can be implemented as either an agglomerative (bottom-up) or divisive (top-down) approach to produce the nested hierarchy of clusters. In this chapter we focus on agglomerative approaches which are more often used in practice. Common choices of dissimilarity and linkage functions used for agglomerative hierarchical clustering are provided in Section 1.2. The sequence of clustering solutions obtained by hierarchical clustering is naturally visualized as a binary tree, commonly referred to as a dendrogram. An example of hierarchical clustering is shown in Figure 1.3. For a more complete review of hierarchical clustering, we refer the reader to Subsection 1.2.2.

3.2.2 Statistical Significance

We next describe the SigClust hypothesis test of Liu et al. (2008) for assessing significance of clustering. To make inference in the HDLSS setting tractable, SigClust makes the simplifying assumption that a cluster may be characterized as a subset of the data which follows a single Gaussian distribution. While no universal definition for a "cluster" exists, the Gaussian definition is often used as a reasonable approximation (McLachlan and Peel, 2000; Fraley and Raftery, 2002). While potentially restrictive, the Gaussian definition and SigClust approach have provided sensible results in real high-dimensional datasets (Verhaak et al., 2010; Bastien et al., 2012). Therefore, to determine whether a dataset is comprised of more than a single cluster, the approach tests the following hypotheses:

H_0 : the data follow a single Gaussian distribution

H_1 : the data follow a non-Gaussian distribution.

The corresponding p-value is calculated using the 2-means cluster index (CI), a statistic sensitive to the null and alternative hypotheses. Letting C_k denote the set of indices of observations in cluster k and using $\bar{\boldsymbol{x}}_k$ to denote the corresponding cluster mean, the 2-means CI is defined as

$$\text{CI} = \frac{\sum_{k=1}^{2} \sum_{i \in C_k} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k\|_2^2}{\sum_{i=1}^{N} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}\|_2^2} = \frac{SS_1 + SS_2}{TSS}, \tag{3.1}$$

where TSS and SS_k are the total and within-cluster sum of squares. Smaller values of the 2-means CI correspond to tighter clusters, and provide stronger evidence of clustering of the data. The statistical significance of a given pair of clusters is calculated by comparing the observed 2-means CI against the distribution of 2-means CIs under the null hypothesis of a single Gaussian distribution. Since a closed form of the distribution of CIs under the null is unavailable, it is empirically approximated by the CIs computed for hundreds, or thousands, of datasets simulated from a null Gaussian distribution estimated using the original dataset. An empirical p-value is calculated by the proportion of simulated null CIs less than the observed CI. Approximations to the optimal 2-means CI for both the observed and simulated datasets can be obtained using the K-means algorithm for two clusters.

In the presence of strong clustering, the empirical p-value may simply return 0 if all simulated CIs fall above the observed value. This can be particularly uninformative when trying to compare the significance of multiple clustering events. To handle this problem, Liu et al. (2008) proposed computing a *Gaussian fit p-value* in addition to the empirical p-value. Based on the observation that the distribution of CIs appears roughly Gaussian, the Gaussian fit p-value is calculated as the lower tail probability of the best-fit Gaussian distribution to the simulated null CIs.

An important issue not discussed above is the estimation of the covariance matrix of the null distribution, a non-trivial task in the HDLSS setting. A key part of the SigClust approach is the simplification of this problem, by making use of the invariance of the 2-means CI to translations and rotations of the data in the Euclidean space. It therefore suffices to simulate data from an estimate of any rotation and shift of the null distribution. Conveniently, by centering the distribution at the origin, and rotating along the eigendirections of the covariance matrix, the task can be reduced to estimating only the eigenvalues of the null covariance matrix. As a result, the number of parameters to estimate is reduced from $p(p+1)/2$ to p. However, in the HDLSS setting, even the estimation of p parameters is challenging, as $N \ll p$. To solve this problem, the additional assumption is made that the null covariance matrix follows a factor analysis model. That is, under the null hypothesis, the observations are assumed to have been drawn from a single Gaussian distribution, $N(\boldsymbol{\mu}, \Sigma)$, with Σ having eigendecomposition $\Sigma = U\Lambda U^T$ such that

$$\Lambda = \Lambda_0 + \sigma_b^2 \mathbf{I}_p,$$

where Λ_0 is a low rank ($< N$) diagonal matrix of true signal, σ_b^2 is a relatively small amount of background noise, and $\mathbf{I}_p$ is the p-dimensional identity matrix. Letting w denote the number of non-zero entries of Λ_0, under the factor analysis model, only $w+1$ parameters must be estimated to implement SigClust. Several approaches have been proposed for estimating σ_b^2 and the w non-zero entries of Λ_0, including the hard-threshold, soft-threshold, and sample-based approaches (Liu et al., 2008; Huang et al., 2014). Briefly, given the eigenvalues of the sample covariance matrix, $\hat{\lambda}_j$, and an estimate of the background noise, $\hat{\sigma}_b^2$, the hard and soft approaches estimate the diagonal entries of Λ to be $\max\{\hat{\lambda}_j, \hat{\sigma}_b^2\}$ and $\max\{\hat{\lambda}_j - \tau, \hat{\sigma}_b^2\}$, respectively, with tuning parameter $\tau \geq 0$. The sample-based approach simply estimates the diagonal entries of Λ by the $\hat{\lambda}_j$. In the simulation studies presented in Section 3.6, we implement SigClust using the soft-thresholding approach, as

suggested in Huang et al. (2014). In the original SigClust paper, Liu et al. (2008) proposed to estimate the background noise by:

$$\hat{\sigma}_{Raw} = \frac{MAD_{p \cdot N \text{ data}}}{MAD_{N(0,1)}}, \tag{3.2}$$

where $MAD_{p \cdot N \text{ data}}$ is used to denote the median absolute deviation about the median (MAD) computed from the $p \cdot N$ total entries of the original data matrix, and similarly, $MAD_{N(0,1)}$ is used to denote the MAD of a standard Gaussian distribution. In Section 3.5, we propose an alternative background noise estimator, $\hat{\sigma}_{PC}$, using scaled principal component (PC) scores.

3.3 Methodology

To assess significance of clustering in a hierarchical partition, we propose a sequential testing procedure in which Monte Carlo based hypothesis tests are preformed at select nodes along the corresponding dendrogram. In this section, we introduce our SHC algorithm in two parts. First, using a toy example, we describe the hypothesis test performed at individual nodes. Then, we describe our sequential testing procedure for controlling the FWER of the algorithm along the entire dendrogram.

3.3.1 SHC Hypothesis Test

Throughout, we use $j \in \{1, \ldots, N-1\}$ to denote the node index, such that $j = 1$ and $j = (N-1)$ correspond to the top-most (root) and bottom-most merges along the dendrogram, respectively. In Figure 3.1, we illustrate one step of our sequential algorithm using a toy dataset of $N = 150$ observations drawn from $\mathbb{R}^2$ (Figure 3.1A). Agglomerative hierarchical clustering was applied using Ward's linkage to obtain the dendrogram in Figure 3.1B. Consider the second node from the top, i.e. $j = 2$. The corresponding observations and subtree are highlighted in panels A and B of Figure 3.1. Here, we are interested in whether the sets of 43 and 53 observations joined at node 2, denoted by dots and $\times$'s, more naturally define one or two distinct clusters. Assuming that a cluster may be well approximated by a single Gaussian distribution, we propose to test the following hypotheses at node 2:

H_0 : The 96 observations follow a single Gaussian distribution

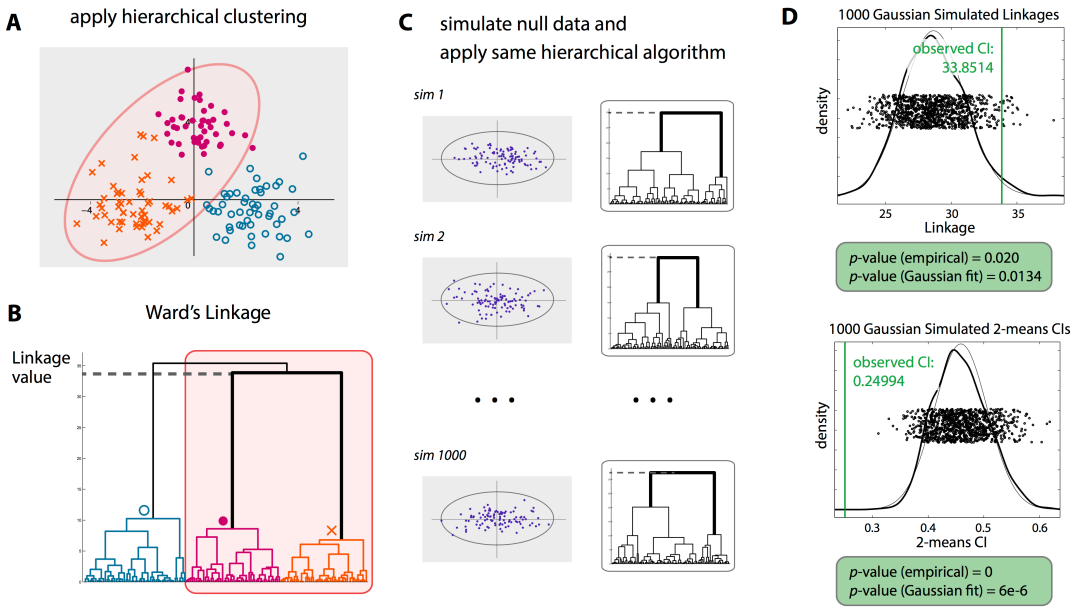

Figure 3.1: The SHC testing procedure illustrated using a toy example. Testing is applied to the 96 observations joined at the second node from the root. (A) Scatterplot of the observations in $\mathbb{R}^2$. (B) The corresponding dendrogram. (C) Hierarchical clustering applied to 1000 datasets simulated from a null Gaussian estimated from the 96 observations. (D) Distributions of null cluster indices used to calculate the empirical SHC p-values.

H_1 : The 96 observations do not follow a single Gaussian distribution.

The p-value at the node, denoted by p_j, is calculated by comparing the strength of clustering in the observed data against that for data clustered using the same hierarchical algorithm under the null hypothesis. We consider two cluster indices, linkage value and the 2-means CI, as natural measures for the strength of clustering in the hierarchical setting. To approximate the null distribution of cluster indices, 1000 datasets of 96 observations are first simulated from a null Gaussian distribution estimated using only the 96 observations included in the highlighted subtree. Then, each simulated dataset is clustered using the same hierarchical algorithm as was applied to the original dataset (Figure 3.1C). As with the observed data, the cluster indices are computed for each simulated dataset using the two cluster solution obtained from the hierarchical algorithm. Finally, p-values are obtained from the proportion of null cluster indices indicating stronger clustering than the observed indices (Figure 3.1D). For the linkage value and 2-means CI, this corresponds to larger and smaller values, respectively. As in SigClust, we also compute a Gaussian approximate p-value

in addition to the empirical p-value. In this example, the resulting empirical p-values, 0.020 and 0, using linkage and the 2-means CI, both suggest significant clustering at the node.

In estimating the null Gaussian distribution, we first note that many popular linkage functions, including Ward's, single, complete and average, are defined with respect to the pairwise dissimilarities of observations belonging to two clusters. As such, the use of these linkage functions with any dissimilarity satisfying translation and rotation invariance, such as Euclidean or squared Euclidean distance, naturally leads to the invariance of the entire hierarchical procedure. Thus, for several choices of linkage and dissimilarity, the SHC p-value can be equivalently calculated using data simulated from a simplified distribution centered at the origin, with diagonal covariance structure. To handle the HDLSS setting, as in SigClust, we further assume that the covariance matrix of the null Gaussian distribution follows a factor analysis model, such that the problem may be addressed using the hard-threshold, soft-threshold and sample approaches previously proposed in Liu et al. (2008) and Huang et al. (2014).

Throughout this chapter we derive theoretical and simulation results using squared Euclidean dissimilarity with Ward's linkage, an example of a translation and rotation invariant choice of dissimilarity and linkage function. However, our approach may be implemented using a larger class of linkages and appropriately chosen dissimilarity functions. We focus on Ward's linkage clustering as the approach may be interpreted as characterizing clusters as single Gaussian distributions, as in the hypotheses we propose to test. Additionally, we have observed that Ward's linkage clustering often provides strong clustering results in practice.

Note that at each node, the procedure requires fitting a null Gaussian distribution using only the observations contained in the corresponding subtree. We therefore set a minimum subtree size, $N_{\min}$, for testing at any node. For the simulations described in Section 3.6, we use $N_{\min} = 10$.

In this subsection, we have described only a single test of the entire SHC procedure. For a dataset of N observations, at most $(N-1)$ tests may be performed at the nodes along the dendrogram. While the total possible number of tests is typically much smaller due to the minimum subtree criterion, care is still needed to account for the issue of multiple testing. In the following subsection, we describe a sequential approach for controlling the FWER to address this issue.

3.3.2 Multiple Testing Correction

To control the FWER of the SHC procedure, one could simply test at all nodes simultaneously, and apply an equal Bonferroni correction to each test. However, this approach ignores the clear hierarchical nature of the tests. Furthermore, the resulting dendrogram may have significant calls at distant and isolated nodes, making the final output difficult to interpret. Instead, we propose to control the FWER using a sequential approach which provides greater power at the more central nodes near the root of the dendrogram, and also leads to more easily interpretable results.

To correct for multiple testing, we employ the FWER controlling procedure of Meinshausen (2008) original proposed in the context of variable selection. For the SHC approach, the FWER along the entire dendrogram is defined to be the probability of at least once, falsely rejecting the null at a subtree of the dendrogram corresponding to a single Gaussian cluster. To control the FWER at level $\alpha \in (0, 1)$, we perform the hypothesis test described above at each node j, with the modified significance cutoff:

$$\alpha_j^* = \alpha \cdot \frac{N_j - 1}{N - 1},$$

where N_j is used to denote the number of observations clustered at node j. Starting from the root node, i.e. $j = 1$, we descend the dendrogram rejecting at nodes for which the following two conditions are satisfied: (C1) $p_j < \alpha_j^*$, and (C2) the parent node was also rejected, where the parent of a node is simply the one directly above it. For the root node, condition (C2) is ignored. As the procedure moves down the dendrogram, condition (C1) and the modified cutoff, α_j^*, apply an increasingly stringent correction to each test, proportional to the size of the corresponding subtree. Intuitively, if the subtree at a node contains multiple clusters, the same is true of any node directly above it. Condition (C2) formalized this intuition by forcing the set of significant nodes to be well connected from the root. Furthermore, recall that the hypotheses tested at each node assess whether or not the two subtrees were generated from a single Gaussian distribution. While appropriate when testing at nodes which correspond to one or more Gaussian distributions, the interpretation of the test becomes more difficult when applied to only a portion of a single Gaussian distribution, e.g. only half of a Gaussian cluster. This can occur when testing is performed at a node which falls below a truly null node. In this case, while the two subtrees of the node correspond to non-Gaussian

distributions, they do not correspond to interesting clustering behavior. Thus, testing at such nodes may result in truly positive, but uninteresting, significant calls. By restricting the set of significant nodes to be well connected from the root, in addition to controlling the FWER, our procedure also limits the impact of such undesirable tests.

3.4 Theoretical Development

In this section, we study the theoretical behavior of our SHC procedure with linkage value as the measure of cluster strength applied to Ward's linkage hierarchical clustering. We derive theoretical results for the approach under both the null and alternative hypotheses. In the null setting, the data are sampled from a single Gaussian distribution. Under this setting, we show that the empirical SHC p-value at the root node follows the $U(0,1)$ distribution. In the alternative setting, we consider the case when the data follow a mixture of two spherical Gaussian distributions. Since SHC is a procedure for assessing statistical significance given a hierarchical partition, the approach depends heavily on the algorithm used for clustering. We therefore first provide conditions for which Ward's linkage clustering asymptotically separates samples from the two components at the root node. Given these conditions are satisfied, we then show that the corresponding empirical SHC p-value at the root node tends to 0 asymptotically as both the sample size and dimension grow to infinity. All proofs are included in Section 3.9.

We first consider the null case where the data, $\mathbb{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$, are sampled from a single Gaussian distribution, $N(\mathbf{0}, \boldsymbol{\Sigma})$. The following proposition describes the behavior of the empirical p-value at the root node under this setting.

Proposition 3.1. *Suppose $\mathbb{X}$ were drawn from a single Gaussian distribution, $N(\mathbf{0}, \boldsymbol{\Sigma})$, with known covariance matrix $\boldsymbol{\Sigma}$. Then, the SHC empirical p-value at the root node follows the $\mathbf{U}(0,1)$ distribution.*

The proof of Proposition 3.1 is omitted, as it follows directly from an application of the probability integral transform. We also note that the result of Proposition 3.1 similarly holds for any subtree along a dendrogram corresponding to a single Gaussian distribution. Combining this with Theorem 1 of Meinshausen (2008), we have that the modified p-value cutoff procedure of Section 3.3.2 controls the FWER at the desired level α.

We next consider the alternative setting. Suppose the data, $\mathbb{X}$, were drawn from a mixture of two Gaussian subpopulations in $\mathbb{R}^p$, denoted by $N(\boldsymbol{\mu}_1, \sigma_1^2 \mathbf{I}_p)$ and $N(\boldsymbol{\mu}_2, \sigma_2^2 \mathbf{I}_p)$. Let $\mathbb{X}^{(1)} = \{\boldsymbol{X}_1^{(1)}, \ldots, \boldsymbol{X}_n^{(1)}\}$ and $\mathbb{X}^{(2)} = \{\boldsymbol{X}_1^{(2)}, \ldots, \boldsymbol{X}_m^{(2)}\}$ denote the $N = n+m$ observations of $\mathbb{X}$ drawn from the two mixture components. In the following results, we consider the HDLSS asymptotic setting where $p \to \infty$ and $n = p^\alpha + o(p)$, $m = p^\beta + o(p)$ for $\alpha, \beta \in (0, 1)$ (Hall et al., 2005). As in Borysov et al. (2014), we assume that the mean of the difference $(\boldsymbol{X}_i^{(1)} - \boldsymbol{X}_j^{(2)})$ is not dominated by a few large coordinates in the sense that for some $\epsilon > 0$,

$$\sum_{k=1}^{p} (\mu_{1,k} - \mu_{2,k})^4 = o\left(p^{2-\epsilon}\right), \quad p \to \infty. \tag{3.3}$$

Given this assumption, the following theorem provides necessary conditions for Ward's linkage clustering to correctly separate observations of the two mixture components.

Theorem 3.1. *Suppose (3.3) is satisfied and the dendrogram is constructed using the Ward's linkage function. Let n, m be the number of observations sampled from the two Gaussian mixture components, $N(\boldsymbol{\mu}_1, \sigma_1^2 \boldsymbol{I}_p)$ and $N(\boldsymbol{\mu}_2, \sigma_2^2 \boldsymbol{I}_p)$, with $\sigma_1 \leq \sigma_2$. Additionally, suppose $n = p^\alpha + o(p)$, $m = p^\beta + o(p)$ for $\alpha, \beta \in (0, 1)$, and let μ^2 denote $p^{-1}\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2^2$. Then, if $\limsup \frac{n^{-1}(\sigma_2^2 - \sigma_1^2)}{\mu^2} < 1$, $\boldsymbol{X}^{(1)}$ and $\boldsymbol{X}^{(2)}$ are separated at the root node with probability converging to 1 as $p \to \infty$.*

Theorem 3.1 builds on the asymptotic results for hierarchical clustering described in Borysov et al. (2014). The result provides a theoretical analysis of Ward's linkage clustering, independent of our SHC approach. In the following result, using Theorem 3.1, we show that under further assumptions, the SHC empirical p-value is asymptotically powerful at the root node of the dendrogram. That is, the p-value converges to 0 as p, n, m grow to infinity.

Theorem 3.2. *Suppose the assumptions for Theorem 3.1 are satisfied. Furthermore, suppose σ_1^2 and σ_2^2 are known. Then, using linkage as the measure of cluster strength, the empirical SHC p-value at the root node along the dendrogram equals 0 with probability converging to 1 as $p \to \infty$.*

By Theorem 3.2, the SHC procedure is asymptotically well powered to identify significant clustering structure in the presence of multiple Gaussian components. While in this section we only considered the theoretical behavior of SHC using linkage value as the measure of cluster strength, empirical results presented in Section 3.6 provide justification for alternatively using the 2-means CI.

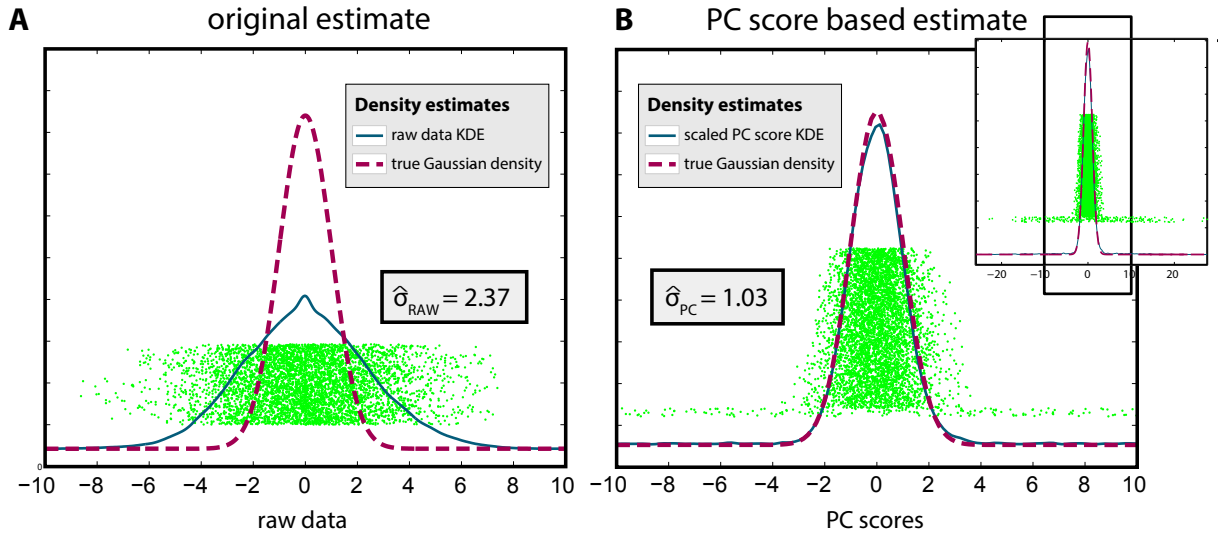

Figure 3.2: (A) Estimation using vectorized $(p \cdot N)$ entries. (B) Estimation by vectorized $\big(N(N-1)\big)$ PC scores scaled by a factor $(p/(N-1))^{1/2}$. While the raw data diverges substantially from the true background noise distribution, the scaled PCs approximate the noise distribution fairly well. Further, the scaled PC scores include a small number of large observations corresponding to true signal.

3.5 Background Noise Estimation

Accurate estimation of the background noise for the null Gaussian distribution is a critical part of both SigClust and SHC in the HDLSS setting. Liu et al. (2008) proposed using the robust estimator $\hat{\sigma}_{Raw}$ defined in (3.2). This approach relies on the assumption that a majority of the p variables are pure noise, as in microarray studies where expression is measured for thousands of genes, most of which are of no interest. However, in this section, we show that when this assumption does not hold, (3.2) can vastly overestimate the noise level and produce poor estimates of Λ, even when the underlying distribution follows a factor analysis model.

Consider the following motivating example with $N = 100$ observations drawn from a $p = 1024$ dimensional Gaussian distribution, $N(\mathbf{0}, \mathbf{\Sigma})$, where the 5 leading eigendirections of $\mathbf{\Sigma}$ corresponding to true signal are spread evenly across all 1024 dimensions. That is, the distribution lies stretched along 5 diagonals of the 1024 dimensional space. More formally, the covariance matrix $\mathbf{\Sigma}$ has

eigendecomposition $\mathbf{\Sigma} = U\Lambda U^T$ with eigenvalues λ_j, and corresponding eigenvectors $u_{\cdot j}$:

$$\lambda_j = \begin{cases} 1000 & \text{for } j \leq 5 \\ 1 & \text{for } j > 5 \end{cases}, \quad u_{\cdot j}^T \propto \begin{cases} \mathbf{e}_{2^1 0} & \text{for } j = 1 \\ \mathbf{e}_{2^{10-j}} \otimes [\mathbf{e}_{2^{j-1}}, -\mathbf{e}_{2^{j-1}}] & \text{for } j = 2, 3, 4, 5 \\ \text{arbitrary} & \text{for } j > 5 \end{cases}, \tag{3.4}$$

where $\mathbf{e}_a$ denotes the $(1 \times a)$ row vector of ones. We do not explicitly define the final 1019 directions spanned by the background noise, as their specific orientation is of no consequence. In Figure 3.2A, we show the distribution of all $(p \cdot N)$ values used to calculate $\hat{\sigma}_{Raw}$ as in (3.2). A kernel density estimate (KDE) is overlaid in solid blue. Additionally, the true background distribution, $N(0, 1)$, is overlaid with a dashed red line. In this toy example, the approach of Liu et al. (2008) estimates $\hat{\sigma}_{Raw} = 2.42$. Since the few directions of signal are spread over all dimensions, $\hat{\sigma}_{Raw}$ vastly overestimates the true background noise, $\sigma_b = 1$.

Estimating σ_b^2 using the input data fails in the current example since the true signal spans a non-trivial proportion of dimensions in the data. To address this problem, we propose estimating σ_b from the $\big(N(N-1)\big)$ non-zero principal component (PC) scores of the $p \times N$ data matrix, $\mathbb{X}$ (Figure 3.2B). Intuitively, since the directions of true signal approximately lie within the first few PC directions, a robust estimate of spread based on the PC scores should accurately target the background noise. Specifically, we propose the estimator:

$$\hat{\sigma}_{PC} = \frac{MAD_{N(N-1) \text{ scaled PC scores}}}{MAD_{N(0,1)}}. \tag{3.5}$$

using PC scores scaled by $(\frac{p}{N-1})^{1/2}$. The scaling factor is obtained from the following derivation under the trivial setting with identity covariance $\mathbf{\Sigma} = \sigma_b^2 \mathbf{I}$.

Let $\widetilde{\Lambda} = \text{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_p)$ denote the eigenvalues of the sample covariance matrix $\widetilde{\mathbf{\Sigma}}$ of the $p \times N$ row-centered sample matrix $\mathbb{X}_c$. Further, let $\mathbb{X}_c = U\big(N\widetilde{\Lambda}\big)^{1/2} V^T$ be the singular value decomposition of $\mathbb{X}_c$. Note that only the first $(n-1)$ diagonal elements of $\widetilde{\Lambda}^{1/2}$ and first $(N-1)$ columns of V are non-zero. Thus, the collection of PC scores are given by the first $(N-1)$ non-zero rows of $(n\widetilde{\Lambda})^{1/2} V^T$, denoted $P_{(N-1)\times N} = \{p_{ij}\}$. Then, the variance of the vectorized set of PC scores, p_{ij}, is given by:

$$\mathbb{E}\big((N(N-1))^{-1} \sum\nolimits_{i,j} p_{ij}^2\big) = (N(N-1))^{-1}\, \mathbb{E}\Big(\text{tr}(PP^T)\Big)$$

$$
\begin{aligned}
&= (N(N-1))^{-1}\ \mathbb{E}\Big(\mathrm{tr}((N\widetilde{\Lambda})^{1/2}V^TV(N\widetilde{\Lambda})^{1/2})\Big) \\
&= (N-1)^{-1}\ \mathbb{E}\Big(\mathrm{tr}(\widetilde{\Lambda})\Big) \\
&= (N-1)^{-1}\ \mathbb{E}\Big(\mathrm{tr}(\widetilde{\mathbf{\Sigma}})\Big) \\
&= (N-1)^{-1}\ \mathrm{tr}(\mathbf{\Sigma}) = \left(\frac{p}{N-1}\right)\sigma_b^2,
\end{aligned}
$$

since $\mathrm{tr}(\widetilde{\mathbf{\Sigma}})$ is an unbiased estimator of $\mathrm{tr}(\mathbf{\Sigma})$. However, it is not immediately obvious whether similar results hold under a factor model. Furthermore, since our proposed estimator $\hat{\sigma}_{PC}$ relies on rescaling by $MAD_{N(0,1)}$, the estimator may fair poorly if the PC scores, p_{ij}, do not approximately follow a Gaussian distribution. For the example shown in Figure 3.2, $\hat{\sigma}_{PC}$ provides a significant improvement over $\hat{\sigma}_{Raw}$. The PC score based estimator, $\hat{\sigma}_{PC}$, is used in all simulations and analyses reported in Sections 3.6 and 3.7. In the next section, we evaluate the power and level of SHC and other approaches through several simulation studies.

3.6 Simulations

In this section we illustrate the performance of our proposed SHC approach using simulation studies. In Section 3.3, we described SHC as the combination of two elements: (1) a sequential testing scheme for controlling the FWER applied to the results of hierarchical clustering, and (2) a simulation-based hypothesis test for assessing the statistical significance of hierarchical clustering at a single node. To evaluate the advantage of tuning the test at each node for hierarchical clustering, we consider two implementations of our SHC approach, denoted by SHC1 and SHC2. In SHC1, we combine our proposed iterative testing scheme with the classical SigClust test applied at each node. In SHC2, we implement our complete procedure, which directly accounts for the effect of hierarchical clustering in the calculation of the p-value. Two implementations of SHC2 are further considered, denoted by $\mathrm{SHC2}_L$ and $\mathrm{SHC2}_2$, differing by whether the linkage value or the 2-means CI is used to measure the strength of clustering. Note that both SHC1 and SHC2 may be viewed as contributions of our work with differing levels of adjustment for the hierarchical setting.

The performance of SHC1 and SHC2 are compared against the existing `pvclust` approach. In each simulation evaluating the performance of SHC, Ward's linkage clustering was applied to a dataset drawn from a mixture of Gaussian distributions in $\mathbb{R}^p$. A range of simulation settings were

considered, including the null setting with $K = 1$ and alternative settings with $K = 2$, $K = 3$, and $K = 4$. A representative set of simulation results for $K = 1$, $K = 3$ and $K = 4$ are reported in this section. As the $K = 2$ setting reduces to a non-nested clustering problem, these results are omitted from this section. However, complete simulation results, including the entire set of $K = 2$ results (Table 3.5), may be found in Section 3.10.

In all simulations evaluating the performance of SHC, SHC1 and SHC2 p-values were calculated using 100 simulated null cluster indices, and the corresponding Gaussian-fit p-values are reported. When $p > n$, the covariance matrix for the Gaussian null was estimated using the soft-threshold approach described in Huang et al. (2014). The `pvclust` approach was implemented using 1000 bootstrap samples, as suggested in Suzuki and Shimodaira (2006). However, to keep the total computational time of the entire set of simulations manageable, the complete set of simulations reported in Section 3.10 were completed using 100 bootstrap samples. Results for `pvclust` are only reported for high dimensional simulations, as the approach does not appear to be able to handle datasets in lower dimensions, e.g. $p = 2$. All simulation settings were replicated 100 times. Before presenting the simulation results, we first provide a brief review of the fundamental difference between `pvclust` and our proposed SHC method.

The `pvclust` method of Suzuki and Shimodaira (2006) computes two values: an approximately unbiased (AU) p-value based on a multi-step multi-scale bootstrap resampling procedure (Shimodaira, 2004), and a bootstrap probability (BP) p-value calculated from ordinary bootstrap resampling (Efron et al., 1996). Similar to SHC, `pvclust` also tests at nodes along the dendrogram. However, no test is performed at the root node, and the corresponding hypotheses tested at each node is given by:

$$H_0 : \text{ the cluster does not exist}$$

$$H_1 : \text{ the cluster exists.}$$

The difference between the two approaches can be understood by examining the dendrogram presented in Figure 3.1B. Using SHC, significant evidence of the three clusters is obtained if the null hypothesis is rejected at the top two nodes of the dendrogram. In contrast, to identify the three

Table 3.1: Simulation 3.6.1 ($K = 1$). Number of false positives at $\alpha = 0.05$, mean p-value, median computation time over 100 replications. $*$: `pvclust` times scaled by 1/10.

parameters			$\lvert p\text{-value} < 0.05\rvert$ (mean p-value)					median time (sec.)			
N	w	v	pvAU	pvBP	SHC1	$SHC2_L$	$SHC2_2$	pv	SHC1	$SHC2_L$	$SHC2_2$
50	0	–	0 (0.99)	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	30.61*	20.56	12.33	14.59
50	1	100	2 (0.48)	0 (0.98)	0 (0.59)	0 (0.49)	0 (0.47)	30.54*	22.31	13.35	15.60
50	5	100	1 (0.61)	0 (1.00)	0 (0.83)	0 (0.73)	0 (0.65)	30.52*	21.11	12.68	14.82
100	0	–	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	108.52*	48.18	29.19	35.04
100	1	100	2 (0.89)	0 (1.00)	0 (0.69)	0 (0.49)	0 (0.49)	108.70*	50.49	30.73	36.85
100	5	100	1 (0.98)	0 (1.00)	0 (0.96)	0 (0.72)	0 (0.72)	108.85*	51.04	30.74	37.01

clusters using `pvclust`, the null hypothesis must be rejected at the three nodes directly above each cluster, denoted by their respective cluster symbol.

3.6.1 Null Setting

We first considered the null setting to evaluate the ability of SHC to control for false positives. In these simulations, datasets of size $N = 50$ and 100 were sampled from a single Gaussian distribution in $p = 1000$ dimensions with diagonal covariance structure given by:

$$\mathbf{\Sigma} = \text{diag}\{\underbrace{v, \ldots, v}_{w}, \underbrace{1, \ldots, 1}_{p-w}\},$$

where the first w diagonal entries represent low dimensional signal in the data, of magnitude $v > 1$. A subset of the simulation results are presented in Table 3.1, with complete results provided in Table 3.4.

For `pvclust` AU and BP values, summaries are reported for tests at the second and third nodes from the root, i.e. $j = 2$ and $j = 3$. For both SHC1 and SHC2, summaries are reported for the p-value at the root node of each simulated dataset. Under each set of simulation parameters, for each method, we report the number of replications with false positive calls using a significance threshold of 0.05, as well as the mean p-value, and the median computing time of a single replication. For `pvclust`, a false positive was recorded if either of the two nodes was significant, and the mean p-value was calculated using both nodes. For a fair comparison of the computational times required by `pvclust` using 1000 bootstraps and the SHC procedures using 100 Monte Carlo simulations, we report the computational times of `pvclust` after scaling by 1/10. Only a single computing time is reported for `pvclust`, as the implementation computes both AU and BP values simultaneously.

Table 3.2: Simulation 3.6.2 ($K = 3$). Number of replications identifying the correct number of significant clusters, median computation time over 100 replications. $*$: scaled by 1/10.

parameters			$\lvert\hat{K} = 3\rvert$					median time (sec.)			
p	δ	arr.	pvAU	pvBP	SHC1	$SHC2_L$	$SHC2_2$	pv	SHC1	$SHC2_L$	$SHC2_2$
2	3	$\cdots$	–	–	18	0	29	–	2.42	1.08	1.95
2	4	$\cdots$	–	–	84	6	87	–	2.39	1.08	1.92
1000	8	$\cdots$	0	0	0	5	66	231.62^*	79.85	48.87	59.33
1000	12	$\cdots$	0	0	16	93	100	231.53^*	79.35	49.14	59.21
1000	20	$\cdots$	13	0	70	79	99	231.55^*	78.62	48.67	58.71
2	4	$\triangle$	–	–	26	32	84	–	2.40	1.07	1.97
2	5	$\triangle$	–	–	96	93	99	–	2.40	1.06	1.94
1000	8	$\triangle$	0	0	0	4	84	231.84^*	79.76	49.19	59.21
1000	12	$\triangle$	0	0	100	100	100	231.75^*	80.06	49.43	59.17
1000	20	$\triangle$	52	0	100	100	100	232.54^*	80.71	49.29	59.58

Since the data were generated from a single Gaussian distribution, we expect the SHC2 p-value at the root node to be approximately uniformly distributed over $[0, 1]$. In Table 3.1, all methods show generally conservative behavior, making less false positive calls than expected by chance. The `pvclust` BP value (pvBP) shows the most strongly conservative behavior, reporting mean p-values close to 1 for most settings. The remaining approaches, including the `pvclust` AU value (pvAU), and both SHC1 and SHC2, are consistently conservative across all settings considered. The conservative behavior of the classical SigClust procedure was previously described in Liu et al. (2008) and Huang et al. (2014) as being a result of the challenge of estimating the null eigenvalues and the corresponding covariance structure in the HDLSS setting (Baik and Silverstein, 2006). As both SHC1 and SHC2 rely on the same null covariance estimation procedure, this may also explain the generally conservative behavior similarly observed with our proposed approaches. Both SHC approaches required substantially less computational time than `pvclust`, even after correcting for the larger number of bootstrap samples required by the method.

3.6.2 Three Cluster Setting

We next considered the alternative setting in which datasets were drawn equally from three spherical Gaussian distributions each with covariance matrix $\mathbf{I}_p$. The setting illustrates the simplest case for which significance must be attained at multiple nodes to discern the true clustering structure from the dendrogram using SHC. Two arrangements of the three Gaussian components were studied. In the first, the Gaussian components were placed along a line with distance δ between the means of

neighboring components. In the second, the Gaussian components were placed at the corners of an equilateral triangle with side length δ. Several values of δ were used to evaluate the relative power of each method across varying levels of signal. Both low ($p = 2$) and high ($p = 1000$) dimensional settings were also considered. For each dataset, 50 samples were drawn from each of the three Gaussian components. As in Simulation 3.6.1, to make timing results comparable between `pvclust` and SHC, `pvclust` times are reported after scaling by 1/10. Select simulation results are presented in Table 3.2, with complete results presented in Tables 3.6 and 3.7. We report the number of replications out of 100 for which each method detected statistically significant evidence of three ($\hat{K} = 3$) clusters, as well as the median computation time across replications. For the two `pvclust` approaches, the numbers of predicted clusters were determined by the number of significant subtrees with at least $\lceil (3/4) \cdot 50 \rceil = 38$ observations. This criterion was used to minimize the effect of small spurious clusters reported as being significant by the methods. For SHC1 and SHC2, the numbers of predicted clusters were determined by the resulting number of subtrees after cutting a dendrogram at all significant nodes identified while controlling the FWER at 0.05.

In both arrangements of the components, the `pvclust` based methods showed substantially lower power than the proposed three approaches, with `pvBP` achieving no power at all. Across all settings reported in Table 3.2, SHC2_2 consistently achieves the greatest power. The relative performance of SHC2_L and SHC1 appears to depend on both the arrangement of the cluster components and the dimension of the dataset. When the components are arranged along a line, SHC2_L outperforms SHC1 in the high dimensional setting, while the performance is reversed in the low dimensional setting. In contrast, when the components are placed in the triangular arrangement, SHC2_L shows a slight advantage in both high and low dimensional settings. Timing results were comparable to those observed in Simulation 3.6.1, with `pvclust` requiring substantially more time than the other approaches, even after scaling. Again, SHC2_L required the least amount of computational time.

3.6.3 Four Cluster Setting

Lastly, we considered the alternative setting in which datasets were drawn equally from four spherical Gaussian distributions each with covariance matrix $\mathbf{I}_p$. Two arrangements of the Gaussian components were studied. In the first, the four components were placed at the vertices of a square

Table 3.3: Simulation 3.6.3 ($K = 4$). Number of replications identifying the correct number of significant clusters, median computation time over 100 replications. $*$: scaled by 1/10.

parameters			$\lvert\hat{K} = 4\rvert$					median time (sec.)			
p	δ	arr.	pvAU	pvBP	SHC1	$SHC2_L$	$SHC2_2$	pv	SHC1	$SHC2_L$	$SHC2_2$
2	3	square	–	–	3	0	17	–	2.54	1.16	2.06
2	4	square	–	–	78	12	90	–	2.57	1.16	2.07
1000	8	square	0	0	0	0	75	401.83*	110.50	69.01	82.59
1000	10	square	0	0	97	100	100	400.36*	110.79	69.30	82.67
3	4	tetra.	–	–	0	9	33	–	2.84	1.28	2.25
3	5	tetra.	–	–	24	86	99	–	2.40	1.09	2.03
1000	8	tetra.	0	0	0	0	31	402.00*	113.49	71.03	85.27
1000	10	tetra.	0	0	50	99	100	399.85*	113.67	71.17	85.19

with side length δ. In the second, the four components were placed at the vertices of a regular tetrahedron, again with side length δ. As in Simulation 3.6.2, for each dataset, 50 samples were drawn from each of the Gaussian components. A representative subset of simulation results are presented in Table 3.3 for several values of p and δ, with complete results presented in Tables 3.8 and 3.9. In Section 3.10, we also include simulation results for a rectangular arrangement with side lengths δ and $(3/2) \cdot \delta$ (Table 3.10), and a stretched tetrahedral arrangement, also having side lengths δ and $(3/2) \cdot \delta$ (Table 3.11).

The results presented in Table 3.3 largely support the results observed in Simulation 3.6.2. Again, the pvAU and pvBP values provide little power to detect significant clustering in the data, while $SHC2_2$ consistently achieves the greatest power. Additionally, the relative performance of $SHC2_L$ and SHC1 again depends on the arrangement of the components and the dimension of the dataset. In the square arrangement, while SHC1 performs better in the low dimensional setting, the approaches perform equally well in high dimensions. In the tetrahedral arrangement, $SHC2_L$ achieves substantially greater power than SHC1 in both high and low dimensional settings.

3.7 Real Data Analysis

To further demonstrate the power of SHC, we applied the approach to two cancer gene expression datasets. We first considered a dataset of 300 tumor samples drawn from three distinct cancer types: head and neck squamous cell carcinoma (HNSC), lung squamous cell carcinoma (LUSC), and lung adenocarcinoma (LUAD). As distinct cancers, we expect observations from the three groups to be easily separated by hierarchical clustering and detected by SHC. For the second

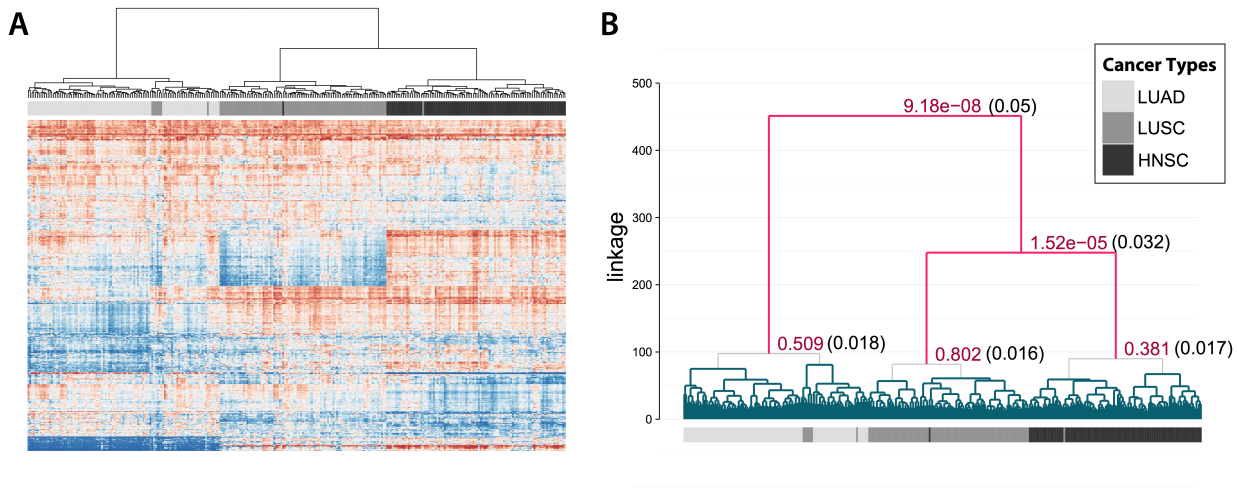

Figure 3.3: Analysis of gene expression for 300 LUAD, LUSC, and HNSC samples. (A) Heatmap of log-transformed gene expression for the 300 samples (columns), clustered by Ward's linkage. (B) Dendrogram with corresponding SHC p-values (red) and α^* cutoffs (black) given only at nodes tested according to the FWER controlling procedure at $\alpha = 0.05$.

dataset, we considered a cohort of 337 breast cancer (BRCA) samples, previously categorized into five molecular subtypes (Parker et al., 2009). The greater number of subpopulations, as well as the more subtle differences between them, makes this dataset more challenging than the first. In both examples, the data were clustered using Ward's linkage and the $\text{SHC}2_2$ approach was implemented as described in Section 3.6 using 1000 simulations at each node. The FWER controlling procedure of Section 3.3.2 was applied with $\alpha = 0.05$.

3.7.1 Multi-Cancer Gene Expression

A dataset of 300 samples was constructed by combining 100 samples from each of HNSC, LUSC and LUAD made publicly available by the TCGA Research Network (The Cancer Genome Atlas Research Network, 2012, 2014). Gene expression was estimated for 20,531 genes from RNA-seq data using RSEM (Li and Dewey, 2011), as described in the TCGA RNA-seq v2 pipeline (`https://wiki.nci.nih.gov/display/TCGA/RNASeq+Version+2`). To adjust for technical effects of the data collection process, expression values were first normalized using the upper-quartile procedure of Bullard et al. (2010). Then, all expression values of zero were replaced by the smallest non-zero expression value across all genes and samples. A subset of 500 most variably expressed genes were selected according to the median absolute deviation about the median (MAD) expression across all samples. Finally, SHC was applied to the log-transformed expression levels at the 500 most

variable loci. Similar results were also obtained when using the 100, 1000, and 2000 most variable genes.

In Figure 3.3A, the log-transformed expression values are visualized using a heatmap, with rows corresponding to genes, and columns corresponding to samples. Lower and higher expression values are shown in blue and red, respectively. For easier visual interpretation, rows and columns of the heatmap were independently clustered using Ward's linkage clustering. The corresponding dendrogram and cancer type labels are shown above the heatmap. The dendrogram and labels in Panel A of Figure 3.3 are reproduced in Panel B, along with the SHC p-values (red) and modified significance cutoffs (black) at nodes tested according to the FWER controlling procedure. Branches corresponding to statistical significant nodes and untested nodes are shown in red and blue, respectively. Ward's linkage clustering correctly separates the three cancer types, with the exception of seven LUSC samples clustered with the LUAD samples, one LUSC sample clustered with HNSC, and one HNSC sample clustered with LUSC. Interestingly, the LUSC and HNSC samples cluster prior to joining with the LUAD samples, suggesting the greater molecular similarity between squamous cell tumors of different sites, than different cancers of the lung. This agrees with the recently identified genomic similarity of the two tumors reported in Hoadley et al. (2014). Furthermore, we note that no HNSC and LUAD samples are jointly clustered, highlighting the clear difference between tumors of both distinct histology and site. As shown in Figure 3.3B, statistically significant evidence of clustering was determined at the top two nodes, with respective Gaussian-fit p-values $9.18e-8$ and $1.52e-5$ at the modified significance cutoffs, $\alpha_1^* = 0.05$ and $\alpha_2^* = 0.032$. Additionally, the three candidate nodes corresponding to splitting each of the cancer types all give insignificant results, suggesting no further clustering in the cohort. Finally, we note that when analyzed using `pvclust`, no statistically significant evidence of clustering was found, with AU p-values of 0.26, 0.28, and 0.13 obtained at the three nodes corresponding to primarily LUAD, LUSC and HNSC samples.

3.7.2 Breast Cancer (BRCA) Gene Expression

As a second example, we considered a microarray gene expression dataset from 337 BRCA samples. The dataset was compiled, filtered and normalized as described in Prat et al. (2010) and obtained from the University of North Carolina (UNC) Microarray Database (`https://genome.unc.edu/`

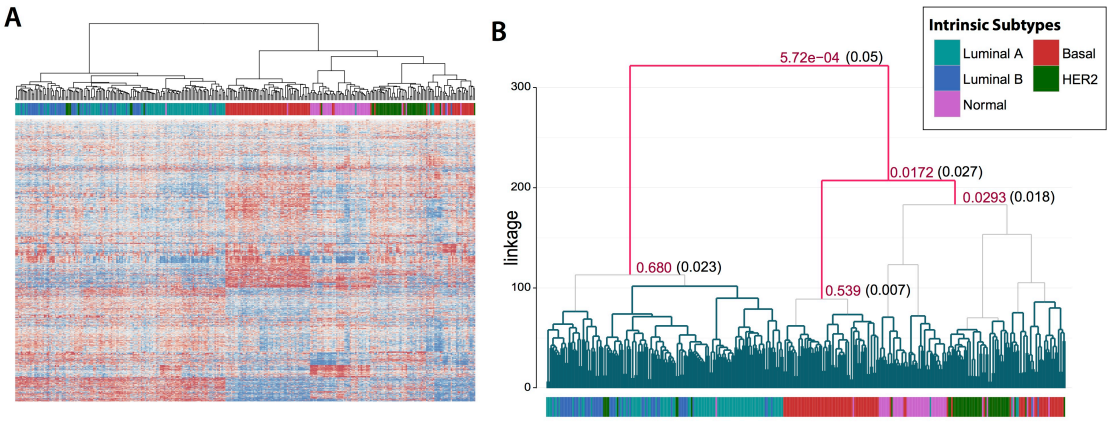

Figure 3.4: Analysis of gene expression for 337 BRCA samples. (A) Heatmap of gene expression for the 337 samples (columns) clustered by Ward's linkage. (B) Dendrogram with corresponding SHC p-values (red) and α^* cutoffs (black) given only at nodes tested according to the FWER controlling procedure at $\alpha = 0.05$.

`pubsup/clow/`). Gene expression was analyzed for a subset of 1645 well-chosen *intrinsic* genes identified in Prat et al. (2010). We evaluate the ability of our approach to detect biologically relevant clustering based on five molecular subtypes: luminal A (LumA), luminal B (LumB), basal-like, normal breast-like, and HER2-enriched (Parker et al., 2009). The dataset is comprised of 97 LumA, 54 LumB, 91 basal-like, 47 normal breast-like, and 48 HER2-enriched samples.

The expression dataset is shown as a heatmap in Figure 3.4A, with the corresponding dendrogram and subtype labels reproduced in Figure 3.4B. The corresponding SHC p-values (red) and modified significance thresholds (black) are again given at only nodes tested while controlling the FWER at $\alpha = 0.05$. SHC identifies at least three significantly differentiated clusters in the dataset, primarily corresponding to luminal (LumA and LumB), basal-like, and all remaining subtypes. At the root node, the LumA and LumB samples are separated from the remaining subtypes with a p-value of $5.72e-4$ at a threshold of $\alpha_1^* = 0.05$. However, Ward's linkage clustering and SHC are unable to identify significant evidence of clustering between the two luminal subtypes. The difficultly of clustering LumA and LumB subtypes based on gene expression was previously described in Mackay et al. (2011). Next, the majority of basal-like samples are separated from the remaining set of observations, with a p-value of 0.0172 at a cutoff of $\alpha_2^* = 0.027$. The remaining HER2-enriched, normal breast-like and basal-likes samples show moderate separation by Ward's linkage clustering. However, controlling the FWER at $\alpha = 0.05$, the subsequent node is non-significant, with a p-value of 0.0293 against a corrected threshold of $\alpha_3^* = 0.0180$. This highlights the difficulty of assessing

statistical significance in the presence of larger numbers of clusters, while controlling for multiple testing.

3.8 Discussion

While hierarchical clustering has become widely popular in practice, few methods have been proposed for assessing the statistical significance of a hierarchical partition. SHC was developed to address this problem, using a sequential testing and FWER controlling procedure. Through an extensive simulation study, we have shown that SHC provides competitive results compared to existing methods. Furthermore, in applications to two gene expression datasets, we showed that the approach is capable of identifying biologically meaningful clustering.

In this chapter, we focused on the theoretical and empirical properties of SHC using Ward's linkage. However, there exist several different approaches to hierarchical clustering, and Ward's linkage may not always be the most appropriate choice. In these situations, as mentioned in Section 3.3, SHC may be implemented with other linkage and dissimilarity functions which satisfy mean shift and rotation invariance. Further investigation is necessary to fully characterize the behavior of the approach for different hierarchical clustering procedures.

Some popular choices of dissimilarity, such as those based on Pearson correlation of the covariates between pairs of samples, fail to satisfy the necessary mean shift and rotation invariance properties in the original covariate space. As a consequence, the covariance of the Gaussian null distribution must be fully estimated, and cannot be approximated using only the eigenvalues of the sample covariance matrix. When $N \gg p$, the SHC method can still be applied by estimating the complete covariance matrix. However, in HDLSS settings, estimation of the complete covariance matrix can be difficult and computationally expensive. A possible direction of future work is the development of a computationally efficient procedure for non-invariant hierarchical clustering procedures.

3.9 Proofs

3.9.1 Proof of Theorem 3.1

Let $d_W(\cdot,\cdot)$ denote the Ward's linkage function defined over sets of observation indices. Additionally, let $\mathbb{X}^{(1)}$ and $\mathbb{X}^{(2)}$ denote n and m samples drawn from two Gaussian components with distributions $N(\boldsymbol{\mu}_1, \sigma_1^2\mathbf{I}_p)$ and $N(\boldsymbol{\mu}_2, \sigma_2^2\mathbf{I}_p)$, with corresponding observation index sets, $\mathbb{C}^{(1)}$ and $\mathbb{C}^{(2)}$. For $k = 1, 2$, let $C_0^{(k)}$, $C_1^{(k)}$ and $C_2^{(k)}$ denote subsets of $\mathbb{C}^{(k)}$, where $C_1^{(k)}$ and $C_2^{(k)}$ are necessarily disjoint. Let $n_0 = |C_0^{(1)}|$, $n_1 = |C_1^{(1)}|$, $n_2 = |C_2^{(1)}|$, $m_0 = |C_0^{(2)}|$, $m_1 = |C_1^{(2)}|$ and $m_2 = |C_2^{(2)}|$ denote the size of each subset. Finally, let $\boldsymbol{X}_0^{(k)}$, $\boldsymbol{X}_1^{(k)}$, and $\boldsymbol{X}_2^{(k)}$, denote the corresponding subsets of $\mathbb{X}^{(k)}$, with corresponding sample means, $\overline{\boldsymbol{X}}_0^{(k)}$, $\overline{\boldsymbol{X}}_1^{(k)}$, and $\overline{\boldsymbol{X}}_2^{(k)}$.

Consider the two events: $A = \{\max d_W(C_1^{(1)}, C_2^{(1)}) < \min d_W(C^{(1)}, C^{(2)})\}$, and $B = \{\max d_W(C_1^{(2)}, C_2^{(2)}) < \min d_W(C^{(1)}, C^{(2)})\}$, where maxima and minima are taken with respect to the possible values of $C_1^{(k)}$, $C_2^{(k)}$, and $C^{(k)}$. Note that the joint occurrence of A and B is sufficient for correctly separating observations from the two components at the root node. It therefore suffices to show that $P(A \cap B) \to 1$ as $p \to \infty$.

For some $0 < a_1 \leq a_2$, define the following events: $E_1 = \{\max d_W(C_1^{(1)}, C_2^{(1)}) < a_1 \cdot p\}$, $E_2 = \{\max d_W(C_1^{(2)}, C_2^{(2)}) < a_1 \cdot p\}$, $E_3 = \{\min d_W(C_0^{(1)}, C_0^{(2)}) > a_2 \cdot p\}$. Note that the probability of the joint event $(A \cap B)$ can be bounded below by:

$$\begin{aligned} P(A \cap B) &= 1 - P(A^C \cup B^C) \\ &\geq 1 - \big(P(E_1^C) + P(E_2^C) + P(E_3^C)\big). \end{aligned}$$

We complete the proof by showing that $P(E_1^C)$, $P(E_2^C)$, $P(E_3^C)$ all tend to 0 as $p \to \infty$.

By Lemmas 3 and 4 of Borysov et al. (2014) for $a_1 > 2\sigma_1^2$, we have

$$\begin{aligned} P(E_1^C) &= P(\max d_W(C_1^{(1)}, C_2^{(1)}) > a_1 \cdot p) \\ &= P\left(\max \frac{2n_1n_2}{n_1+n_2} \left\|\overline{\boldsymbol{X}}_1^{(1)} - \overline{\boldsymbol{X}}_2^{(1)}\right\|^2 > a_1 \cdot p\right) \\ &\leq 3^n P\left(\left\|\left(\frac{2n_1n_2}{n_1+n_2}\right)^{1/2}\left(\overline{\boldsymbol{X}}_1^{(1)} - \overline{\boldsymbol{X}}_2^{(1)}\right)\right\|^2 > a_1 \cdot p\right) \\ &\leq e^{-c_1 p + n\log 3}, \end{aligned}$$

where $c_1 = a_1/\sigma_1^2 - (1 + \log(a_1/\sigma_1^2))$. Note that since $c_1 > 0$ and $n = o(p) + p^\alpha$ for some $\alpha \in (0, 1)$, $P(E_1^C) \to 0$ as $p \to \infty$. Similarly, for $a_2 > 2\sigma_2^2$, we have

$$P(E_2^C) \leq e^{-c_2 p + m \log 3},$$

where $c_2 = a_2/\sigma_2^2 - (1 + \log(a_2/\sigma_2^2))$, such that $P(E_2^C) \to 0$ as $p \to \infty$. Finally, to bound $P(E_3^C)$, we make use of Lemmas 2 and 4 of Borysov et al. (2014):

$$\begin{aligned} P(E_3^C) &= P(\min d_W(C_0^{(1)}, C_0^{(2)}) < a_2 \cdot p) \\ &\leq \sum_{i=1}^{n} \sum_{j=1}^{m} P\left(\frac{2ij}{i+j} \left\| \overline{\boldsymbol{X}}_0^{(1)} - \overline{\boldsymbol{X}}_0^{(2)} \right\|^2 < a_2 \cdot p \right) \\ &\leq 2^{n+m} \max_{i \leq n,\ j \leq m} P\left(\frac{2ij}{i+j} \left\| \overline{\boldsymbol{X}}_0^{(1)} - \overline{\boldsymbol{X}}_0^{(2)} \right\|^2 < a_2 \cdot p \right). \end{aligned}$$

Suppose that i and j are fixed, and let $\mu^2 = p^{-1}(\frac{2ij}{i+j})\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|^2$, $\mu_k = (\frac{2ij}{i+j})^{1/2}(\boldsymbol{\mu}_{1,k} - \boldsymbol{\mu}_{2,k})$, $\sigma^2 = (\frac{2ij}{i+j})(\frac{i\sigma_2^2 + j\sigma_1^2}{ij})$. Then, using the result of Lemma 2 of Borysov et al. (2014), for $0 < a_2 < \sigma^2 + \mu^2$, we have

$$P\left(\frac{2ij}{i+j} \left\| \overline{\boldsymbol{X}}_0^{(1)} - \overline{\boldsymbol{X}}_0^{(2)} \right\|^2 < a_2 \cdot p \right) \leq e^{-c_3 p},$$

where $c_3 = (a_2 - \sigma^2 - \mu^2)^2/(6\sigma^4 + 12\sigma^2\mu^2 + 2p^{-1}\sum_{k=1}^{p} \mu_k^4)$.

Using the fourth moment bound of (2), and the fact that $n = o(p) + p^\alpha$, $m = o(p) + p^\beta$, we have that $P(E_3^C) \to 0$ as $p \to \infty$. Thus, for $a_1 > 2\sigma_1^2$, $P(E_1^C) \to 0$, for $a_1 > 2\sigma_2^2$, $P(E_2^C) \to 0$, and for $a_2 < (\frac{2nm}{n+m})(\frac{\sigma_1^2}{n} + \frac{\sigma_2^2}{m} + \frac{\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|^2}{p})$, $P(E_3^C) \to 0$. Combining the necessary inequalities on a_1, a_2, we obtain the stated condition:

$$\begin{aligned} 2\sigma_2^2 < a_1 \leq a_2 & \\ &< \frac{2nm}{n+m}\left(\frac{\sigma_1^2}{n} + \frac{\sigma_2^2}{m} + \frac{\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|^2}{p} \right) \\ \frac{1}{n}(\sigma_2^2 - \sigma_1^2) &< \frac{\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|^2}{p}. \end{aligned}$$

3.9.2 Proof of Theorem 3.2

Let $d_W(\cdot, \cdot)$ denote the Ward's linkage function. Further, let $\mathbb{X}^{(1)}$ and $\mathbb{X}^{(2)}$ denote the n and m observations from the first and second Gaussian components with distributions $N(\boldsymbol{\mu}_1, \sigma_1^2 \mathbf{I}_p)$ and

$N(\boldsymbol{\mu}_2, \sigma_2^2 \mathbf{I}_p)$. Assume that $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, σ_1^2 and σ_2^2 are known. Then, the theoretical best fit Gaussian to the mixture distribution is equivalent (up to a mean shift and rotation) to $N(\mathbf{0}, \widehat{\boldsymbol{\Sigma}})$, where

$$\begin{aligned}
\widehat{\boldsymbol{\Sigma}} &= \text{diag}\{\widehat{\lambda}_k\}_{k=1}^p \\
\widehat{\lambda}_1 &= \frac{nm}{n+m}\left((n+m)^{-1}\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|^2 + \frac{\sigma_1^2}{m} + \frac{\sigma_2^2}{n}\right) \\
\widehat{\lambda}_k &= \frac{nm}{n+m}\left(\frac{\sigma_1^2}{m} + \frac{\sigma_2^2}{n}\right), \quad \text{for } k \geq 2,
\end{aligned}$$

where the $\widehat{\lambda}_k$ are derived by the formula for the variance of a univariate mixture of Gaussians. Let $\mathbb{X}^{(3)}$ denote a sample of $n+m$ observations drawn from $N(\mathbf{0}, \widehat{\boldsymbol{\Sigma}})$. Let $\mathbb{C}^{(k)}$ denote the corresponding observation indices for $k = 1, 2, 3$. Additionally, let $C_1^{(3)}$ and $C_2^{(3)}$ denote disjoint subsets of $\mathbb{C}^{(3)}$, and let $r_1 = |C_1^{(3)}|$ and $r_2 = |C_2^{(2)}|$ denote the size of the subsets. Finally, let $\boldsymbol{X}_1^{(3)}$ and $\boldsymbol{X}_2^{(3)}$ denote the corresponding subsets of $\mathbb{X}^{(3)}$ with means $\overline{\boldsymbol{X}}_1^{(3)}$ and $\overline{\boldsymbol{X}}_2^{(3)}$.

Consider the event: $D = \{\max d_W(C_1^{(3)}, C_2^{(3)}) < d_W(\mathbb{C}^{(1)}, \mathbb{C}^{(2)})\}$, where the maximum is taken with respect the possible values of $C_1^{(3)}$ and $C_2^{(3)}$. By Theorem 1 we have that, asymptotically, Ward's linkage clustering achieves the correct partition of $\mathbb{C}^{(1)}$ and $\mathbb{C}^{(2)}$. Therefore, D is precisely the event that a linkage value simulated from the null distribution is less than the observed linkage value. The proof is completed by showing $P(D) \to 1$ as $p \to \infty$. That is, we wish to show that the empirical p-value tends to 0 as $p \to \infty$.

For some $a > 0$, define the following events: $E_4 = \{\max d_W(C_1^{(3)}, C_2^{(3)}) < a \cdot p\}$, and $E_5 = \{d_W(\mathbb{C}^{(1)}, \mathbb{C}^{(2)}) > a \cdot p\}$. Note that $P(D)$ can be bounded below by:

$$\begin{aligned}
P(D) &= 1 - P(D^C) \\
&\geq 1 - \big(P(E_4^C) + P(E_5^C)\big).
\end{aligned}$$

Thus, it suffices to show the probabilities of E_4^C, E_5^C,both tend to 0 as $p \to \infty$.

First, we state a generalization of Lemma 3 from Borysov et al. (2014) for Gaussian distributions with diagonal covariance.

Lemma 3.1. *Suppose n independent observations, $\mathbb{X}$, are drawn from the p-dimensional Gaussian distribution, $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a diagonal matrix with diagonal entries $\{\lambda_k\}_{k=1}^p$. Define scalars*

$\mu^2 = p^{-1}\|\boldsymbol{\mu}\|^2$, $\overline{\lambda} = p^{-1}\sum_{k=1}^{p}\lambda_k$, *and let* $a > \overline{\lambda} + \mu^2$. *Then, for any* $0 < i \leq n$,

$$P(\|X_i\|^2 > a \cdot p) \leq e^{-cp},$$

where $c = \left[a + \mu^2 - \sqrt{\overline{\lambda}^2 + 4\mu^2 a} + \overline{\lambda}\left(\frac{\overline{\lambda}+\sqrt{\overline{\lambda}^2+4\mu^2 a}}{2a}\right)\right]/\overline{\lambda}$.

The proof of Lemma 3.1 is omitted as it follows exactly as that of Lemma 3 from Borysov et al. (2014). By Lemma 3.1 given above and Lemma 4 of Borysov et al. (2014) , for $a > 2\overline{\widehat{\lambda}}$, where $\overline{\widehat{\lambda}} = p^{-1}\sum_{k=1}^{p}\widehat{\lambda}_k$, we have

$$\begin{aligned}
P(E_4^C) &= P(\max d_W(C_1^{(3)}, C_2^{(3)}) > a \cdot p) \\
&= P\left(\max \frac{2r_1 r_2}{r_1 + r_2}\left\|\overline{\boldsymbol{X}}_1^{(3)} - \overline{\boldsymbol{X}}_2^{(3)}\right\|^2 > a \cdot p\right) \\
&\leq 3^{n+m} P\left(\left\|\left(\frac{2r_1 r_2}{r_1 + r_2}\right)^{1/2}\left(\overline{\boldsymbol{X}}_1^{(3)} - \overline{\boldsymbol{X}}_2^{(3)}\right)\right\|^2 > a \cdot p\right) \\
&\leq e^{-c_4 p + (n+m)\log 3},
\end{aligned}$$

where $c_4 = a/\overline{\widehat{\lambda}} - (1 + \log(a/\overline{\widehat{\lambda}}))$. As for $P(E_1^C)$ from the proof of Theorem 2, we have that as $p \to \infty$, $P(E_4^C) \to 0$ as $p \to 0$. Next, using an argument similar to the one presented above for $P(E_3^C) \to 0$, we show that $P(E_5^C) \to 0$. Let $\mu^2 = p^{-1}(\frac{2nm}{n+m})\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|^2$, $\mu_k = (\frac{2nm}{n+m})^{1/2}(\boldsymbol{\mu}_{1,k} - \boldsymbol{\mu}_{2,k})$, $\sigma^2 = (\frac{2nm}{n+m})(\frac{i\sigma_2^2 + j\sigma_1^2}{nm})$. Then, by Lemmas 2 and 4 of Borysov et al. (2014), for $0 < a < \sigma^2 + \mu^2$, we have

$$\begin{aligned}
P(E_4^C) &= P(d_W(\mathbb{C}^{(1)}, \mathbb{C}^{(2)}) < a \cdot p) \\
&= P\left(\frac{2nm}{n+m}\left\|\overline{\boldsymbol{X}}^{(1)} - \overline{\boldsymbol{X}}^{(2)}\right\|^2 < a \cdot p\right) \\
&\leq e^{-c_5 p},
\end{aligned}$$

where $c_5 = (a - \sigma^2 - \mu^2)^2/(6\sigma^4 + 12\sigma^2\mu^2 + 2p^{-1}\sum_{k=1}^{p}\mu_k^4)$. As for $P(E_3^C)$ from the proof of Theorem 2, we have $P(E_5^C) \to 0$ as $p \to \infty$. Thus, for $a > 2\overline{\widehat{\lambda}}$, $P(E_4^C) \to 0$, and for $a < (\frac{2nm}{n+m})(\frac{\sigma_1^2}{n} + \frac{\sigma_2^2}{m} + \frac{\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|^2}{p})$, $P(E_5^C) \to 0$. Combining the two inequalities on a, we obtain the stated condition:

$$2p^{-1}\sum_{k=1}^{p}\widehat{\lambda}_k < a$$

$$< \left(\frac{2nm}{n+m}\right)\left(\frac{\sigma_1^2}{n}+\frac{\sigma_2^2}{m}+\frac{\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|^2}{p}\right)$$

$$\frac{\sigma_1^2}{m}+\frac{\sigma_2^2}{n}+(n+m)^{-1}\cdot\frac{\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|^2}{p} < \frac{\sigma_1^2}{n}+\frac{\sigma_2^2}{m}+\frac{\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|^2}{p}$$

$$\frac{(m^2-n^2)(\sigma_2^2-\sigma_1^2)}{nm(n+m-1)} < \frac{\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|^2}{p}.$$

3.10 Additional Simulation Results

Table 3.4: Complete results for Simulation 3.6.1. Number of false positives at $\alpha = 0.05$, mean p-value, median computation time over 100 replications.

parameters			$\lvert p\text{-value} < 0.05\rvert$ (mean p-value)					median time (sec.)			
n	w	v	pvAU	pvBP	SHC1	$SHC2_L$	$SHC2_2$	pv	SHC1	$SHC2_L$	$SHC2_2$
50	1	1	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	25.17	17.48	9.62	11.98
50	1	10	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	25.69	16.75	9.58	11.91
50	1	25	0 (0.95)	0 (1.00)	0 (0.89)	0 (0.91)	0 (0.70)	24.16	15.87	9.40	11.32
50	1	100	3 (0.61)	0 (0.98)	0 (0.60)	0 (0.49)	0 (0.46)	24.59	17.24	9.96	12.22
50	1	500	4 (0.28)	0 (0.85)	0 (0.53)	0 (0.47)	1 (0.44)	24.46	17.53	9.93	12.03
50	1	1000	13 (0.19)	0 (0.74)	0 (0.57)	0 (0.50)	0 (0.47)	23.40	16.97	9.81	11.42
100	1	1	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	85.48	37.28	21.69	26.12
100	1	10	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	0 (0.98)	88.23	38.71	21.95	26.97
100	1	25	0 (1.00)	0 (1.00)	0 (0.88)	0 (0.59)	0 (0.46)	81.09	38.12	22.19	26.86
100	1	100	0 (0.95)	0 (1.00)	0 (0.65)	0 (0.46)	0 (0.45)	87.60	37.86	22.18	28.07
100	1	500	4 (0.50)	0 (0.95)	0 (0.63)	0 (0.49)	0 (0.47)	82.48	38.35	23.46	27.62
100	1	1000	5 (0.37)	0 (0.91)	1 (0.62)	0 (0.49)	1 (0.46)	84.30	39.03	22.65	27.49
200	1	1	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	318.44	83.78	51.00	63.95
200	1	10	0 (1.00)	0 (1.00)	0 (1.00)	0 (0.86)	1 (0.68)	313.89	86.42	51.24	63.27
200	1	25	0 (1.00)	0 (1.00)	0 (0.97)	0 (0.50)	0 (0.49)	300.94	86.42	50.56	63.38
200	1	100	0 (1.00)	0 (1.00)	0 (0.77)	0 (0.49)	0 (0.48)	290.64	81.59	48.98	59.55
200	1	500	0 (0.93)	0 (1.00)	0 (0.69)	0 (0.49)	0 (0.46)	270.83	80.69	47.29	59.16
200	1	1000	0 (0.76)	0 (0.98)	0 (0.72)	0 (0.51)	1 (0.49)	264.99	80.24	46.28	58.00
50	5	10	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	21.40	14.53	8.24	9.72
50	5	25	0 (0.97)	0 (1.00)	0 (0.92)	0 (0.92)	0 (0.76)	21.45	14.66	8.29	9.80
50	5	100	4 (0.85)	0 (1.00)	0 (0.84)	0 (0.73)	0 (0.68)	21.25	14.48	8.27	9.91
50	5	500	7 (0.66)	0 (0.99)	0 (0.91)	0 (0.77)	0 (0.79)	21.30	14.53	8.28	9.82
50	5	1000	6 (0.58)	0 (0.99)	0 (0.92)	0 (0.75)	0 (0.79)	21.31	14.60	8.40	9.95
100	5	10	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	0 (0.92)	73.15	33.96	18.93	23.42
100	5	25	0 (1.00)	0 (1.00)	0 (0.96)	0 (0.73)	1 (0.67)	72.81	34.26	19.04	23.45
100	5	100	0 (1.00)	0 (1.00)	0 (0.95)	0 (0.72)	0 (0.72)	72.71	34.30	19.06	23.54
100	5	500	0 (0.98)	0 (1.00)	0 (0.97)	0 (0.76)	0 (0.78)	72.99	34.25	19.15	23.53
100	5	1000	1 (0.88)	0 (1.00)	0 (0.97)	0 (0.73)	0 (0.76)	72.41	34.49	19.78	23.81
200	5	10	0 (1.00)	0 (1.00)	0 (1.00)	0 (0.89)	0 (0.82)	259.63	76.34	44.28	55.53
200	5	25	0 (1.00)	0 (1.00)	0 (1.00)	0 (0.72)	0 (0.71)	259.76	76.57	44.01	55.23
200	5	100	0 (1.00)	0 (1.00)	0 (1.00)	0 (0.73)	0 (0.73)	274.21	77.72	45.59	56.19
200	5	500	0 (1.00)	0 (1.00)	0 (0.99)	1 (0.74)	1 (0.75)	281.71	77.21	46.14	56.43
200	5	1000	0 (1.00)	0 (1.00)	0 (1.00)	0 (0.73)	0 (0.74)	275.00	76.04	45.95	55.88

Table 3.5: Complete results for the $K = 2$ alternative setting. Number of replications identifying the correct number of significant clusters, mean p-value, median computation time over 100 replications.

parameters			$\lvert p\text{-value} < 0.05\rvert$ (mean p-value)					median time (sec.)			
n_k	p	δ	pvAU	pvBP	SHC1	$SHC2_L$	$SHC2_2$	pv	SHC1	$SHC2_L$	$SHC2_2$
50	2	1	–	–	0 (0.81)	0 (0.55)	0 (0.55)	–	1.37	0.54	1.13
50	2	2	–	–	1 (0.56)	1 (0.36)	13 (0.32)	–	1.37	0.55	1.13
50	2	3	–	–	67 (0.10)	18 (0.13)	77 (0.04)	–	1.42	0.56	1.12
50	2	4	–	–	98 (0.00)	81 (0.04)	99 (0.00)	–	1.42	0.54	1.21
50	2	5	–	–	100 (0.00)	98 (0.01)	100 (0.00)	–	1.36	0.51	1.08
100	2	1	–	–	0 (0.88)	0 (0.56)	0 (0.56)	–	2.89	1.20	2.50
100	2	2	–	–	2 (0.59)	1 (0.31)	14 (0.27)	–	3.02	1.16	2.62
100	2	3	–	–	80 (0.07)	53 (0.07)	86 (0.03)	–	2.85	1.27	2.31
100	2	4	–	–	100 (0.00)	98 (0.01)	100 (0.00)	–	2.92	1.25	2.37
100	2	5	–	–	100 (0.00)	100 (0.00)	100 (0.00)	–	2.86	1.18	2.52
50	1000	2	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	79.05	35.82	20.92	24.98
50	1000	4	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	78.60	34.99	20.49	24.73
50	1000	6	0 (1.00)	0 (1.00)	0 (0.99)	0 (1.00)	1 (0.81)	78.51	34.41	20.05	24.20
50	1000	8	1 (0.99)	0 (1.00)	18 (0.23)	28 (0.14)	97 (0.01)	78.51	34.95	20.51	24.41
50	1000	10	1 (0.68)	0 (0.92)	99 (0.00)	100 (0.00)	100 (0.00)	78.76	35.35	20.62	24.57
50	1000	12	1 (0.50)	0 (0.68)	100 (0.00)	100 (0.00)	100 (0.00)	78.68	35.03	20.51	24.64
50	1000	14	6 (0.25)	0 (0.52)	100 (0.00)	100 (0.00)	100 (0.00)	78.36	35.04	19.92	24.41
50	1000	16	48 (0.11)	0 (0.42)	100 (0.00)	100 (0.00)	100 (0.00)	77.75	34.63	19.76	24.12
50	1000	18	75 (0.07)	0 (0.39)	100 (0.00)	100 (0.00)	100 (0.00)	75.94	34.79	19.69	24.11
50	1000	20	84 (0.12)	0 (0.42)	100 (0.00)	100 (0.00)	100 (0.00)	74.95	34.55	19.61	23.76
100	1000	2	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	342.34	85.62	51.24	62.91
100	1000	4	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	0 (1.00)	329.18	86.59	50.56	62.71
100	1000	6	0 (1.00)	0 (1.00)	0 (0.95)	21 (0.29)	57 (0.10)	320.53	86.29	49.44	62.58
100	1000	8	0 (1.00)	0 (1.00)	84 (0.03)	100 (0.00)	100 (0.00)	311.57	86.83	51.29	65.07
100	1000	10	0 (0.81)	0 (0.97)	100 (0.00)	100 (0.00)	100 (0.00)	314.98	90.41	50.89	63.54
100	1000	12	1 (0.58)	0 (0.75)	100 (0.00)	100 (0.00)	100 (0.00)	321.48	85.76	50.40	60.86
100	1000	14	0 (0.29)	0 (0.51)	100 (0.00)	100 (0.00)	100 (0.00)	314.38	86.93	50.78	63.36
100	1000	16	43 (0.15)	0 (0.44)	100 (0.00)	100 (0.00)	100 (0.00)	311.31	84.92	50.30	61.36
100	1000	18	78 (0.10)	0 (0.41)	100 (0.00)	100 (0.00)	100 (0.00)	315.93	84.87	50.21	63.05
100	1000	20	89 (0.09)	0 (0.40)	100 (0.00)	100 (0.00)	100 (0.00)	295.15	81.43	48.60	59.26

Table 3.6: Complete results for the "line" arrangement considered in Simulation 3.6.2. Number of replications identifying the correct number of significant clusters, median computation time over 100 replications.

parameters			$\lvert\hat{K}=3\rvert$					median time (sec.)			
n_k	p	δ	pvAU	pvBP	SHC1	SHC2$_L$	SHC2$_2$	pv	SHC1	SHC2$_L$	SHC2$_2$
50	2	1	–	–	0	0	0	–	2.22	0.99	1.77
50	2	2	–	–	0	0	1	–	2.17	0.98	1.73
50	2	3	–	–	20	0	30	–	2.23	0.99	1.77
50	2	4	–	–	79	3	85	–	2.24	0.99	1.80
50	2	5	–	–	98	9	99	–	2.23	1.00	1.78
100	2	1	–	–	0	0	0	–	4.88	2.41	3.92
100	2	2	–	–	0	0	3	–	4.78	2.37	3.85
100	2	3	–	–	46	8	60	–	4.85	2.40	3.92
100	2	4	–	–	93	44	96	–	4.87	2.42	3.93
100	2	5	–	–	100	72	100	–	4.90	2.42	3.95
50	1000	2	0	0	0	0	0	200.13	72.28	41.60	49.02
50	1000	4	0	0	0	0	0	199.97	62.44	38.41	45.68
50	1000	6	0	0	0	0	0	200.07	63.92	40.08	47.30
50	1000	8	0	0	0	2	69	199.51	63.79	39.84	47.00
50	1000	10	0	0	5	91	97	199.88	66.00	40.53	48.15
50	1000	12	0	0	19	93	100	199.34	65.57	41.28	48.58
50	1000	14	0	0	20	87	100	199.98	65.49	41.16	48.18
50	1000	16	3	0	48	84	98	199.62	66.19	40.76	48.14
50	1000	18	9	0	56	81	100	199.78	65.33	40.78	48.09
50	1000	20	8	0	73	71	100	200.12	65.53	41.47	48.91
100	1000	2	0	0	0	0	0	762.83	155.28	101.07	120.10
100	1000	4	0	0	0	0	0	763.74	154.52	97.56	116.54
100	1000	6	0	0	0	2	26	768.71	153.66	98.31	117.24
100	1000	8	0	0	1	99	98	768.34	160.74	101.88	119.63
100	1000	10	0	0	71	100	100	775.47	176.17	101.00	131.41
100	1000	12	0	0	98	100	100	881.97	153.13	99.08	118.12
100	1000	14	0	0	99	100	100	882.03	175.52	113.72	121.56
100	1000	16	1	0	100	100	100	883.02	152.94	97.92	135.61
100	1000	18	12	0	100	100	100	883.94	180.11	100.53	120.02
100	1000	20	6	0	100	100	100	881.96	153.88	113.75	118.49

Table 3.7: Complete results for the "triangle" arrangement considered in Simulation 3.6.2. Number of replications identifying the correct number of significant clusters, median computation time over 100 replications.

parameters			$\lvert\hat{K}=3\rvert$					median time (sec.)			
n_k	p	δ	pvAU	pvBP	SHC1	SHC2$_L$	SHC2$_2$	pv	SHC1	SHC2$_L$	SHC2$_2$
50	2	1	–	–	0	0	0	–	2.22	1.00	1.77
50	2	2	–	–	0	0	0	–	2.13	0.98	1.76
50	2	3	–	–	0	0	8	–	2.22	1.00	1.78
50	2	4	–	–	28	29	81	–	2.18	0.98	1.76
50	2	5	–	–	98	94	99	–	2.22	1.00	1.76
100	2	1	–	–	0	0	0	–	4.86	2.44	3.94
100	2	2	–	–	0	0	0	–	4.84	2.40	3.94
100	2	3	–	–	2	11	32	–	4.81	2.40	3.89
100	2	4	–	–	72	89	100	–	4.93	2.44	3.93
100	2	5	–	–	100	100	100	–	4.86	2.43	3.92
50	1000	2	0	0	0	0	0	232.70	75.08	46.81	56.06
50	1000	4	0	0	0	0	0	232.78	75.45	47.00	56.05
50	1000	6	0	0	0	0	0	232.53	76.34	47.24	56.53
50	1000	8	0	0	0	1	78	232.41	75.60	46.97	56.26
50	1000	10	0	0	89	100	100	232.28	76.72	47.90	57.14
50	1000	12	0	0	100	100	100	232.50	76.51	47.84	57.37
50	1000	14	0	0	100	100	100	232.32	75.86	47.16	56.51
50	1000	16	12	0	100	100	100	232.47	75.57	47.38	56.62
50	1000	18	48	0	100	100	100	232.28	76.29	47.43	57.08
50	1000	20	33	0	100	100	100	232.46	75.72	47.29	56.83
100	1000	2	0	0	0	0	0	885.44	176.23	113.44	137.73
100	1000	4	0	0	0	0	0	885.49	179.78	115.81	140.23
100	1000	6	0	0	0	0	5	885.93	181.16	116.15	140.19
100	1000	8	0	0	2	100	100	885.79	182.45	117.99	142.00
100	1000	10	0	0	100	100	100	885.94	183.59	118.18	142.90
100	1000	12	0	0	100	100	100	885.36	181.59	116.98	141.46
100	1000	14	0	0	100	100	100	886.11	184.46	117.54	141.82
100	1000	16	4	0	100	100	100	886.11	182.38	117.42	142.67
100	1000	18	39	0	100	100	100	886.50	181.53	116.95	140.76
100	1000	20	47	0	100	100	100	886.73	179.93	116.39	140.08

Table 3.8: Complete results for the "square" arrangement considered in Simulation 3.6.3. Number of replications identifying the correct number of significant clusters, median computation time over 100 replications.

parameters			$\lvert\hat{K}=4\rvert$					median time (sec.)			
n_k	p	δ	pvAU	pvBP	SHC1	SHC2$_L$	SHC2$_2$	pv	SHC1	SHC2$_L$	SHC2$_2$
50	2	1	–	–	0	0	0	–	2.47	1.09	2.08
50	2	2	–	–	0	0	0	–	2.29	1.04	1.95
50	2	3	–	–	3	0	17	–	2.54	1.16	2.06
50	2	4	–	–	78	12	90	–	2.57	1.16	2.07
50	2	5	–	–	100	84	100	–	2.70	1.22	2.16
100	2	1	–	–	0	0	0	–	5.31	2.69	4.61
100	2	2	–	–	0	0	0	–	5.14	2.60	4.32
100	2	3	–	–	18	4	53	–	5.33	2.54	4.42
100	2	4	–	–	98	85	99	–	4.87	2.43	4.11
100	2	5	–	–	100	100	100	–	5.04	2.37	4.39
50	1000	2	0	0	0	0	0	305.45	81.27	49.64	59.28
50	1000	4	0	0	0	0	0	299.88	80.44	48.94	58.58
50	1000	6	0	0	0	0	0	299.85	80.59	48.69	57.49
50	1000	8	0	0	0	1	67	300.88	81.43	49.38	58.52
50	1000	10	0	0	95	100	100	301.08	82.16	50.41	59.38
50	1000	12	0	0	100	100	100	300.91	82.17	49.69	59.50
50	1000	14	1	0	100	100	100	298.67	81.54	49.50	58.98
50	1000	16	77	0	100	100	100	405.38	109.32	68.26	81.71
50	1000	18	97	0	100	100	100	402.78	109.64	68.78	82.11
50	1000	20	99	0	100	100	100	403.52	110.56	68.81	82.55

Table 3.9: Complete results for the "tetrahedron" arrangement considered in Simulation 3.6.3. Number of replications identifying the correct number of significant clusters, median computation time over 100 replications.

parameters			$\lvert\hat{K}=4\rvert$					median time (sec.)			
n_k	p	δ	pvAU	pvBP	SHC1	SHC2$_L$	SHC2$_2$	pv	SHC1	SHC2$_L$	SHC2$_2$
50	3	1	–	–	0	0	0	–	2.54	1.10	1.98
50	3	2	–	–	0	0	0	–	2.57	1.12	2.02
50	3	3	–	–	0	0	0	–	2.43	1.10	1.91
50	3	4	–	–	0	9	33	–	2.84	1.28	2.25
50	3	5	–	–	24	86	99	–	2.40	1.09	2.03
100	3	1	–	–	0	0	0	–	5.24	2.61	4.44
100	3	2	–	–	0	0	0	–	5.79	2.74	4.77
100	3	3	–	–	0	1	2	–	5.80	2.95	4.55
100	3	4	–	–	2	84	94	–	5.99	2.92	4.76
100	3	5	–	–	88	99	100	–	5.11	2.63	4.19
50	1000	2	0	0	0	0	0	370.14	94.85	58.06	69.12
50	1000	4	0	0	0	0	0	363.98	92.81	57.70	69.67
50	1000	6	0	0	0	0	0	367.20	95.37	57.39	75.23
50	1000	8	0	0	0	0	37	385.62	93.53	58.15	71.01
50	1000	10	0	0	56	98	100	364.72	98.81	60.74	72.49
50	1000	12	0	0	100	100	100	383.10	98.10	61.74	78.22
50	1000	14	0	0	100	100	100	368.79	98.68	60.75	77.42
50	1000	16	16	0	100	100	100	403.31	114.20	71.40	85.96
50	1000	18	53	0	100	100	100	402.61	112.10	70.47	84.58
50	1000	20	68	0	100	100	100	404.02	113.07	70.92	85.09

Table 3.10: Complete results for the "rectangle" arrangement considered in Simulation 3.6.3. Number of replications identifying the correct number of significant clusters, median computation time over 100 replications.

parameters			$\lvert\hat{K}=4\rvert$					median time (sec.)			
n_k	p	δ	pvAU	pvBP	SHC1	SHC2$_L$	SHC2$_2$	pv	SHC1	SHC2$_L$	SHC2$_2$
50	2	1	–	–	0	0	0	–	2.19	0.99	1.76
50	2	2	–	–	0	0	0	–	2.20	0.97	1.78
50	2	3	–	–	10	0	22	–	2.20	0.99	1.75
50	2	4	–	–	88	26	94	–	2.49	1.11	1.95
50	2	5	–	–	100	96	100	–	2.49	1.07	1.97
100	2	1	–	–	0	0	0	–	5.12	2.66	4.30
100	2	2	–	–	0	0	0	–	5.03	2.58	4.12
100	2	3	–	–	43	7	54	–	4.77	2.36	3.98
100	2	4	–	–	98	89	99	–	4.78	2.38	3.98
100	2	5	–	–	100	100	100	–	4.80	2.37	3.93
50	1000	2	0	0	0	0	0	402.83	108.12	67.98	81.53
50	1000	4	0	0	0	0	0	402.46	107.61	67.66	81.23
50	1000	6	0	0	0	0	0	402.53	108.08	67.88	81.06
50	1000	8	0	0	0	1	78	401.98	109.30	68.57	81.89
50	1000	10	0	0	98	99	100	401.32	107.70	67.78	81.17
50	1000	12	0	0	100	100	100	401.78	108.67	68.35	81.28
50	1000	14	22	0	100	100	100	401.97	108.83	68.82	81.95
50	1000	16	58	0	100	100	100	401.86	108.93	68.57	81.90
50	1000	18	66	0	100	100	100	401.78	107.34	67.93	80.97
50	1000	20	64	0	100	100	100	401.13	108.17	68.24	81.58

Table 3.11: Complete results for the "stretched tetrahedron" arrangement considered in Simulation 3.6.3. Number of replications identifying the correct number of significant clusters, median computation time over 100 replications.

parameters			$\lvert\hat{K}=4\rvert$					median time (sec.)			
n_k	p	δ	pvAU	pvBP	SHC1	SHC2$_L$	SHC2$_2$	pv	SHC1	SHC2$_L$	SHC2$_2$
50	3	1	–	–	0	0	0	–	2.30	1.04	1.83
50	3	2	–	–	0	0	0	–	2.41	1.09	1.95
50	3	3	–	–	0	0	5	–	2.38	1.07	1.91
50	3	4	–	–	8	12	72	–	2.33	1.06	1.89
50	3	5	–	–	88	96	100	–	2.45	1.09	1.93
100	3	1	–	–	0	0	0	–	4.96	2.54	4.14
100	3	2	–	–	0	0	0	–	5.04	2.59	4.13
100	3	3	–	–	0	9	29	–	5.13	2.65	4.22
100	3	4	–	–	40	90	98	–	5.23	2.63	4.21
100	3	5	–	–	100	99	100	–	5.06	2.61	4.15
50	1000	2	0	0	0	0	0	300.42	79.91	48.80	57.90
50	1000	4	0	0	0	0	0	298.10	80.43	49.44	58.76
50	1000	6	0	0	0	0	0	297.97	81.65	49.68	59.07
50	1000	8	0	0	0	2	62	301.60	82.63	50.71	60.76
50	1000	10	0	0	87	99	100	300.92	83.61	51.22	61.01
50	1000	12	0	0	100	100	100	298.19	82.12	50.50	59.58
50	1000	14	3	0	100	100	100	401.31	111.78	70.41	84.39
50	1000	16	6	0	100	100	100	401.12	111.20	70.38	84.10
50	1000	18	2	0	100	100	100	401.11	110.89	70.20	84.63
50	1000	20	0	0	100	100	100	403.34	114.01	71.74	86.03

CHAPTER 4

Large-Margin Classification with Multiple Decision Rules

4.1 Introduction

Classification is one of the most widely applied and well studied problems in supervised learning. Given a training set of observed covariates and outcomes, in classification, the outcome is modeled as a function of the set of covariates. However, in contrast to standard regression with a continuous response, classification describes the setting where the outcome is a discrete class label. While generalizations to more than two classes exist, in this chapter we focus on the standard binary problem where the label takes one of two possible values, typically denoted by $+1$ and -1.

Given such a dataset, commonly, the goal is to build a model, either to predict the class of a new observation from the covariate space, or to estimate the probably of each class as a function of the covariates. The tasks correspond respectively to hard and soft classification, as described in Section 1.1. Briefly, we refer to methods which only target the optimal prediction rule as hard classifiers, and those which produce estimates of class probability as soft classifiers. Examples of hard classifiers include the support vector machine (SVM) (Vapnik, 1995, 1998) and ψ-learning (Shen et al., 2003; Liu and Shen, 2006), and examples of soft classifiers include logistic regression and other likelihood-based approaches. Often, soft classifiers are also used to obtain hard classification rules by predicting the class with greater estimated probability. These rules are commonly referred to as plug-in classifiers. While hard classification rules do not directly provide conditional class probability estimates, several approaches have been proposed for estimating class probabilities based on hard classifiers, including those of Platt (1999) and Wang et al. (2008). As such, methods which may be traditionally viewed as soft and hard classifiers are often used for either task. Naturally, a question of interest is: how are hard and soft classifiers related, and how do they differ in practice?

Recently, Liu et al. (2011) introduced the Large-margin Unified Machines (LUM) family of margin-based classifiers, shedding some light on the the relationship between hard and soft classifiers. The LUM family connects several popular margin-based classification methods, including SVM, distance-weighted discrimination (DWD; Marron et al., 2007), and a new hybrid logistic loss. Their approach was further extended to the multi-category case by Zhang and Liu (2013). Margin-based approaches to classification are popular in practice for their accuracy and computational efficiency in both low and high-dimensional settings. While a flexible family of margin-based classifiers, the LUM approach examines only a specific parameterized collection of classifiers along the gradient of soft to hard classification. In this chapter, we similarly focus on connecting hard and soft margin-based methods. However, we consider a more natural approach based on connecting the tasks of hard and soft classification rather than specific hard and soft classifiers. We propose a novel framework of binary learning problems which may be formulated as partial or full estimation of the conditional class probability based on fitting an arbitrary number of boundaries to the data. As an example, suppose we are interested in separating patients into four disease risk groups based on clinical measurements. One possible approach is to group patients according to whether their conditional probability of being positive for the disease is less than 25%, between 25% to 50%, between 50% to 75%, or greater than 75%. In this setting, the emphasis is not on the accuracy of class probability estimates, but instead, on the correct stratification of individuals into risk groups. Therefore, only partial estimation of the conditional class probability is required; in particular, at the three boundaries, 25%, 50%, and 75%. While stratification of the patient classes is possible using a soft classifier, an approach directly targeting the three boundaries may provide improved stratification by requiring weaker assumptions on the entire form of the underlying conditional class probability.

In addition to hard and soft classification, the proposed framework also encompasses rejection-option classification (Herbei and Wegkamp, 2006; Bartlett and Wegkamp, 2008; Yuan and Wegkamp, 2010; Wegkamp and Yuan, 2011) and weighted classification (Lin et al., 2002; Qiao and Liu, 2009), two other well-studied binary learning problems. Briefly, the rejection-option problem expands on standard binary classification by introducing a third option to reject, where neither label is predicted. Notably, it can be shown that the decision to reject directly corresponds to a prediction that the probability of belonging to either class does not exceed a specified threshold.

Since the task requires estimation of more than a single classification boundary, but less than the full class conditional probability, it may be viewed as an intermediate problem to hard and soft classification, as in the example given above. Applications of rejection option classification include certain medical settings where predictions should only be made when a level of certainty is obtained. Additionally, weighted classification extends the standard classification problem by accounting for differences or biases in class populations. We define these problems more formally, along with hard and soft classification, in Section 4.2.

The remainder of this chapter is organized as follows. In the first part of Section 4.2 we provide a review of margin-based learning previously introduced in Section 1.1. Then, in the remainder of Section 4.2, we define our family of binary learning problems and introduce a corresponding *theoretical loss*, which generalizes the standard misclassification error to connect class prediction with probability estimation. In Section 4.3 we provide necessary and sufficient conditions for consistency of a surrogate loss function, and propose a class of consistent piecewise linear surrogates akin to the SVM hinge loss for binary classification. In Section 4.4, we present theoretical bounds on the empirical performance of classification rules obtained using surrogate loss functions. In Section 4.5, we provide a sub-gradient descent (SGD) algorithm for solving the corresponding optimization problem using the proposed piecewise linear surrogates. We then illustrate the behavior of our generalized family of classifiers using simulation in Section 4.6, and a real data example from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database in Section 4.7. We conclude in Section 4.8 with a discussion of the proposed framework. All technical proofs are included in Section 4.9.

4.2 Methodology

In this section, we first briefly review margin-based classifiers, and formally define the notion of classification consistency for loss functions. We then state the general form of our unified framework of problems and introduce a corresponding family of theoretical loss functions which encompasses the standard misclassification error as a special case.

4.2.1 Margin-Based Classifiers

Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ denote a training set of n covariate–label pairs drawn from $\mathcal{X} \times \mathcal{Y}$ according to some unknown distribution $\mathcal{P}(\boldsymbol{X}, Y)$. For binary problems, $\mathcal{Y} = \{-1, +1\}$ is used to denote the label space, and often $\mathcal{X} \subset \mathbb{R}^p$, with $p \geq 1$. Given a training set, margin-based classifiers minimize a penalized loss over a class, $\mathcal{F}$, of margin functions, $f : \mathcal{X} \to \mathbb{R}$. Typically, the corresponding optimization problem is written as:

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \underbrace{L\big(y_i f(\boldsymbol{x}_i)\big)}_{loss} + \underbrace{\lambda J(f)}_{penalty}, \tag{4.1}$$

where $L : \mathbb{R} \to \mathbb{R}$ is a loss function defined with respect to the functional margin, $yf(\boldsymbol{x})$, and $J : \mathcal{F} \to \mathbb{R}$ is some roughness measure on $\mathcal{F}$ with corresponding tuning parameter $\lambda \geq 0$. Both hard and soft classification may be formulated as margin-based problems. In the case of hard classification, with a little abuse of notation, we use $\widehat{Y} \in \mathcal{Y}$ to denote a predicted class label, and $\widehat{Y} : \mathbb{R} \to \mathcal{Y}$ to denote a prediction rule on $\mathbb{R}$. In margin-based classification, $\widehat{Y}(\cdot)$ is combined with a margin function, $f \in \mathcal{F}$, to obtain predictions in $\mathcal{Y}$. Most commonly, in hard classification the sign rule, $\widehat{Y}(f(\boldsymbol{X})) = \text{sign}(f(\boldsymbol{x}))$, is used assuming $f(\boldsymbol{x}) \neq 0$ almost surely (*a.s.*). Thus, given a new $(\boldsymbol{x}^*, y^*)$ pair with $f(\boldsymbol{x}^*) \neq 0$, correct classification is obtained if and only if $y^* f(\boldsymbol{x}^*) > 0$. Since the functional margin, $yf(x)$, serves as an approximate measure for classification correctness, the loss function, L, in (4.1) is often chosen to be a non-increasing function over $yf(\boldsymbol{x})$. A natural choice of L in hard classification is the misclassification error, or $0-1$ loss, given by:

$$\ell_{0-1}(Y, \widehat{Y}) = \mathbf{I}\{\widehat{Y} \neq Y\}, \tag{4.2}$$

where $\mathbf{I}\{\cdot\}$ is used to denote the indicator function. Using the sign rule, the loss may be equivalently written over the class of margin functions as: $L_{0-1}(Yf(\boldsymbol{X})) = \mathbf{I}\{Yf(\boldsymbol{X}) < 0\}$. However, direct optimization of the non-convex and discontinuous loss, L_{0-1}, is NP-hard and often infeasible in practice. Thus, continuous convex losses, called surrogates, are commonly used instead. Choices of the surrogate loss function corresponding to existing margin-based classifiers include the SVM hinge loss, $L(z) = \max\{0, 1 - z\}$, logistic loss, $L(z) = \log(1 + e^{-z})$, and the DWD loss, $L(z) = \frac{1}{4z} \cdot \mathbf{I}\{z \geq \frac{1}{2}\} + (1 - z) \cdot \mathbf{I}\{z < \frac{1}{2}\}$ (Figure 1.2). Finally, the penalty term, $J(\cdot)$ is used to prevent over-fitting and improve generalizability of the resulting classifier. The amount of penalization is

commonly determined by cross-validation over a grid of λ values. Here, we note that while in the literature there exists a natural theoretical loss for hard classification, i.e. the $0-1$ loss, there is no equivalent theoretical loss targeting consistent probability estimation for soft classification. In addition to providing a spectrum of theoretical loss functions covering soft and hard classifications at the two extremes, our proposed framework also naturally defines precisely such a theoretical loss for the soft classification problem (Figure 4.2C).

In Section 4.1, we briefly discussed the learning tasks of rejection-option and weighted classification. As with hard and soft classification, these tasks may also be formulated as margin-based problems. We next describe how rejection-option classification may be formulated as a problem of the form (4.1). Borrowing the notation of Yuan and Wegkamp (2010), we use 0 to denote the rejection option such that a prediction, $\widehat{Y}_{rej}$, takes values in $\mathcal{Y}_{rej} = \{+1, 0, -1\}$. Then, for some pre-specified *rejection cost* $\pi \in (0, \frac{1}{2})$, they propose the following theoretical loss for rejection-option classification:

$$\ell_{rej,\pi}(Y, \widehat{Y}_{rej}) = \begin{cases} 1 & \text{if } \widehat{Y}_{rej} \neq Y,\ \widehat{Y}_{rej} \neq 0 \\ \pi & \text{if } \widehat{Y}_{rej} = 0 \\ 0 & \text{otherwise} \end{cases} . \tag{4.3}$$

To express the loss as a function over $Yf(\boldsymbol{X})$, Yuan and Wegkamp (2010) propose the prediction rule $\widehat{Y}_{rej}(f(\boldsymbol{X}); \delta) = \mathbf{I}\{|Yf(\boldsymbol{X})| > \delta\} \cdot \text{sign}(Yf(\boldsymbol{X}))$ for some appropriately chosen $\delta > 0$. Then, $\ell_{rej,\pi}$ may be written as the following generalized $0-1$ loss on $Yf(\boldsymbol{X})$:

$$L_{rej,\pi}(Yf(\boldsymbol{X}); \delta) = (1 - \pi)\mathbf{I}\{Yf(\boldsymbol{X}) \leq -\delta\} + \pi\mathbf{I}\{Yf(\boldsymbol{X}) < \delta\}.$$

We finally consider the task of weighted classification. In contrast to the problems mentioned thus far, to fit the form of (4.1), weighted classification requires specifying separate theoretical loss functions for observations from the $+1$ and -1 classes, denoted by $\ell^{+}_{\text{w},\pi}$ and $\ell^{-}_{\text{w},\pi}$. For simplicity, we use $\ell^{Y}_{\text{w},\pi}$ to denote the loss for both classes. Similar to hard classification, the task is to predict class labels in $\mathcal{Y} = \{+1, -1\}$. The loss function depends on a weight parameter, π, which accounts for imbalances between the two classes. Commonly, π is constrained to the interval $(0, 1)$ without

loss of generality. Then, for fixed weight π, the weighted loss is given by:

$$\begin{aligned}
\ell^Y_{\mathrm{w},\pi}(Y,\widehat{Y}) &= \mathbf{I}\{Y=+1\}\cdot\ell^+_{w,\pi}(\widehat{Y}) + \mathbf{I}\{Y=-1\}\cdot\ell^-_{w,\pi}(\widehat{Y}), \qquad (4.4)\\
\ell^+_{\mathrm{w},\pi}(\widehat{Y}) &= (1-\pi)\cdot\mathbf{I}\{\widehat{Y}\neq+1\},\\
\ell^-_{\mathrm{w},\pi}(\widehat{Y}) &= \pi\cdot\mathbf{I}\{\widehat{Y}\neq-1\}.
\end{aligned}$$

Note that the standard $0-1$ loss corresponds to the special case of the weighted loss (4.4) when equal weight is assigned to the two classes with $\pi=\frac{1}{2}$. Using the same prediction rule as for hard classification, $\widehat{Y}(f(\boldsymbol{x}))=\mathrm{sign}(f(\boldsymbol{x}))$, the loss over the functional margin may be written:

$$\begin{aligned}
L^Y_{\mathrm{w},\pi}(Yf(\boldsymbol{X})) &= \mathbf{I}\{Y=+1\}\cdot L^+_{\mathrm{w},\pi}(Yf(\boldsymbol{X})) + \mathbf{I}\{Y=-1\}\cdot L^-_{\mathrm{w},\pi}(Yf(\boldsymbol{X})),\\
L^+_{\mathrm{w},\pi}(Yf(\boldsymbol{X})) &= (1-\pi)\cdot\mathbf{I}\{Yf(\boldsymbol{X})<0\},\\
L^-_{\mathrm{w},\pi}(Yf(\boldsymbol{X})) &= \pi\cdot\mathbf{I}\{Yf(\boldsymbol{X})<0\}.
\end{aligned}$$

As with the usual $0-1$ loss, optimization of $L_{rej,\pi}$ and $L^Y_{\mathrm{w},\pi}$ is NP-hard, and in practice should be approximated using a convex surrogate loss. In the next section, we introduce the notion of consistency, an important statistical property of surrogate loss functions.

4.2.2 Classification Consistency

Much work has been done to study the statistical properties of classifiers of the *loss + penalty* form given in (4.1) (Steinwart and Scovel, 2007; Blanchard et al., 2008; Bartlett et al., 2006; Cristianini and Shawe-Taylor, 2000). Of these, consistency of loss functions is one of the most fundamental. In general, a loss function is called consistent for a margin-based learning problem if it recovers in expectation the optimal rule, often called the Bayes rule, to the theoretical loss function, e.g. ℓ_{0-1}, $\ell_{rej,\pi}$ or $\ell^Y_{\mathrm{w},\pi}$. More formally, for a theoretical loss function, ℓ, and a surrogate loss, ϕ, let $Y^*_\ell(\boldsymbol{X})=\mathrm{argmin}_{Y^*}\,\mathbb{E}_{Y|\boldsymbol{X}}\{\ell(Y,Y^*)\}$ and $f^*_\phi(\boldsymbol{X})=\mathrm{argmin}_f\,\mathbb{E}_{Y|\boldsymbol{X}}\{\phi(Yf(\boldsymbol{X}))\}$ denote the Bayes rule and ϕ-optimal margin function, respectively. Then, we call ϕ consistent for ℓ if $\widehat{Y}_\ell(f^*_\phi(\boldsymbol{X}))=Y^*_\ell(\boldsymbol{X})$, where $\widehat{Y}_\ell$ is the appropriate prediction rule, e.g. the sign function. Equivalently, using the margin-based formulation of the theoretical loss, L, and letting $f^*_L(\boldsymbol{X})=\mathrm{argmin}_f\,\mathbb{E}_{Y|\boldsymbol{X}}\{L(Yf(\boldsymbol{X}))\}$ denote the L-optimal margin function, consistency may be expressed as $\widehat{Y}_\ell(f^*_\phi(\boldsymbol{X}))=\widehat{Y}_\ell(f^*_L(\boldsymbol{X}))$. For

rejection-option classification, the Bayes optimal rule is given by:

$$Y^*_{rej,\pi}(\boldsymbol{X}) = \begin{cases} +1 & \text{if } p(\boldsymbol{X}) \geq 1-\pi \\ 0 & \text{if } p(\boldsymbol{X}) \in (\pi, 1-\pi) \\ -1 & \text{if } p(\boldsymbol{X}) \leq \pi \end{cases}. \tag{4.5}$$

The Bayes optimal rule for weighted classification is given by:

$$Y^*_{\mathrm{w},\pi}(\boldsymbol{X}) = \begin{cases} +1 & \text{if } p(\boldsymbol{X}) > \pi \\ -1 & \text{if } p(\boldsymbol{X}) \leq \pi \end{cases}. \tag{4.6}$$

For hard classification, the Bayes optimal rule corresponds to $Y_{\mathrm{w},0.5}$, and consistency is often referred to as Fisher consistency or classification calibrated (Bartlett et al., 2006). While no theoretical loss has been proposed for soft classification, using $p(\boldsymbol{X}) = \mathbb{P}(Y = +1|\boldsymbol{X})$ to denote the conditional class probability at $\boldsymbol{X} \in \mathcal{X}$, commonly, ϕ is called consistent for soft classification if there exists some monotone mapping, $C : \mathbb{R} \to [0, 1]$ such that $C(f^*_\phi(\boldsymbol{X})) = p(\boldsymbol{X})$. Naturally, $C(\cdot)$ may be viewed as an extension of the prediction rules $\widehat{Y}(\cdot)$ and $\widehat{Y}_{rej}(\cdot; \delta)$ given for hard and rejection-option classification. Necessary and sufficient conditions for Fisher, rejection-option, and probability estimation consistency have been described in Lin (2002); Yuan and Wegkamp (2010); Zhang et al. (2013).

In this chapter, we propose a novel framework for unifying hard, soft, rejection-option, and weighted classification through a generalized formulation of their corresponding theoretical losses, corresponding Bayes optimal rules, and necessary and sufficient conditions for consistency. Our generalized formulation not only provides a platform for comparing existing binary classification tasks, but also introduces an entire family of new tasks which fills the gap between these problems. We next formally introduce our unified framework of binary learning problems.

4.2.3 Unified Framework

First, we note that all of the classification tasks described in Section 4.2.1 may be formulated as learning problems which target partial or complete estimation of the conditional class probability, $p(\boldsymbol{x})$. We propose our framework of unified margin-based learning problems based on this insight. Let $\Omega_{\boldsymbol{\pi}}$ denote the ordered $(K + 1)$ partition of the interval $[0, 1]$ obtained by splitting at $\boldsymbol{\pi} =$

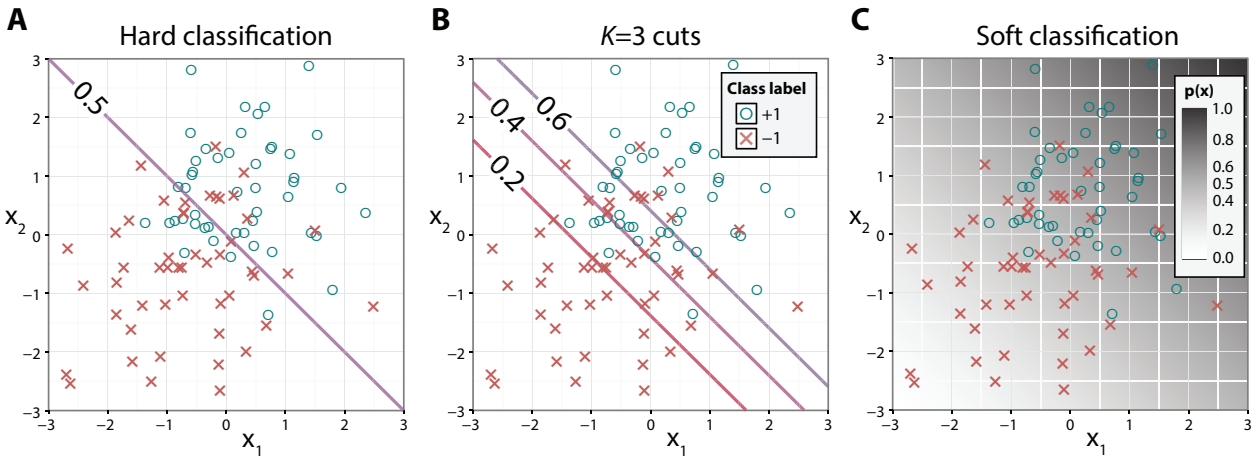

Figure 4.1: Boundaries are shown separating the input space, $\mathbb{R}^2$ into the $K+1$ prediction sets, $\Omega_{\boldsymbol{\pi}}$. A sample of 100 observations drawn from the underlying class populations are overlaid to show the distribution over the space. (A) The $K=1$ boundary for $\boldsymbol{\pi}=\{0.5\}$ corresponding to hard classification is shown by the separating hyperplane corresponding to the set $\{\boldsymbol{x}\in\mathbb{R}^2 : p(\boldsymbol{x})=0.5\}$. (B) The set of $K=3$ boundaries are shown for $\boldsymbol{\pi}=\{0.2,0.4,0.6\}$ separating the 4 prediction sets. (C) The soft classification results are shown spanning the entire set of $\pi_k\in(0,1)$. As $K\to\infty$, moving from hard to soft classification, the set of learning problems becomes increasingly complex.

$\{\pi_1,\ldots,\pi_K\}$, where $0<\pi_1<\ldots<\pi_K<1$. Assume $p(\boldsymbol{x})\neq\pi_k$ *a.s.* for all k, such that observations belong to only a single region of interval. Letting $\pi_0=0$ and $\pi_{K+1}=1$ for ease of notation, we write:

$$\Omega_{\boldsymbol{\pi}}=\{\omega_0,\ldots,\omega_K\},$$

where $\omega_0=[\pi_0,\pi_1]$, and $\omega_k=(\pi_k,\pi_{k+1}]$, for $k\geq 1$. As our framework, we propose the class of problems which target a partition of the covariate space, $\mathcal{X}$, into the $K+1$ regions, $\{\boldsymbol{x} : p(\boldsymbol{x})\in\omega_k\}$. In Figure 4.1, we show a sample of 100 observations drawn from the same underlying distribution, $\mathcal{P}(\boldsymbol{X},Y)$, along with optimal solutions to three representative problems from our proposed framework. Note that the extreme cases of $K=1$ with $\boldsymbol{\pi}=\{0.5\}$ (Figure 4.1A), and $K=\infty$ with $\boldsymbol{\pi}$ dense on $(0,1)$ (Figure 4.1C) correspond to hard and soft classification, respectively. We discuss these connections in more detail later in this section. To illustrate the spectrum of problems in our framework, we also show a new intermediate problem in Figure 4.1B, with $K=3$ and $\boldsymbol{\pi}=\{0.2,0.4,0.6\}$.

Formally, we define our framework as the collection of minimization tasks of a theoretical loss which generalizes the $0-1$ loss, over the collection of rules $\mathcal{G}_{\boldsymbol{\pi}}=\{g : \mathcal{X}\to\Omega_{\boldsymbol{\pi}}\}$. Recall the

weighted $0-1$ loss, ℓ_{w}, for weighted classification described above. For positive and negative class weights $(1-\pi)$ and π where $\pi \in (0,1)$, the weighted $0-1$ loss has corresponding Bayes boundary at $\{\boldsymbol{x} : p(\boldsymbol{x}) = \pi\}$. Problems under our framework may be viewed as the task of simultaneously estimating K such boundaries. Intuitively, we formulate our theoretical loss as the average of K weighted $0-1$ loss functions with corresponding weights $\boldsymbol{\pi}$. Throughout, we use $\ell^{+}_{\boldsymbol{\pi}}(g(\boldsymbol{x}))$ and $\ell^{-}_{\boldsymbol{\pi}}(g(\boldsymbol{x}))$ to denote the loss for positive and negative class observations, respectively. As with the weighted loss, we use $\ell^{Y}_{\boldsymbol{\pi}}$ to denote the loss for both classes:

$$\ell^{Y}_{\boldsymbol{\pi}}(g(\boldsymbol{X})) = \frac{2}{K}\sum_{k=1}^{K} \ell^{Y}_{\pi_k}(g(\boldsymbol{X})), \tag{4.7}$$

where

$$\ell^{+}_{\pi_k}(g(\boldsymbol{X})) = (1-\pi_k)\cdot \mathbf{I}\{g(\boldsymbol{X}) \leq \pi_k\},$$
$$\ell^{-}_{\pi_k}(g(\boldsymbol{X})) = \pi_k \cdot \mathbf{I}\{g(\boldsymbol{X}) > \pi_k\},$$

and the notion of inequalities is extended to elements of $\Omega_{\boldsymbol{\pi}}$ such that $(\pi_j, \pi_{j+1}] \leq \pi_k$ if $\pi_{j+1} \leq \pi_k$ and $(\pi_j, \pi_{j+1}] > \pi_k$ if $\pi_j \geq \pi_k$. Our theoretical loss encompasses the usual $0-1$ loss, its weighted variant, and the rejection-option loss proposed by Yuan and Wegkamp (2010). The multiplicative constant, 2, is included in (4.7) such that $\ell^{Y}_{\boldsymbol{\pi}}$ reduces precisely to the usual $0-1$ loss when $\boldsymbol{\pi} = \{0.5\}$. Note that since $\ell^{Y}_{\boldsymbol{\pi}}$ is effectively the average of K indicator functions scaled by 2, the function takes values in the interval $[0,2]$. In Figure 4.2, we show $\ell^{Y}_{\boldsymbol{\pi}}$ as a function of $g(\boldsymbol{x}) \mapsto \Omega_{\boldsymbol{\pi}}$, corresponding to the problems in Figure 4.1. Along the horizontal axis, the range $[0,1]$ is split into corresponding $\omega_j = (\pi_j, \pi_{j+1}]$ intervals. Note that the loss function is constant within each interval, giving the appearance of a step function, except in the extreme case when $K = \infty$. As K increases, the theoretical loss becomes smoother, with the limit at $\boldsymbol{\pi} = (0,1)$ corresponding to the proposed theoretical loss for consistent soft classification mentioned in Subsection 4.2.1. Additionally, note that while the loss functions, $\ell^{+}_{\boldsymbol{\pi}}$ and $\ell^{-}_{\boldsymbol{\pi}}$, are symmetric in Panels A and C of Figure 4.2, the same is not true for the loss functions in Panel B. This is due to the fact that the boundaries of interest, $\boldsymbol{\pi}$, are symmetric between the two classes, i.e. $\boldsymbol{\pi} = 1 - \boldsymbol{\pi}$, when $\boldsymbol{\pi} = \{0.5\}$ and $\boldsymbol{\pi} = (0,1)$, but not when $\boldsymbol{\pi} = \{0.2, 0.4, 0.6\}$.

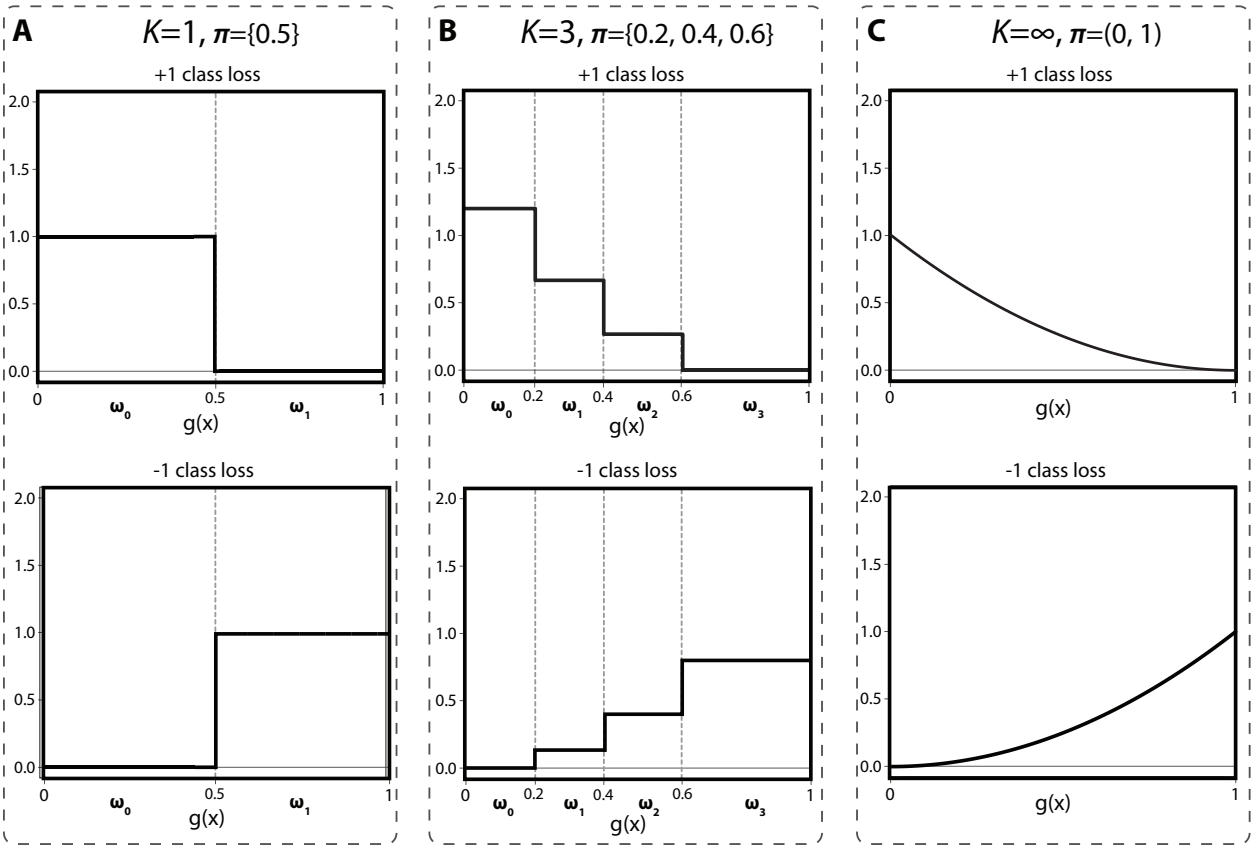

Figure 4.2: Examples of the theoretical loss functions, $\ell_{\boldsymbol{\pi}}^{Y}$, for observations from the positive and negative classes over $g(\boldsymbol{x})$ are shown for boundaries, $\boldsymbol{\pi}$ at (A) $\{0.5\}$, (B) $\{0.2, 0.4, 0.6\}$, and (C) $(0, 1)$, corresponding to the problems shown in Figure 4.1. The theoretical loss generalizes the standard $0-1$ loss given in (A) by incorporating K steps. As K increases and the problem approaches soft classification, the theoretical loss becomes noticeably smoother.

The following result states that the class of problems defined by our theoretical loss indeed corresponds to the proposed framework of learning tasks. That is, the Bayes optimal rule given by $W_{\boldsymbol{\pi}}^{*}(\boldsymbol{X}) = \operatorname{argmin}_g \mathbb{E}_{Y|\boldsymbol{X}}\{\ell_{\boldsymbol{\pi}}^{Y}(g(\boldsymbol{X}))\}$, is precisely the partitioning task described above.

Theorem 4.1. *For fixed K and $\boldsymbol{\pi}$ defined as above, the Bayes optimal rule for the theoretical loss (4.7) is given by:*

$$\begin{aligned} W_{\boldsymbol{\pi}}^{*}(\boldsymbol{X}) &= \operatorname*{argmin}_{g \in \mathcal{G}_{\boldsymbol{\pi}}} \mathbb{E}_{Y|\boldsymbol{X}}\{\ell_{\boldsymbol{\pi}}^{Y}(g(\boldsymbol{X}))\} \\ &= \sum_{k=0}^{K} \omega_k \cdot \boldsymbol{I}\{p(\boldsymbol{X}) \in \omega_k\}. \end{aligned}$$

In addition to the results of Theorem 4.1, the theoretical loss functions for hard (4.2), rejection-option (4.3), and weighted (4.4) classification can be derived as special cases of (4.7). This is shown

by first noting the equivalence of $\Omega_{\boldsymbol{\pi}}$ to $\mathcal{Y}$ and $\mathcal{Y}_{rej}$ based on the Bayes optimal rules, (4.5) and (4.6). From this equivalence, (4.3) and (4.4) can be obtained directly from (4.7). For soft classification, we derive a new theoretical loss from the limiting form of (4.7):

$$\begin{aligned}\ell^Y_{\boldsymbol{\pi}}(g(\boldsymbol{X})) &= \lim_{K\to\infty} \frac{2}{K} \sum_{k=1}^{K} \ell^Y_{\pi_k}(g(\boldsymbol{X})), \\ &= \big(\mathbf{I}\{Y = +1\} - g(\boldsymbol{X})\big)^2.\end{aligned}$$

The resulting theoretical loss is shown in Figure 4.2C. Since $\Omega_{\boldsymbol{\pi}} = (0, 1)$, the Bayes rule is simply the conditional class probability, $g(\boldsymbol{X}) = p(\boldsymbol{X})$, corresponding to soft classification.

As with the problems described in Section 4.2.1, optimization of $\ell_{\boldsymbol{\pi}}$ with respect to $g \in \mathcal{G}_{\boldsymbol{\pi}}$ is NP-hard. Thus, we first reformulate $\ell_{\boldsymbol{\pi}}$ as a function on $\mathbb{R}$ to express the optimization over a collection of margin functions, $\mathcal{F}$. We then propose in Section 4.3 to solve the approximate problem using convex surrogate loss functions. Generalizing the approach of Yuan and Wegkamp (2010) for rejection-option classification, we frame the optimization task over the class of margin functions, $\mathcal{F}$, using a prediction rule $C : \mathbb{R} \times \mathbb{R}^K \to \Omega_{\boldsymbol{\pi}}$ of the form:

$$C(f(\boldsymbol{x}); \boldsymbol{\delta}) = \sum_{k=0}^{K} \omega_k \cdot \mathbf{I}\{f(\boldsymbol{x}) \in (\delta_{k-1}, \delta_k]\}, \tag{4.8}$$

for monotone increasing $\boldsymbol{\delta} = \{\delta_1, \ldots, \delta_K\}$, and $\delta_0 = -\infty$, $\delta_{K+1} = \infty$. Intuitively, each δ_k corresponds to the π_k-boundary along the range of the margin function, $f(\boldsymbol{X})$. As is common in margin-based learning, we write the theoretical loss as the following function over $Yf(\boldsymbol{X})$:

$$\begin{aligned}L^Y_{\boldsymbol{\pi}}(Yf(\boldsymbol{X}); \boldsymbol{\delta}) &= \ell^Y_{\boldsymbol{\pi}}(C(f(\boldsymbol{X}); \boldsymbol{\delta})) \\ &= \begin{cases} \frac{2}{K} \sum_{k=1}^{K} (1 - \pi_k) \cdot \mathbf{I}\{Yf(\boldsymbol{X}) \le \delta_k\} & \text{if } Y = +1 \\ \frac{2}{K} \sum_{k=1}^{K} \pi_k \cdot \mathbf{I}\{Yf(\boldsymbol{X}) < -\delta_k\} & \text{if } Y = -1 \end{cases}.\end{aligned} \tag{4.9}$$

In Figure 4.3, we plot the corresponding margin-based formulations of the theoretical loss functions shown in Figure 4.2, with well chosen $\boldsymbol{\delta}$. Intuitively, both $L^+_{\boldsymbol{\pi}}(\cdot\,; \boldsymbol{\delta})$ and $L^-_{\boldsymbol{\pi}}(\cdot\,; \boldsymbol{\delta})$ are non-increasing on $yf(\boldsymbol{x})$. We also note that $\ell^-_{\boldsymbol{\pi}}$ and $L^-_{\boldsymbol{\pi}}(\cdot\,; \boldsymbol{\delta})$ differ by a reflection along the vertical axis since $L^-_{\boldsymbol{\pi}}(\cdot\,; \boldsymbol{\delta})$ is defined with respect to $yf(\boldsymbol{x}) = -f(\boldsymbol{x})$. Given the margin-based formulation (4.9), we propose to solve our class of problems using convex surrogate loss functions. In the following

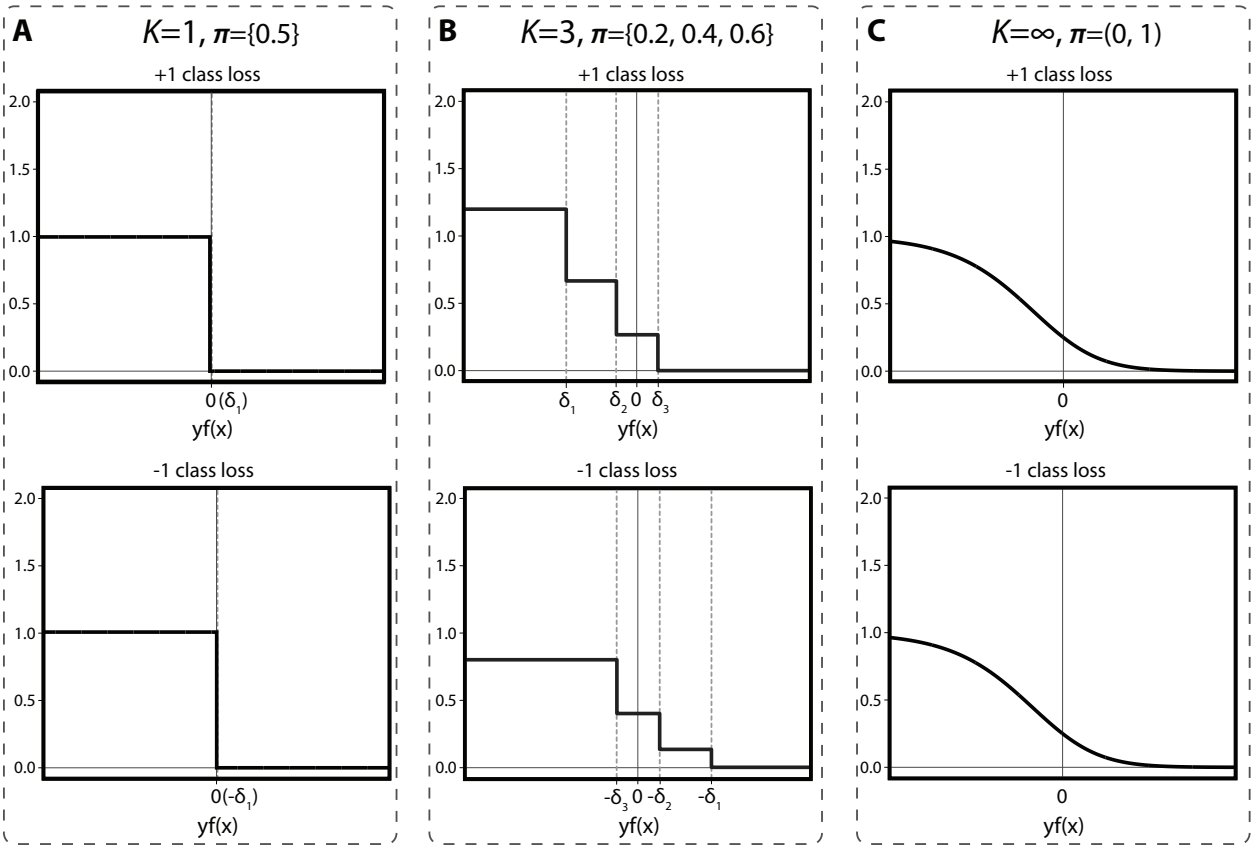

Figure 4.3: Examples of the margin-based formulation of the theoretical loss function, $L^Y_{\boldsymbol{\pi}}(\cdot\ , \boldsymbol{\delta})$, for observations from the positive and negative classes over $yf(\boldsymbol{x})$ are shown for boundaries, $\boldsymbol{\pi}$, at (A) $\{0.5\}$, (B) $\{0.2, 0.4, 0.6\}$, and (C) $(0, 1)$, using well-chosen $\boldsymbol{\delta}$.

section, we first present necessary and sufficient conditions for a surrogate loss to be consistent to (4.7). We then introduce a class of consistent piecewise linear surrogates, which includes the SVM hinge loss as a special case.

4.3 Convex Surrogate Loss Functions

Since the proposed theoretical loss function (4.7) and its margin-based reformulation (4.9) are discontinuous and non-convex for any finite choice of K and $\boldsymbol{\pi}$, empirical minimization can quickly become intractable. Therefore, we propose to instead minimize a convex surrogate loss over a class of margin functions, as in hard and soft classification. In this section, we first provide necessary and sufficient conditions for a surrogate loss to be consistent for (4.7) with fixed K and $\boldsymbol{\pi}$. Then, we introduce a class of convex piecewise linear surrogates which includes the SVM hinge loss as a special case. Intuitively, the piecewise linear surrogates each consist of K non-zero segments,

corresponding to the K boundaries, $\boldsymbol{\pi}$. In the limit, as $\boldsymbol{\pi}$ becomes dense on $(0,1)$, the piecewise linear surrogate tends towards a smooth loss, as in Panel C of Figures 4.2 and 4.3.

4.3.1 Consistency

Throughout this section, we assume K and $\boldsymbol{\pi}$ to be fixed. First, let ϕ^+ and ϕ^- denote a pair of convex surrogate loss functions for $\ell^+_{\boldsymbol{\pi}}$ and $\ell^-_{\boldsymbol{\pi}}$. Further, let $f^*_\phi = \operatorname{argmin}_f \mathbb{E}_{Y|\boldsymbol{X}}\{\phi^Y(Yf(\boldsymbol{X}))\}$ denote the ϕ^Y-optimal rule over the class of all measurable functions. We call ϕ^Y consistent if there exists $\boldsymbol{\delta} \in \mathbb{R}^K$ such that the prediction rule (4.8) satisfies $C(f^*_\phi(\boldsymbol{x}); \boldsymbol{\delta}) = W^*_{\boldsymbol{\pi}}(\boldsymbol{x})$, i.e. if there exists a known monotone mapping from the ϕ^Y-optimal rule to the $K+1$ partition of $\mathcal{X}$ to $\Omega_{\boldsymbol{\pi}}$. The following result provides necessary and sufficient conditions for the consistency of the surrogate loss ϕ^Y to $\ell^Y_{\boldsymbol{\pi}}$.

Theorem 4.2. *A pair of convex surrogate loss functions, ϕ^Y, are consistent for $\ell^Y_{\boldsymbol{\pi}}$ if and only if there exists $\boldsymbol{\delta} \in \mathbb{R}^K$ such that for each $k = 1, \ldots, K$: $\phi^{+\prime}(\delta_k)$ and $\phi^{-\prime}(-\delta_k)$ exist, $\phi^{+\prime}(\delta_k)$ and $\phi^{-\prime}(-\delta_k) < 0$, and*

$$\frac{\phi^{-\prime}(-\delta_k)}{\phi^{-\prime}(-\delta_k) + \phi^{+\prime}(\delta_k)} = \pi_k. \tag{4.10}$$

Naturally, any surrogate loss satisfying the conditions of Theorem 4.2 for some $\boldsymbol{\pi}$, must also satisfy the set of conditions for any subset of the boundaries, $\boldsymbol{\pi}' \subseteq \boldsymbol{\pi}$. Thus, for surrogate loss functions consistent for soft classification, i.e. when $\boldsymbol{\pi} = (0,1)$, there exists an appropriate $\boldsymbol{\delta}$ for any possible K and $\boldsymbol{\pi}$. Similar intuition is used to justify the use of soft classification based plug-in classifiers described in Section 4.1. Examples of surrogate losses consistent for soft classification include the logistic, squared hinge, exponential, and DWD losses. Values of δ_k such that the conditions of Theorem 4.2 are met for these loss functions are provided in Corollaries 3-8 of Yuan and Wegkamp (2010). In the next section, we introduce a class of piecewise linear surrogates which, similar to the SVM loss for hard classification, satisfy consistency for the $\boldsymbol{\pi}$ of interest, but not for any $\boldsymbol{\pi}' \supset \boldsymbol{\pi}$. We refer to such a piecewise linear surrogate as being minimally consistent for a corresponding set of boundaries, $\boldsymbol{\pi}$. In contrast to soft classification losses which satisfy consistency for all $\boldsymbol{\pi} \subseteq (0,1)$, minimally consistent surrogates are well-tuned for a given $\ell^Y_{\boldsymbol{\pi}}$, and may provide improved stratification of $\mathcal{X}$ to the sets, $\Omega_{\boldsymbol{\pi}}$.

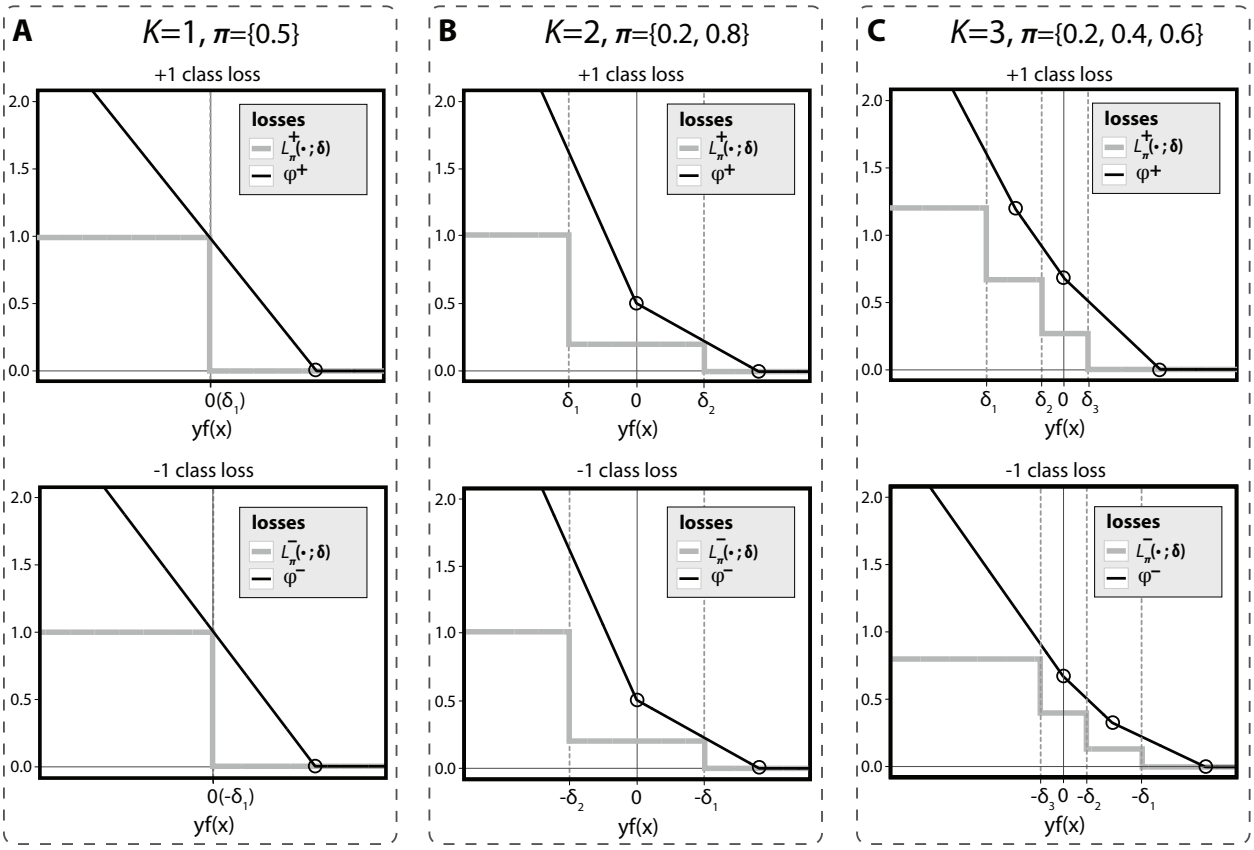

Figure 4.4: Examples of piecewise linear surrogates are shown along with the corresponding theoretical loss, $L_{\boldsymbol{\pi},\boldsymbol{\delta}}$ for (A) $\boldsymbol{\pi} = \{0.5\}$ (hard classification), (B) $\boldsymbol{\pi} = \{0.2, 0.8\}$ (rejection-option classification), and (C) $\boldsymbol{\pi} = \{0.2, 0.4, 0.6\}$.

4.3.2 Piecewise Linear Surrogates

Throughout, we use φ^+ and φ^- to denote piecewise linear surrogates. To build intuition, in the columns of Figure 4.4, we show examples of φ^Y for $K = 1, 2, 3$, corresponding to hard classification, rejection-option classification, and the new problem shown in Figure 4.1B. Circles are used to highlight the hinges, i.e. non-differentiable points, along the piecewise linear loss functions. The corresponding margin-based theoretical loss, $L^Y_{\boldsymbol{\pi}}(\cdot\ ; \boldsymbol{\delta})$, is also shown in each panel using appropriately chosen $\boldsymbol{\delta}$. First, note that the losses in Panels A and B of Figure 4.4 correspond to the standard SVM hinge loss and generalized hinge loss of Bartlett and Wegkamp (2008), respectively. Consider the new surrogate losses in Figure 4.4C for boundaries at $\boldsymbol{\pi} = \{0.2, 0.4, 0.6\}$. Note that φ^+ and φ^- each consist of K non-zero linear segments. Furthermore, each linear segment only spans a single δ_k or $-\delta_k$ for φ^+ and φ^-, respectively. We will refer to these pairs of linear segments as the π_k-consistent segments. This construction allows for the consistency of the surrogate loss

for each $\pi_k \in \boldsymbol{\pi}$ to be controlled separately by the K pairs of π_k-consistent segments along the piecewise linear loss.

We formulate our collection of piecewise linear surrogate losses as the maximum of the K linear segments and 0. Consider first the surrogate loss for positive observations, φ^+. Using $A^+(\pi),\ B^+(\pi)$ to denote the intercept and slope of the π_k-consistent segment, we express the piecewise linear loss as:

$$\varphi^+(z) = \max\{0,\ A^+(\pi_1) + B^+(\pi_1) \cdot z,\ \ldots,\ A^+(\pi_K) + B^+(\pi_K) \cdot z\}. \tag{4.11}$$

We similarly use $A^-(\pi)$ and $B^-(\pi)$ to denote the intercept and slope of the π_k-consistent segment for the negative class loss such that:

$$\varphi^-(z) = \max\{0,\ A^-(\pi_1) + B^-(\pi_1) \cdot z,\ \ldots,\ A^-(\pi_K) + B^-(\pi_K) \cdot z\}. \tag{4.12}$$

By construction, the resulting piecewise linear losses are non-negative, convex and continuous. While (4.11) and (4.12) define a general class of piecewise linear losses, we focus on a subset of minimally consistent piecewise linear surrogates. In the following theorem, we provide a set of sufficient conditions for a piecewise linear loss to be minimally consistent for a specified $\boldsymbol{\pi}$.

Theorem 4.3. *Let $H^Y(\pi, \pi') = (A^Y(\pi) - A^Y(\pi'))/(B^Y(\pi') - B^Y(\pi))$ denote the location of the hinges along the respective loss functions between consecutive boundaries, $\pi < \pi'$. Then, φ^Y is a minimally consistent piecewise linear surrogate for $\boldsymbol{\pi}$ if the intercept and slope parameters, $A^Y(\pi)$ and $B^Y(\pi)$, satisfy the following conditions:*

(C1) $B^+(\pi)$ is non-decreasing, and $B^-(\pi)$ is non-increasing in π.

(C2) The hinge points are such that:

$$\begin{aligned}
-H^-(\pi_{k-1}, \pi_k) &= H^+(\pi_{k-1}, \pi_k) \qquad \text{for } k = 2, \ldots, K, \\
H^+(\pi_{k-1}, \pi_k) &< H^+(\pi_k, \pi_{k+1}) \qquad \text{for } k = 2, \ldots, K-1, \\
A^-(\pi_1)/B^-(\pi_1) &< H^+(\pi_1, \pi_2), \\
A^+(\pi_K)/B^+(\pi_K) &> H^-(\pi_{K-1}, \pi_K).
\end{aligned}$$

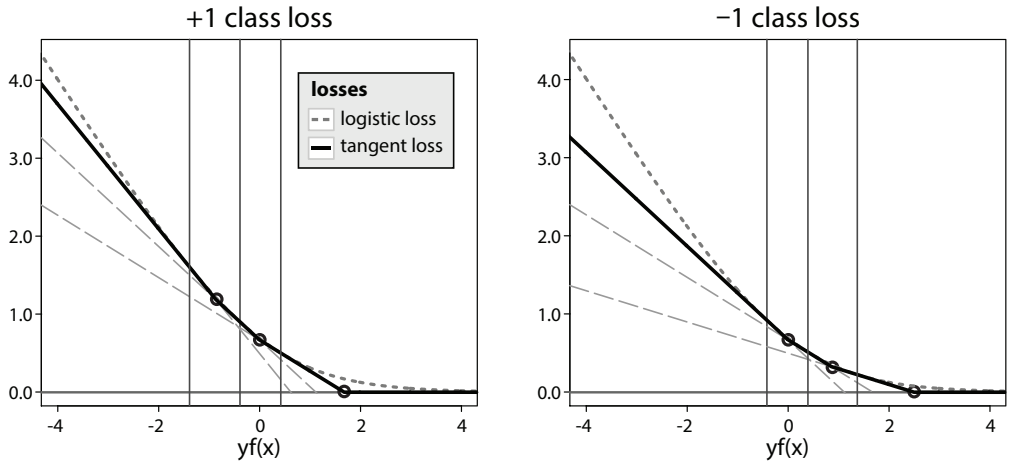

Figure 4.5: A pair of piecewise linear loss functions, φ^Y, obtained from the logistic loss for $\boldsymbol{\pi} = \{0.2, 0.4, 0.6\}$ is shown along with the logistic loss (dotted lines), and the set of tangent lines used to derive $A^Y(\pi)$ and $B^Y(\pi)$ (dashed lines).

(C3) $B^+(\pi), B^-(\pi)$ *satisfy:*

$$\frac{B^-(\pi_k)}{B^-(\pi_k) + B^+(\pi_k)} = \pi_k \quad \textit{for } 1 \leq k \leq K.$$

Conditions (C1) and (C2) guarantee that the linear segments are well-ordered and non-degenerate along $Yf(\boldsymbol{X})$ with appropriately aligned hinge points. Condition (C3) guarantees the consistency of φ^Y to the corresponding $\ell_{\boldsymbol{\pi}}$. Most importantly, by aligning the hinge points, $-H^-(\pi_{k-1}, \pi_k)$ and $H^+(\pi_{k-1}, \pi_k)$, we ensure that there does not exist a $\delta \in \mathbb{R}$ such that (4.10) is satisfied for any $\pi \notin \boldsymbol{\pi}$. Next, we present an approach to obtaining $A^Y(\pi)$ and $B^Y(\pi)$ which satisfy the conditions of Theorem 4.3 using the logistic loss as an example.

4.3.3 Logistic Derived Surrogates

In this subsection, we propose to construct piecewise linear losses by choosing $A^Y(\pi_k) + B^Y(\pi_k) \cdot z$ to be the tangent lines to the logistic loss at $Y \cdot \log(\frac{\pi_k}{1-\pi_k})$. A similar approach was used by Grandvalet et al. (2009) to construct a piecewise linear loss for the rejection-option problem. The following Proposition states that piecewise linear loss functions constructed using this approach satisfy the conditions of Theorem 4.3 for any choice of K and $\boldsymbol{\pi}$.

Proposition 4.1. *For fixed K and $\boldsymbol{\pi}$, let φ^Y be the piecewise linear loss constructed from the tangent lines to the logistic loss such that $A^Y(\pi)$ and $B^Y(\pi)$ are defined as:*

$$A^+(\pi) = A^-(1-\pi) = -\pi\log(\pi) - (1-\pi)\log(1-\pi),$$
$$B^+(\pi) = B^-(1-\pi) = -(1-\pi).$$

Then, φ^Y is a minimally consistent piecewise linear surrogate for $\boldsymbol{\pi}$ satisfying the conditions of Theorem 4.3.

In Figure 4.5, we illustrate the logistic-derived piecewise linear loss for $\boldsymbol{\pi} = \{0.2, 0.4, 0.6\}$. The logistic loss is shown by dotted lines, with the piecewise linear surrogate functions for the positive and negative classes shown in solid black. Thin vertical lines are used to denote the tangent points where the losses are equal, and thin dashed lines give the tangent lines to the logistic loss corresponding to $A^Y(\pi_k) + B^Y(\pi_k) \cdot yf(\boldsymbol{x})$ for $\pi_k \in \boldsymbol{\pi}$. Additionally, the non-differentiable hinge points are highlighted by circles. While the loss functions appear roughly equivalent within the region of the tangent points, the difference is non-negligible above and below these bounds. Notably, the piecewise linear losses diverge slower as $yf(\boldsymbol{x})$ tends to $-\infty$, suggesting the losses may be more robust to outliers (Liu et al., 2011). Additionally, the logistic derived loss functions provide a natural spectrum for comparing the impact of targeting different partitions, $\Omega_{\boldsymbol{\pi}}$, on the same dataset. We explore these issues using simulation in Section 4.6.

4.4 Statistical Properties

We next derive statistical properties for surrogate loss functions to the theoretical loss, $\ell^Y_{\boldsymbol{\pi}}$. In Subsection 4.4.1, we first show that the excess risk with respect to $\ell^Y_{\boldsymbol{\pi}}$ may be bounded by the excess risk of a consistent surrogate loss. Then, in Subsection 4.4.2, we use these risk bounds to derive convergence rates for the empirical minimizer of a surrogate loss to the Bayes optimal rule. Our results generalize and extend those derived for the particular case of rejection-option classification in Herbei and Wegkamp (2006); Bartlett and Wegkamp (2008); Yuan and Wegkamp (2010), to an arbitrary number of boundaries.

4.4.1 Excess Risk Bounds

For a rule $g \in \mathcal{G}_{\boldsymbol{\pi}}$, we define the $\ell^Y_{\boldsymbol{\pi}}$-risk of g to be the expected loss of the rule, denoted by $R(g) = \mathbb{E}_{Y,\boldsymbol{X}}\{\ell^Y_{\boldsymbol{\pi}}(g(\boldsymbol{X}))\}$. In statistical machine learning, a natural measure of the performance of a rule is its excess risk: $\Delta R(g) = R(g) - R(W^*_{\boldsymbol{\pi}})$, where $R(W^*_{\boldsymbol{\pi}}) = \min_{g\in\mathcal{G}_{\boldsymbol{\pi}}} R(g)$ such that $\Delta R(g) \geq 0$. In this subsection, we derive convergence rates on $\Delta R(g)$ for rules obtained using consistent surrogate loss functions. For a surrogate loss ϕ^Y, we similarly define the ϕ-risk and excess ϕ-risk over the class of margin functions, $\mathcal{F}$, to be $Q(f) = \mathbb{E}_{Y,\boldsymbol{X}}\{\phi^Y(Yf(\boldsymbol{X}))\}$ and $\Delta Q(f) = Q(f) - Q(f^*_\phi)$. To obtain convergence rates on $\Delta R(g)$, we first show that under certain conditions, the excess ϕ-risk of a margin function f can be used to bound the corresponding excess $\ell^Y_{\boldsymbol{\pi}}$-risk of $g = C(f;\boldsymbol{\delta})$. Using this bound, we then derive rates of convergence on $\Delta R(g)$ through rates of convergence on $\Delta Q(g)$. The following additional notation is used to denote excess conditional $\ell^Y_{\boldsymbol{\pi}}$-risk and excess conditional ϕ-risk:

$$R_p(g) := \mathbb{E}_{Y|\boldsymbol{X}}\{\ell^Y_{\boldsymbol{\pi}}(g(\boldsymbol{X}))\}, \qquad Q_p(f) := \mathbb{E}_{Y|\boldsymbol{X}}\{\phi^Y(Yf(\boldsymbol{X}))\},$$
$$\Delta R_p(g) := R_p(g) - R_p(W^*_{\boldsymbol{\pi}}), \qquad \Delta Q_p(f) := Q_p(f) - Q_p(f^*_\phi).$$

In the following results, we provide conditions under which there exists some function, $\rho : \mathbb{R} \to \mathbb{R}$, such that $\rho(\Delta Q(f))$ can be used to bound the corresponding $\Delta R(C(f;\boldsymbol{\delta}))$.

Theorem 4.4. *Let ϕ^Y be a consistent surrogate loss for ℓ^Y_π satisfying the conditions for Theorem 4.2 at $\boldsymbol{\delta}$. Furthermore, suppose there exist constants $C > 0$ and $s \geq 1$ such that for all k,*

$$|p(\boldsymbol{X}) - \pi_k|^s \leq C^s \Delta Q_p(\delta_k). \tag{4.13}$$

Then,

$$\Delta R\big(C(f;\boldsymbol{\delta})\big) \leq C[2 \cdot \Delta Q(f)]^{1/s}.$$

The above bound may be tightened as in Yuan and Wegkamp (2010) by the additional assumption:

$$\mathbb{P}\{|p(\boldsymbol{X}) - \pi_k| \leq t\} \leq At^\alpha, \quad k = 1, \ldots, K, \tag{4.14}$$

for some $\alpha \geq 0$, $A \geq 1$. The bound (4.14) generalizes the margin condition introduced by Mammen and Tsybakov (1999) and used in Herbei and Wegkamp (2006).

Theorem 4.5. *In addition to the assumptions of Theorem 4.4, assume that there exists $\alpha \geq 0$ and $A \geq 1$, such that (4.14) holds for $t \in [0, \min_k\{\pi_k - \pi_{k-1}, \pi_{k+1} - \pi_k\})$. Then, for some D depending on A, α,*

$$\Delta R\big(C(f;\boldsymbol{\delta})\big) \leq D \cdot \Delta Q(f)^{1/(s+\beta-\beta s)}$$

where $\beta = \alpha/(1+\alpha)$.

Note that when $\alpha = 0$, Theorem 4.5 provides the same bound as Theorem 4.4. However, as $\alpha \to \infty$, the bound becomes tighter, with $1/(s+\beta-\beta s)$ limiting to 1. While neither result depends explicitly on $\boldsymbol{\pi}$, Theorem 4.5 suggests that tighter bounds may be achieved by only targeting $\boldsymbol{\pi}$ such that the margin condition is satisfied with large α. This reiterates the motivating intuition for our proposed framework, in which we formalize a class of learning problems for settings where more information than hard classification is desired, but soft classification may not be appropriate.

Corresponding values of C and s for the exponential, logistic, squared hinge and DWD losses, are provided in Corollaries 13–16 of Yuan and Wegkamp (2010). In the following result, we derive values of C and s for our class of minimally consistent piecewise linear surrogates presented in Subsection 4.3.2.

Corollary 4.1. *For minimally consistent piecewise linear loss, φ^Y, defined as in (4.11) and (4.12) and satisfying the conditions of Theorem 4.3 for boundaries $\boldsymbol{\pi}$, the inequality (4.13) is satisfied by $s = 1$ and*

$$C = \max\left\{ -\frac{\pi_k}{B^-(\pi_k) \cdot |\delta_k - H_j|} : k = 1, \ldots, K;\ j = 0, \ldots, K \right\},$$

where H_0 is used to denote $A^-(\pi_1)/B^-(\pi_1)$, H_j to denote $H^+(\pi_j, \pi_{j+1})$ for $j = 2, \ldots, K-1$, and H_K to denote $A^+(\pi_K)/B^+(\pi_K)$.

Consider now a sequence of margin functions, $\{f_n\}_{n\geq 1}$. By Theorems 4.4 and 4.5, to show that the excess $\ell^Y_{\boldsymbol{\pi}}$-risk, $\Delta R(C(f_n;\boldsymbol{\delta}))$, converges to 0 as $n \to \infty$, it suffices to show that $\Delta Q(f_n) \to 0$ as $n \to \infty$. In the following results, we derive convergence rates for $\Delta R\big(C(\cdot;\boldsymbol{\delta})\big)$ for the sequence of functions, $\{\hat{f}_n\}_{n\geq 1}$, where $\hat{f}_n$ is used to denote the empirical minimizer of the surrogate loss over a training set of size n.

4.4.2 Rates of Convergence

In this subsection, we derive convergence results for two classes of surrogate loss functions separately. We first consider Lipschitz continuous and differentiable surrogate loss functions which satisfy a modulus of convexity condition specified below. Examples of such loss functions include the exponential, logistic, squared hinge and DWD losses. We then separately consider the class of piecewise linear surrogates described in Section 4.3.

Let ϕ^Y denote a Lipschitz continuous and differentiable surrogate loss function. Assume that the corresponding ϕ-risk, $Q(\cdot)$, has modulus of convexity,

$$\delta(\epsilon) = \inf\left\{ \frac{Q(f) + Q(g)}{2} - Q\left(\frac{f+g}{2}\right) : \mathbb{E}[(f-g)^2(\boldsymbol{X})] \geq \epsilon^2 \right\} \tag{4.15}$$

satisfying $\delta(\epsilon) > c\epsilon^2$ for some $c > 0$. Furthermore, let $L < \infty$ denote the Lipschitz constant, such that $|\phi^y(\boldsymbol{x}) - \phi^y(\boldsymbol{x}')| \leq L|x - x'|$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}$ and $y = +1, -1$. Letting $\mathcal{F}_B$ denote the class of uniformly bounded functions such that $|f| \leq B$ for all $f \in \mathcal{F}_B$, we use $N_n = N(\frac{1}{n}, L_\infty, \mathcal{F}_B)$ to denote the cardinality of the set of closed balls with radius $\frac{1}{n}$ in L_∞ needed to cover $\mathcal{F}_B$. Finally, as stated above, let $\hat{f}_n = \operatorname{argmin}_{f \in \mathcal{F}_B} \sum_{i=1}^n \phi^{y_i}(y_i f(\boldsymbol{x}_i))$ denote the empirical minimizer of ϕ^Y over the training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. For the following corollary, we make use of Theorem 18 from Yuan and Wegkamp (2010) which provides a bound on the expected estimation error, $Q(\hat{f}_n) - \inf_{f \in \mathcal{F}_B} Q(f)$, for consistent loss functions satisfying the modulus of convexity condition stated above. Combining Theorem 18 of Yuan and Wegkamp (2010) with the excess risk bounds of Theorems 4.4 and 4.5, we obtain the following result.

Corollary 4.2. *If ϕ^Y satisfies the assumptions of Theorems 4.2 and 4.4, and has modulus of convexity (4.15) satisfying $\delta(\epsilon) > c\epsilon^2$ for some $c > 0$, then with probability at least $1 - \gamma$,*

$$\Delta R\big(C(\hat{f}_n; \boldsymbol{\delta})\big) \leq C \cdot 2^{1/s} \left\{ \inf_{f \in \mathcal{F}_B} \Delta Q(f) + \frac{3L}{n} + 8\left(\frac{L^2}{2c} + \frac{LB}{3}\right) \frac{\log(N_n/\gamma)}{n} \right\}^{1/s}.$$

Furthermore, if the generalized margin condition of Theorem 4.5 holds, then with probability at least $1 - \gamma$,

$$\Delta R\big(C(\hat{f}_n; \boldsymbol{\delta})\big) \leq D \left\{ \inf_{f \in \mathcal{F}_B} \Delta Q(f) + \frac{3L}{n} + 8\left(\frac{L^2}{2c} + \frac{LB}{3}\right) \frac{\log(N_n/\gamma)}{n} \right\}^{1/(s+\beta-\beta s)}, \tag{4.16}$$

for constants $C, D > 0$ defined as in Theorems 4.4 and 4.5.

From the bound on excess risk obtained in Corollary 4.2, corresponding rates of convergence can be derived based on the cardinality, N_n, of the class of functions, $\mathcal{F}_B$.

Due to the non-differentiability of the loss at hinge points, our class of piecewise linear surrogates do not satisfy the modulus of convexity condition (4.15). The following theorem provides separate convergence results for our class of minimally consistent piecewise linear surrogates. Again, we use $\mathcal{F}_B$ to denote a class of uniformly bounded functions, and let $\hat{f}_n = \text{argmin}_{f \in \mathcal{F}_B} \sum_{i=1}^n \varphi^{y_i}(y_i f(\boldsymbol{x}_i))$ denote the empirical minimizer of φ^Y.

Theorem 4.6. *If φ^Y is a minimally consistent piecewise linear loss satisfying the conditions of Theorem 4.3, satisfying the generalized margin condition of Theorem 4.5, then with probability at least $1-\gamma$,*

$$\Delta Q(\hat{f}_n) \leq \frac{3L}{n} + \frac{4LB}{3} \cdot G(\gamma) + \left(\left(\frac{4LB}{3} \cdot G(\gamma) \right)^2 + 8 \cdot B' \cdot G(\gamma) \right)^{1/2},$$

where $G(\gamma) = \log(N_n/\gamma)/n$, and $B' > 0$ is some constant depending on B, φ^Y, and margin constants A, α.

Combining Theorems 4.4, 4.5, and 4.6, we obtain the following corollary.

Corollary 4.3. *If φ^Y is a minimally consistent piecewise linear loss satisfying the assumptions of Theorems 4.2, 4.4, and 4.5, then with probability at least $1-\gamma$,*

$$\Delta R\big(C(\hat{f}_n; \boldsymbol{\delta})\big) \leq D \left\{ \frac{3L}{n} + \frac{4LB}{3} \cdot G(\gamma) + \left(\left(\frac{4LB}{3} \cdot G(\gamma) \right)^2 + 8 \cdot B' \cdot G(\gamma) \right)^{1/2} \right\}^{1/(s+\beta-\beta s)}, \tag{4.17}$$

for constants $C, D > 0$ defined as in Theorems 4.4 and 4.5.

As in Theorem 4.5, while the convergence rate of Theorem 4.6 does not depend on $\boldsymbol{\pi}$ explicitly, it does depend on the parameters of the margin condition (4.14). Therefore, Theorem 4.6 further suggests the advantage of targeting $\boldsymbol{\pi}$ for which the data show strong separation with large α. Furthermore, in contrast to Theorem 18 of Yuan and Wegkamp (2010) which provides a bound on the expected estimation error, Theorem 4.6 bounds the total φ^Y-risk, including both the expected estimation error, and expected approximation error of the class of functions $\mathcal{F}_B$. As a result, while

the bounds in Corollary 4.2 include the separate approximation error term, $\inf_{f\in\mathcal{F}_B}\Delta Q(f)$, the piecewise linear bound in Corollary 4.3, does not.

Based on the bounds in (4.16) and (4.17), rates of convergence can be obtained as in Yuan and Wegkamp (2010). As an example, we consider the case when $\mathcal{F}_\mathcal{B}$ is the class of linear combinations of decision stumps, f_λ,

$$f_\lambda(x) = \sum_{j=1}^{M} \lambda_j f_j(x)$$

where $\sum_j |\lambda_j| \leq B$, and $|f_j| < 1$. By (4.16) and (4.17), the same rate, $(M \log n/n)^{1/(s+\beta-\beta s)}$, can be obtained as in Yuan and Wegkamp (2010) for both classes of surrogate losses considered above.

4.5 Computational Algorithm

For a piecewise linear surrogate, φ^Y, and convex penalty, $J(f)$, the objective (4.1) is a non-differentiable convex problem. Several approaches have been proposed for solving the similar non-differentiable and convex SVM objective, most commonly by reformulating (4.1) as a quadratic program (QP) with $2n$ constraints. The penalized objective (4.1) with φ^Y may also be formulated as a QP with $(K+1)n$ constraints. However, as with the SVM problem, the complexity of the problem grows almost cubically with the number of constraints, making the problem computationally intensive for moderately large K and n (Bottou and Lin, 2007). We therefore propose a projected sub-gradient descent algorithm similar to the PEGASOS algorithm (Shalev-Shwartz et al., 2010).

We first rewrite (4.1) with piecewise linear surrogate, φ^Y defined as in (4.11) and (4.12) as:

$$\min_{h,b} \quad \frac{1}{n}\sum_{i=1}^{n}\Big(\max_{k=1,\ldots,K}\{A^{y_i}(\pi_k) + B^{y_i}(\pi_k)\cdot y_i(h(\boldsymbol{x}_i)+b)\}\Big)_+ + \frac{\lambda}{2}\|h\|^2_{\mathcal{H}}, \tag{4.18}$$

where $(z)_+ = \max\{0, z\}$, and $\mathcal{H}$ is some Reproducing Kernel Hilbert Space (RKHS) with norm $\|\cdot\|_\mathcal{H}$ and corresponding kernel function $K : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$. Commonly, the margin function is formulated with a non-penalized intercept parameter, b. A more complete review of RKHS may be found in Aronszajn (1950) and Wahba (1999). In margin-based learning, kernel methods are commonly used to estimate non-linear classification boundaries. In the case of linear learning, i.e.

$h(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle$ for $\boldsymbol{w} \in \mathbb{R}^p$, the penalty $\|h\|^2_{\mathcal{H}}$ reduces to $\|\boldsymbol{w}\|^2$ and (4.18) may be written as:

$$\min_{\boldsymbol{w},b} \quad \frac{1}{n}\sum_{i=1}^{n}\Big(\max_{k=1,\ldots,K}\{A^{y_i}(\pi_k) + B^{y_i}(\pi_k)\cdot y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b)\}\Big)_+ + \frac{\lambda}{2}\|\boldsymbol{w}\|^2.$$

We next describe our iterative algorithm for the linear learning setting. Let $\boldsymbol{w}^{(m)}$ and $b^{(m)}$ denote the estimated parameters at the m-th iteration. Furthermore, at each iteration, let B_i^* denote the sub-gradient of φ^{y_i} at $\langle \boldsymbol{w}^{(m)}, \boldsymbol{x}_i\rangle + b^{(m)}$ for $i = 1, \ldots, n$. Using a decreasing step-size parameter, $\eta_m = (\lambda m)^{-1}$, we iterate the following updates until $\boldsymbol{w}^{(m)}$ and $b^{(m)}$ converge:

1. $\boldsymbol{w}^{(m)} = \boldsymbol{w}^{(m-1)} + \eta_m(\frac{1}{n}\sum_i B_i^* y_i \boldsymbol{x}_i - \lambda \boldsymbol{w}^{(m-1)})$,
2. $b^{(m)} = b^{(m-1)} + \eta_m(\frac{1}{n}\sum_i B_i^* y_i)$,
3. $[\boldsymbol{w}^{(m)}, b^{(m)}] = \min\{1, \frac{\lambda^{-1/2}}{\|[\boldsymbol{w}^{(m)}, b^{(m)}]\|}\}[\boldsymbol{w}^{(m)}, b^{(m)}]$,

where B_i^* is used to denote the sub-gradient of φ^{y_i} at $y_i(\langle \boldsymbol{x}_i, \boldsymbol{w}\rangle + b)$. The final projection step is included to ensure $\|[\boldsymbol{w}^{(m)}, b^{(m)}]\|^2 \leq \lambda^{-1}$ at each iteration (Calamai and Moré, 1987; Shalev-Shwartz et al., 2010). In the following section, we apply our projected sub-gradient descent algorithm to simulated datasets to illustrate the utility of our class of problems.

4.6 Simulations

In this section, we use simulations to illustrate the performance achieved by targeting different binary learning problems. Namely, we compare the performance of several minimal consistent piecewise linear losses against the standard logistic classifier, when the underlying conditional class probability, $p(\boldsymbol{X})$, is piecewise constant. Piecewise linear loss functions are derived from the logistic loss as described in Section 4.3.3, and the sets of boundaries, $\boldsymbol{\delta}$, are chosen by the tangent points to the logistic loss. In each simulation, we consider piecewise linear losses with $\boldsymbol{\pi}_1 = \{1/2\}$, $\boldsymbol{\pi}_2 = \{1/3, 2/3\}$, and $\boldsymbol{\pi}_3 = \{1/4, 2/4, 3/4\}$. All methods are tuned over a grid of penalty parameters $\lambda \in \{2^{-15}, 2^{-14}, \ldots, 2^{10}\}$, using training and tuning sets of 100 observations each. Piecewise linear classifiers and the logistic classifier are tuned with respect to the correspond theoretical loss (4.7) and likelihood function, respectively. The performance of each estimated model is evaluated using a test set of 10,000 observations. Each simulation was replicated 100 times.

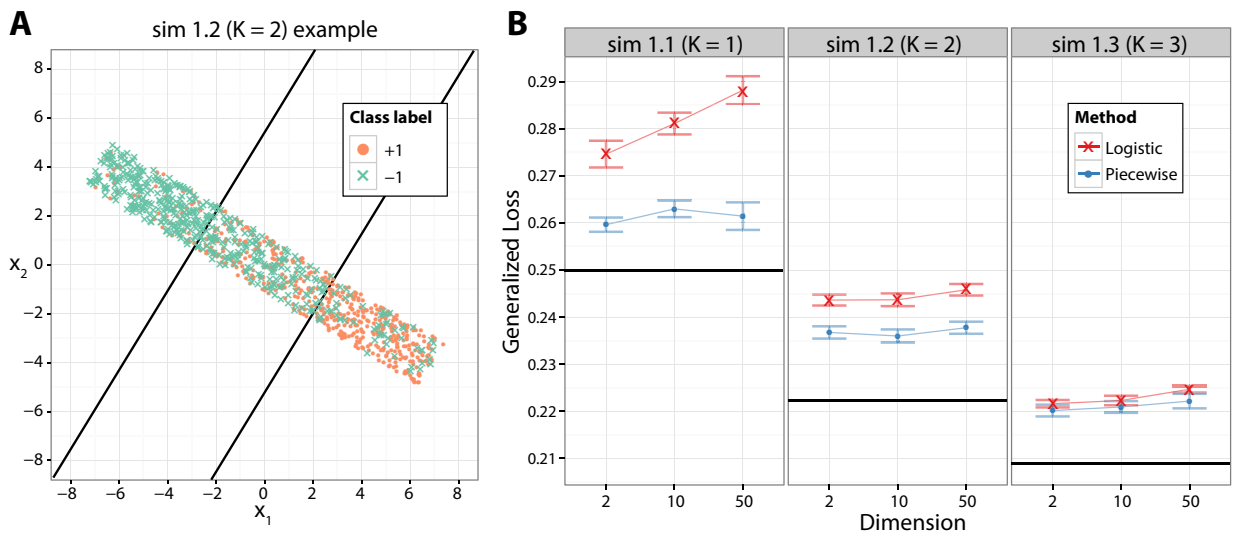

Figure 4.6: (A) Sample dataset of 1000 observations drawn from the generating distribution for Setting 1.2. The two Bayes optimal boundaries separating the three regions of constant $p(\boldsymbol{X})$ are shown with black lines. (B) Comparison of the performance of the piecewise linear and logistic classifiers for the three settings of Setting 1 across varying dimension. In each panel, the median loss and standard error over 100 replications is shown along with the Bayes minimal loss in black.

4.6.1 Setting 1

In this setting, data are simulated uniformly from $[-8, 8] \times [-1, 1]^{p-1}$ for $p = 2, 10, 50$, subject to a random rotation in the p-dimensional space. We consider three variations of this setting, in which the data were simulated with underlying conditional class probability, defined with respect to the sampling space prior to rotation:

1.1 $p(\boldsymbol{X}) = \frac{1}{4}\mathbf{I}\{x_1 \in [-8, 0)\} + \frac{3}{4}\mathbf{I}\{x_1 \in [0, 8]\}$,

1.2 $p(\boldsymbol{X}) = \frac{1}{6}\mathbf{I}\{x_1 \in [-8, -\frac{8}{3})\} + \frac{3}{6}\mathbf{I}\{x_1 \in [-\frac{8}{3}, \frac{8}{3})\} + \frac{5}{6}\mathbf{I}\{x_1 \in [\frac{8}{3}, 8]\}$,

1.3 $p(\boldsymbol{X}) = \frac{1}{8}\mathbf{I}\{x_1 \in [-8, -4)\} + \frac{3}{8}\mathbf{I}\{x_1 \in [-4, 0)\} + \frac{5}{8}\mathbf{I}\{x_1 \in [0, 4)\} + \frac{7}{8}\mathbf{I}\{x_1 \in [4, 8]\}$.

Settings 1.1, 1.2, and 1.3 have one, two and three natural boundaries due to the piecewise constant form of $p(\boldsymbol{X})$. In Figure 4.6A, we show 1000 observations drawn from Setting 1.2, with observations from the positive and negative class shown in orange and green. The Bayes optimal boundaries are also shown in black. For Settings 1.1, 1.2, and 1.3, we use the piecewise linear losses with boundaries at $\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$, and $\boldsymbol{\pi}_3$, respectively. In each setting, the performance of the piecewise linear and logistic classifiers is evaluated using the theoretical loss for boundaries at $\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$, and $\boldsymbol{\pi}_3$. In

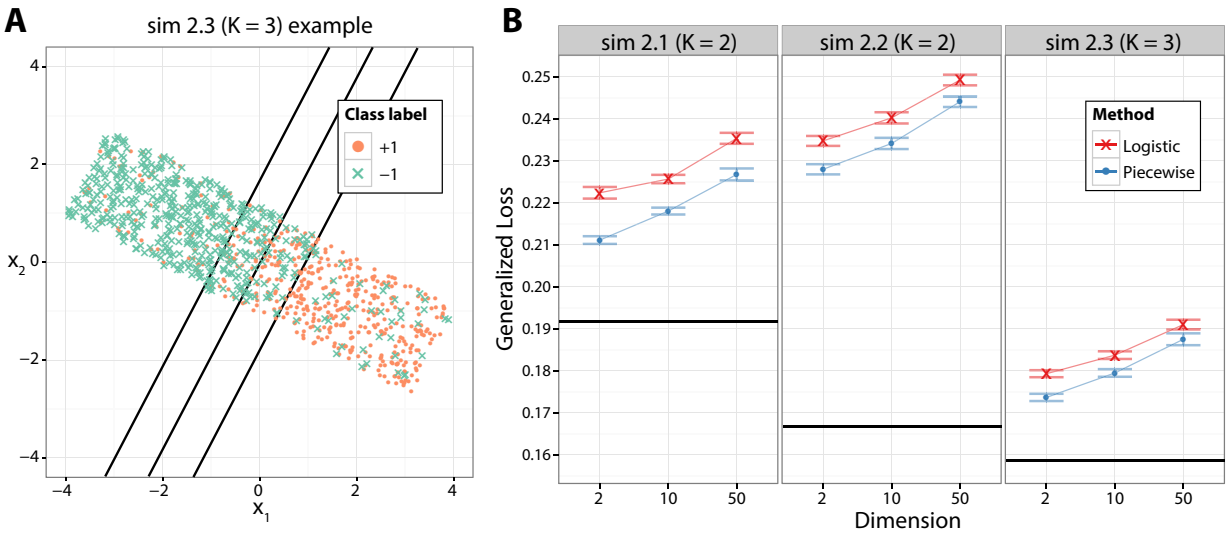

Figure 4.7: (A) Sample dataset of 1000 observations drawn from the generating distribution for Setting 2.3. The three Bayes optimal boundaries separating the four regions of constant $p(\boldsymbol{X})$ are shown with black lines. (B) Comparison of the performance of the piecewise linear and logistic classifiers for the three settings of Setting 2 across varying dimension. In each panel, the median loss and standard error over 100 replications is shown along with the Bayes minimal loss in black.

these simulations, we aim to illustrate the advantage of minimizing and tuning with respect to an appropriate theoretical loss, which matches the underlying form of the data.

The results are shown in Figure 4.6B, along with the Bayes minimal loss, which provides a lower bound on the theoretical loss in each setting. In all settings, the piecewise linear classifier outperforms the logistic classifier, with the improvement decreasing as the number of boundaries, K increases. This makes intuitive sense, as the piecewise linear loss converges to the logistic loss as $K \to \infty$. The most significant improvement is seen in Setting 1.1, in which the piecewise linear classifier and theoretical loss correspond to the standard SVM and misclassification error. These results confirm previous results highlighting the advantage of hard classifiers over soft classifiers when the underlying $p(\boldsymbol{X})$ is piecewise constant (Liu et al., 2011). Furthermore, the complete set of results illustrates the transition of this behavior as the number of boundaries increases.

4.6.2 Setting 2

In Setting 1, the piecewise constant regions of $p(\boldsymbol{X})$ were of equal size. In our second set of simulations, we consider unequally spaced conditional class probabilities. Observations were uniformly sampled over $[-4, 4] \times [-1, 1]^{p-1}$, for $p = 2, 10, 50$, again subject to a random rotation. The follow-

ing conditional class probabilities were considered, again, with respect to the sampling space prior to rotation:

2.1 $p(\boldsymbol{X}) = \frac{1}{6}\mathbf{I}\{x_1 \in [-4, -0.6)\} + \frac{3}{6}\mathbf{I}\{x_1 \in [-0.6, 0.6)\} + \frac{5}{6}\mathbf{I}\{x_1 \in [0.6, 4]\}$,

2.2 $p(\boldsymbol{X}) = \frac{1}{6}\mathbf{I}\{x_1 \in [-4, -2)\} + \frac{3}{6}\mathbf{I}\{x_1 \in [-2, 0)\} + \frac{5}{6}\mathbf{I}\{x_1 \in [0, 4]\}$,

2.3 $p(\boldsymbol{X}) = \frac{1}{8}\mathbf{I}\{x_1 \in [-4, -0.8)\} + \frac{3}{8}\mathbf{I}\{x_1 \in [-0.8, 0)\} + \frac{5}{8}\mathbf{I}\{x_1 \in [0, 0.8)\} + \frac{7}{8}\mathbf{I}\{x_1 \in [0.8, 4]\}$.

In Settings 2.1 and 2.3, we consider $p(\boldsymbol{X})$ with heavy tails, and in Setting 2.2, we consider the case with asymmetric $p(\boldsymbol{X})$. A sample of 1000 observations drawn from Setting 2.3 is shown in Figure 4.7A, with the Bayes optimal boundaries in black. For Settings 2.1, 2.2, and 2.3, we use the piecewise linear losses with boundaries at $\boldsymbol{\pi}_2$, $\boldsymbol{\pi}_2$, and $\boldsymbol{\pi}_3$, respectively. The performance of the piecewise linear and logistic classifiers is again evaluated using the corresponding theoretical loss function. Simulation results are shown in Figure 4.7B. As in Setting 1, the piecewise linear classifier outperforms the logistic classifier in all cases. Again, the improvement is greater in Settings 2.1 and 2.2 than in Setting 2.3, as the piecewise linear loss converges to the logistic loss with increasing K.

4.7 Real Data Analysis

In this section, we apply the proposed interval estimation procedure to a MRI dataset of healthy normal control (NC) and early Alzheimer's disease (AD) subjects. Data were obtained from the ADNI database (`adni.loni.usc.edu`). The ADNI was launched in 2003 by the National Institute on Aging (NIA), the National Institute of Biomedical Imaging and Bioengineering (NIBIB), the Food and Drug Administration (FDA), private pharmaceutical companies and non-profit organizations as a $60 million, 5-year public-private partnership.

The dataset we use consists of 93 MRI features measured for 225 NC and 186 AD subjects, and was processed as described in Yu et al. (2014). As in Section 4.6, the logistic-derived piecewise linear loss is used to target the conditional class probability of AD at $\boldsymbol{\pi} = \{1/4, 2/4, 3/4\}$. Two-fold cross validation is used to determine the optimal λ over $\{2^{-15}, 2^{-14}, \ldots, 2^5\}$. The first two principal components (PCs) of the 411 NC and AD subjects are shown in Figure 4.8A, along with

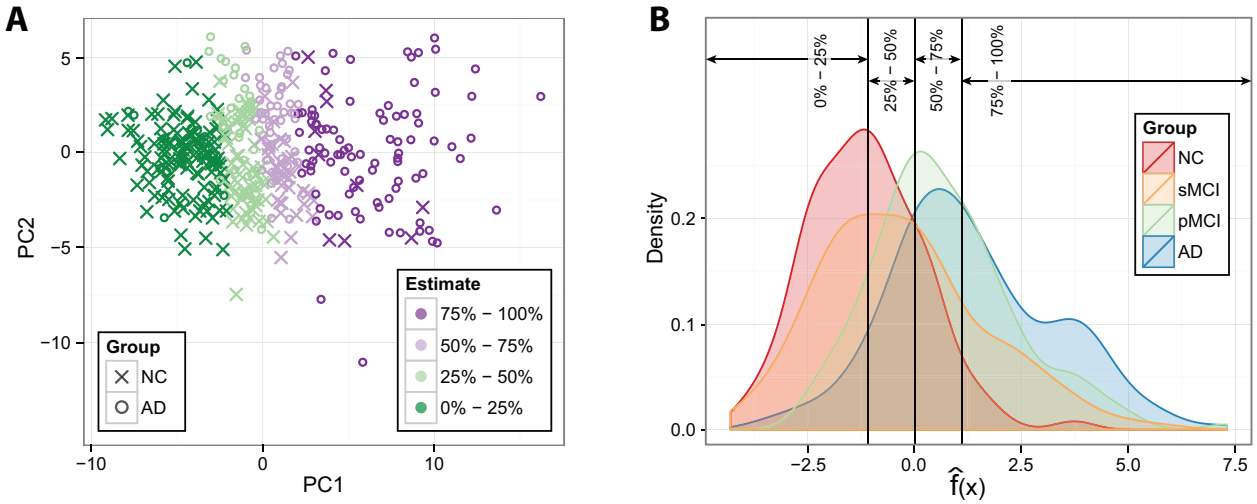

Figure 4.8: Analysis of ADNI MRI dataset with $\boldsymbol{\pi} = \{0.25, 0.5, 0.75\}$. (A) Scatterplot of first two PCs for AD and NC subjects colored by estimated interval. (B) Density plots of predicted $\hat{f}(\boldsymbol{x})$ for AD, NC, and two intermediary subject groups, sMCI and pMCI. Corresponding interval cutoffs are shown with vertical lines.

the estimated interval for each subject. Interestingly, the four distinct probability groups appear to separate along the first PC direction.

In addition to NC and AD subjects, the dataset also includes subjects with mild cognitive impairment (MCI), further classified as either progressive MCI (pMCI, 167 subjects) or stable MCI (sMCI, 226 subjects), depending on whether or not the subject progressed to develop AD during the study. The sMCI and pMCI may be considered as intermediary states between the NC and AD subjects. As such, in Figure 4.8B, we show the distribution of margin values, $\hat{f}(\boldsymbol{x})$, for NC, sMCI, pMCI, and AD subjects to investigate the transition between the four distinct groups. The corresponding interval boundaries are shown by vertical lines. Interestingly, while not well-differentiated, the four groups appear to peak within each of the four intervals, with the densities shifting in the expected order. Overall, our method appears to appropriately divide the subject according to the severity of the disease.

4.8 Discussion

Supervised learning tasks with a discrete class label are commonly encountered in practice. Several problems have been formally defined and studied within this context, including hard, soft, and rejection-option classification. In this chapter, we introduce a unified framework of binary

learning tasks targeting partial or complete estimation of the conditional class probability, $p(\boldsymbol{X})$, which encompassing these problems. In contrast to previous frameworks connecting hard and soft classification, our approach spans a space of learning problems, rather than specific loss functions or classification methods. Our approach thus provides a unique perspective to study the transition between hard and soft classification.

We formalize our family of binary learning problems through a unified theoretical loss (4.7), a corresponding margin based relaxation (4.9), and a proposed class of minimally consistent piecewise linear surrogates. Simulation studies using the class of piecewise linear loss functions reinforce previous results on hard and soft classification, and illustrate the transitional behavior between the class of problems. Finally, an application of our interval estimation approach to a MRI dataset from the ADNI study further illustrates the utility of our proposed class of problems.

4.9 Proofs

4.9.1 Proof of Theorem 4.1

Let $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$ for some $K \geq 1$ such that $0 < \pi_1 < \ldots < \pi_K < 1$. Furthermore, let $h \in \{0, \ldots, K\}$ denote the index for some predicted $\omega_h \in \Omega_{\boldsymbol{\pi}}$. Then,

$$\begin{aligned}\mathbb{E}_{Y|\boldsymbol{X}}\left\{\ell^Y_{\boldsymbol{\pi}}(\omega_h)\right\} &= p(\boldsymbol{X}) \cdot \ell^+_{\boldsymbol{\pi}}(\omega_h) + (1 - p(\boldsymbol{X})) \cdot \ell^-_{\boldsymbol{\pi}}(\omega_h). \\ &\propto p(\boldsymbol{X}) \sum_{k=h+1}^{K+1} (1 - \pi_k) + (1 - p(\boldsymbol{X})) \sum_{k=1}^{h} \pi_k,\end{aligned}$$

Letting $\pi_0 = 0$, $\pi_{K+1} = 1$, we can express the above as:

$$\mathbb{E}_{Y|\boldsymbol{X}}\left\{\ell^Y_{\boldsymbol{\pi}}(\omega_h)\right\} = \sum_{k=0}^{K+1} \Big\{p(\boldsymbol{X})(1 - \pi_k) \cdot \mathbf{I}_{\{k>h\}} + \pi_k(1 - p(\boldsymbol{X})) \cdot \mathbf{I}_{\{k\leq h\}}\Big\}.$$

The sum is minimized by choosing h such that $p(\boldsymbol{X})(1 - \pi_k) \geq \pi_k(1 - p(\boldsymbol{X}))$ for all $k \leq h$ and $p(\boldsymbol{X})(1 - \pi_k) \leq \pi_k(1 - p(\boldsymbol{X}))$ for all $k > h$. Thus, the optimal solution is given by $h^* = \text{argmax}_k\{\pi_k < p(\boldsymbol{X})\}$. The equivalence between ω_{h^*} and $\sum_{k=0}^{K} \omega_k \cdot \mathbf{I}\{p(\boldsymbol{X}) \in \omega_k\}$ is immediate from the fact that $p(\boldsymbol{X}) \in (\pi_{h^*}, \pi_{h^*+1}] = \omega_{h^*}$, and the additional assumption that $p(\boldsymbol{X}) \neq \pi_k$ a.s. for all k.

4.9.2 Proof of Theorem 4.2

Let $\boldsymbol{\pi}$ and $\boldsymbol{\delta}$ be appropriately defined boundaries in $(0,1)$ and $\mathbb{R}$. Note that surrogate losses, ϕ^+, ϕ^- are consistent for boundaries at $\boldsymbol{\pi}$ with $\boldsymbol{\delta}$, i.e. $\boldsymbol{\pi}, \boldsymbol{\delta}$-consistent, if and only if they are π_k, δ_k-consistent for each k separately. Thus, conditions for $\boldsymbol{\pi}, \boldsymbol{\delta}$-consistency are simply the union of the conditions for π_k, δ_k-consistency. Necessary and sufficient conditions for ϕ^Y to be π_k, δ_k-consistent were provided by Theorem 1 of [12].

4.9.3 Proof of Theorem 4.3

Let $\boldsymbol{\pi}$ be an appropriately defined set of boundaries in $(0,1)$. Assume φ^Y to be defined as in (4.11) and (4.12) such that (C1)–(C3) are satisfied. We wish to show that for all $\pi_k \in \boldsymbol{\pi}$, there exists some δ_k such that (4.10) is satisfied, and furthermore, that there does not exist any δ such that (4.10) is satisfied for $\pi \in (0,1) \setminus \boldsymbol{\pi}$. Equivalently, we wish to show that $\varphi^{-\prime}(x)/(\varphi^{-\prime}(x) + \varphi^{+\prime}(x))$ only takes values in $\boldsymbol{\pi}$ over the set of x such that $\varphi^{-\prime}(x) < 0$ and $\varphi^{+\prime}(x) < 0$ are defined. Note that $\varphi^{+\prime}$ and $\varphi^{-\prime}$ are only undefined at the hinge points, $H^Y(\pi_k, \pi_{k+1})$, $A^-(\pi_1)/B^-(\pi_1)$, and $A^+(\pi_K)/B^+(\pi_K)$. By (C2), the set of possible $\varphi^{+\prime}$, $\varphi^{-\prime}$ pairs are given by:

$\varphi^{+\prime}$:	$B^+(\pi_1)$	$B^+(\pi_1)$	$\cdots$	$B^+(\pi_K)$	0
$\varphi^{-\prime}$:	0	$B^-(\pi_1)$	$\cdots$	$B^-(\pi_K)$	$B^-(\pi_K)$

Excluding the cases when $\varphi^{+\prime}(x) = 0$ or $\varphi^{-\prime}(x) = 0$, the set of possible consistent boundaries satisfying (4.10) are given by:

$$\frac{\varphi^{-\prime}(x)}{\varphi^{-\prime}(x) + \varphi^{+\prime}(x)} = \frac{B^{-\prime}(\pi_k)}{B^{-\prime}(\pi_k) + B^{+\prime}(\pi_k)} = \pi_k \qquad \text{for } k = 1, \ldots, K,$$

where the final equality is given by (C3).

4.9.4 Proof of Proposition 4.1

Let $\boldsymbol{\pi}$ be an appropriately defined set of boundaries in $(0,1)$. We wish to show that (C1)–(C3) of Theorem 4.3 are satisfied for $A^+(\pi) = A^-(1-\pi) = -\pi\log(\pi) - (1-\pi)\log(1-\pi)$, and $B^+(\pi) = B^-(1-\pi) = -(1-\pi)$.

Trivially, (C1) is satisfied, as $B^+(\pi) = \pi - 1$ and $B^-(\pi) = -\pi$ are non-decreasing and non-increasing, respectively, in π. To show that (C2) is satisfied, we derive the hinge points for the

positive and negative class losses:

$$H^+(\pi, \pi') = \frac{A^+(\pi) - A^+(\pi')}{B^+(\pi') - B^+(\pi)} = \frac{A^+(\pi) - A^+(\pi')}{\pi' - \pi}$$
$$H^-(\pi, \pi') = \frac{A^-(\pi) - A^-(\pi')}{B^-(\pi') - B^-(\pi)} = -\frac{A^+(\pi) - A^+(\pi')}{\pi' - \pi},$$

where the final equality is obtained by noting $A^+(\pi) = A^+(1-\pi)$. The first equality of (C2) is clearly satisfied by the above derivations. We next show that the remaining three inequalities of (C2) are also satisfied. Let $k \in \{2, \ldots, K-1\}$. By the concavity of $A^+(\pi)$:

$$\begin{aligned} H^+(\pi_{k-1}, \pi_k) &= \frac{A^+(\pi_{k-1}) - A^+(\pi_k)}{\pi_k - \pi_{k-1}} \\ &= -\frac{A^+(\pi_k) - A^+(\pi_{k-1})}{\pi_k - \pi_{k-1}} \\ &< -(A^+)'(\pi_k) \\ &< -\frac{A^+(\pi_{k+1}) - A^+(\pi_k)}{\pi_{k+1} - \pi_k} = H^+(\pi_k, \pi_{k+1}), \end{aligned}$$

Similarly, by the convexity of $A^-(\pi)$ and the fact that $\lim_{\pi \to 0} A^+(\pi) = \lim_{\pi \to 1} A^+(\pi) = 0$, we have:

$$\begin{aligned} \frac{A^-(\pi_1)}{B^-(\pi_1)} &= -\frac{A^-(\pi_1) - \lim_{\pi \to 0} A^-(\pi)}{\pi_1 - 0} \\ &< -(A^-)'(\pi_1) \\ &< -\frac{A^-(\pi_1) - A^-(\pi_2)}{\pi_1 - \pi_2} = H^+(\pi_1, \pi_2) \\ \frac{A^+(\pi_K)}{B^+(\pi_K)} &= -\frac{A^+(\pi_K) - \lim_{\pi \to 1} A^+(\pi)}{\pi_K - 1} \\ &> -(A^+)'(\pi_K) \\ &> -\frac{A^+(\pi_{K-1}) - A^+(\pi_K)}{\pi_{K-1} - \pi_K} = H^-(\pi_{K-1}, \pi_K). \end{aligned}$$

Thus (C2) is satisfied. Finally, (C3) holds, since for any $k = 1, \ldots, K$:

$$\frac{B^-(\pi_k)}{B^-(\pi_k) + B^+(\pi_k)} = \frac{-\pi_k}{-\pi_k - (1 - \pi_k)} = \pi_k.$$

4.9.5 Proof of Theorem 4.4

Let ϕ^Y be a consistent surrogate loss for appropriately defined boundaries $\boldsymbol{\pi}$ in $(0,1)$ at $\boldsymbol{\delta}$. First, note that the excess condition ϕ-risk for a rule $g \in \mathcal{G}$ may be written as:

$$R_p(g(\boldsymbol{x})) = \frac{2}{K}\Big[(1-p(\boldsymbol{x}))\sum_k \pi_k \mathbf{I}\{g(\boldsymbol{x}) > \pi_k\} + p(\boldsymbol{x})\sum_k (1-\pi_k)\mathbf{I}\{g(\boldsymbol{x}) \leq \pi_k\}\Big].$$

Consider a candidate rule $g \in \mathcal{G}$, and recall the Bayes optimal rule over $\mathcal{G}$, $W^*_{\boldsymbol{\pi}}(\boldsymbol{x})$, defined in Theorem 4.1. Suppose that $\boldsymbol{x} \in \mathcal{X}$ is such that $g(\boldsymbol{x}) > W^*_{\boldsymbol{\pi}}(\boldsymbol{x})$. Then, letting

$$\mathcal{K} = \begin{cases} \{k : g(\boldsymbol{X}) \leq \pi_k < W^*_{\boldsymbol{\pi}}(\boldsymbol{X})\} & \text{if } W^*_{\boldsymbol{\pi}}(\boldsymbol{X}) > g(\boldsymbol{X}) \\ \{k : W^*_{\boldsymbol{\pi}}(\boldsymbol{X}) \leq \pi_k < g(\boldsymbol{X})\} & \text{if } W^*_{\boldsymbol{\pi}}(\boldsymbol{X}) < g(\boldsymbol{X}) \\ \emptyset & \text{otherwise} \end{cases},$$

the excess condition ϕ-risk may be expressed as:

$$\begin{aligned} \Delta R_p(g) &= \frac{2}{K}\Big[(1-p(\boldsymbol{x}))\sum_k \pi_k \mathbf{I}\{\pi_k : W^*_{\boldsymbol{\pi}}(\boldsymbol{x}) \leq \pi_k < g(\boldsymbol{x})\} \\ &\qquad - p(\boldsymbol{x})\sum_k (1-\pi_k)\mathbf{I}\{\pi_k : W^*_{\boldsymbol{\pi}}(\boldsymbol{x}) \leq \pi_k < g(\boldsymbol{x})\}\Big] \\ &= \frac{2}{K}\sum_{\mathcal{K}}\big[(1-p(\boldsymbol{x}))\pi_k - p(\boldsymbol{x})(1-\pi_k)\big] \\ &= \frac{2}{K}\sum_{\mathcal{K}}\big[\pi_k - p(\boldsymbol{x})\big]. \end{aligned}$$

Similarly, for $g(\boldsymbol{x}) < W^*_{\boldsymbol{\pi}}(\boldsymbol{x})$, $\Delta R_p(g) = \frac{2}{K}\sum_{\mathcal{K}}\big[p(\boldsymbol{x}) - \pi_k\big]$. If $g(\boldsymbol{x}) = W^*_{\boldsymbol{\pi}}(\boldsymbol{x})$, we have that $\Delta R_p(g) = 0$, such that:

$$\Delta R_p(g) = \tfrac{2}{K}\sum_{\mathcal{K}}\big|p(\boldsymbol{x}) - \pi_k\big|,$$

for all $\boldsymbol{x} \in \mathcal{X}$.

By the stated assumptions, for $g(\boldsymbol{x}) = C(f(\boldsymbol{x}); \boldsymbol{\delta}) \in \mathcal{G}$, we immediately have the following result:

$$\begin{aligned} (\Delta R_p(g))^s &= \Big(\frac{2}{K}\sum_{\mathcal{K}}|p(\boldsymbol{X}) - \pi_k|\Big)^s \\ &\leq \frac{2}{K}\sum_{\mathcal{K}}|p(\boldsymbol{X}) - \pi_k|^s \end{aligned}$$

$$\leq \frac{2}{K} C^s \sum_{\mathcal{K}} \Delta Q_p(\delta_k)$$

$$\Delta R_p(g) \leq C\Big(\frac{2}{K} \sum_{\mathcal{K}} \Delta Q_p(\delta_k)\Big)^{1/s}.$$

Since $\Delta Q_p \geq 0$, it suffices to show that $\sum_{\mathcal{K}} \Delta Q_p(\delta_k) \leq K \cdot \Delta Q_p(f)$. Since $|\mathcal{K}| \leq K$, we complete the proof by showing $\Delta Q_p(f) \geq \Delta Q_p(\delta_k)$ for all $k \in \mathcal{K}$. Without loss of generality, suppose $\boldsymbol{x}$ is such that $g(\boldsymbol{x}) < W^*_{\boldsymbol{\pi}}(\boldsymbol{x})$ and let $k \in \mathcal{K}$. Note that $\pi_k < g(\boldsymbol{x})$ is equivalent to $\delta_k < f(\boldsymbol{x})$. By this fact and the convexity and consistency of ϕ^Y, the following inequalities hold:

$$\frac{\phi^+(f(\boldsymbol{x})) - \phi^+(\delta_k)}{f(\boldsymbol{x}) - \delta_k} \geq \phi^{+\prime}(\delta_k) \qquad\qquad \frac{\phi^-(-f(\boldsymbol{x})) - \phi^-(-\delta_k)}{-f(\boldsymbol{x}) + \delta_k} \leq \phi^{-\prime}(-\delta_k).$$

Thus,

$$\begin{aligned}
Q_p(f) - Q_p(\delta_k) &= p(\boldsymbol{x})(\phi^+(f(\boldsymbol{x})) - \phi^+(\delta_k)) + (1 - p(\boldsymbol{x}))(\phi^-(-f(\boldsymbol{x})) - \phi^-(-\delta_k)) \\
&\geq p(\boldsymbol{x})(f(\boldsymbol{x}) - \delta_k)\phi^{+\prime}(\delta_k) - (1 - p(\boldsymbol{x}))(f(\boldsymbol{x}) - \delta_k)\phi^{-\prime}(-\delta_k) \\
&\geq (f(\boldsymbol{x}) - \delta_k)\big\{p(\boldsymbol{x})\big(\phi^{+\prime}(\delta_k) + \phi^{-\prime}(-\delta_k)\big) - \phi^{-\prime}(-\delta_k)\big\} \\
&\geq (f(\boldsymbol{x}) - \delta_k)\big\{p(\boldsymbol{x})\tfrac{\phi^{-\prime}(-\delta_k)}{\pi_k} - \phi^{-\prime}(-\delta_k)\big\} \\
&\geq (f(\boldsymbol{x}) - \delta_k)\ \phi^{-\prime}(-\delta_k)\ (\tfrac{p(\boldsymbol{x})}{\pi_k} - 1).
\end{aligned}$$

Since $f(\boldsymbol{x}) - \delta_k > 0$, $\phi^{-\prime}(-\delta_k) < 0$, and $p(\boldsymbol{x}) < \pi_k$, $Q_p(f) - Q_p(\delta_k) \geq 0$. The case when $g(\boldsymbol{x}) < W^*_{\boldsymbol{\pi}}(\boldsymbol{x})$ follows similarly, and the proof is complete.

4.9.6 Proof of Theorem 4.5

Let ϕ^Y be a consistent surrogate loss for appropriately defined boundaries $\boldsymbol{\pi}$ in $(0, 1)$ at $\boldsymbol{\delta}$. Throughout, we use $g = C(f; \boldsymbol{\delta})$ to denote the corresponding rule in $\mathcal{G}$ for some margin function $f \in \mathcal{F}$. From the proof of Theorem 4.4, we have that:

$$\begin{aligned}
\Delta R(g) &= \frac{2}{K} \cdot \mathbb{E}\Big\{ \sum_{\mathcal{K}} |p(\boldsymbol{X}) - \pi_k| \Big\} \\
&= \frac{2}{K} \cdot \mathbb{E}\Big\{ \sum_{k=1}^{K} |p(\boldsymbol{X}) - \pi_k| \cdot \mathbf{I}\{k \in \mathcal{K}\} \Big\} \\
&= \frac{2}{K} \cdot \sum_{k=1}^{K} \mathbb{E}\Big\{ |p(\boldsymbol{X}) - \pi_k| \cdot \mathbf{I}\{k \in \mathcal{K}\} \Big\},
\end{aligned}$$

where $\mathcal{K}$ is defined as in the proof of Theorem 4.4 (Section 4.9.5). Additionally, note that for fixed $k \in \{1, \ldots, K\}$:

$$\begin{aligned}
\mathbb{E}\Big\{|p(\boldsymbol{X}) - \pi_k| \cdot \mathbf{I}\{k \in \mathcal{K}\}\Big\} &\geq t \cdot \mathbb{P}\big\{(k \in \mathcal{K}) \cap |p(\boldsymbol{X}) - \pi_k| > t\big\} \\
&= t \cdot \mathbb{P}\big\{|p(\boldsymbol{X}) - \pi_k| > t\big\} - t \cdot \mathbb{P}\big\{(k \notin \mathcal{K}) \cap |p(\boldsymbol{X}) - \pi_k| > t\big\} \\
&\geq t \cdot (1 - At^{\alpha}) - t \cdot \mathbb{P}\{k \notin \mathcal{K}\} \\
&= t \cdot \big(\mathbb{P}\{k \in \mathcal{K}\} - At^{\alpha}\big).
\end{aligned}$$

Combining the above inequalities, we have:

$$\begin{aligned}
\Delta R(g) &\geq \frac{2t}{K} \cdot \Big(\sum_{k=1}^{K} \mathbb{P}\{k \in \mathcal{K}\} - KAt^{\alpha}\Big) \\
&\geq \frac{2t}{K} \cdot \Big(\mathbb{P}\{f \neq f^*\} - KAt^{\alpha}\Big).
\end{aligned}$$

Letting $t = (\frac{\mathbb{P}\{f \neq f^*\}}{2KA})^{1/\alpha}$ and using β to denote $\alpha/(1+\alpha)$,

$$\begin{aligned}
\Delta R(g) &\geq \frac{2}{K} \cdot \Big(\frac{\mathbb{P}\{f \neq f^*\}}{2KA}\Big)^{1/\alpha} \cdot \Big(\frac{\mathbb{P}\{f \neq f^*\}}{2}\Big) \\
&= \frac{\mathbb{P}\{f \neq f^*\}^{(1+\alpha)/\alpha}}{(2A)^{1/\alpha} K^{(1+\alpha)/\alpha}} \\
\frac{\mathbb{P}\{f \neq f^*\}}{K} &\leq \big((2A)^{1/\alpha} \Delta R(g)\big)^{\beta}.
\end{aligned}$$

Now consider,

$$\begin{aligned}
\Delta R(g) &= \frac{2}{K} \cdot \sum_{k=1}^{K} \mathbb{E}\big(|p(\boldsymbol{X}) - \pi_k| \cdot \mathbf{I}\{k \in \mathcal{K}\}\big) \\
&= \frac{2}{K} \cdot \sum_{k=1}^{K} \mathbb{E}\big(|p(\boldsymbol{X}) - \pi_k| \cdot \mathbf{I}\{k \in \mathcal{K}\} \cdot \mathbf{I}\{|p(\boldsymbol{X}) - \pi_k| > \epsilon\}\big) \\
&\quad + \frac{2}{K} \cdot \sum_{k=1}^{K} \mathbb{E}\big(|p(\boldsymbol{X}) - \pi_k| \cdot \mathbf{I}\{k \in \mathcal{K}\} \cdot \mathbf{I}\{|p(\boldsymbol{X}) - \pi_k| \leq \epsilon\}\big).
\end{aligned}$$

Using the inequality: $|x| \cdot \mathbf{I}\{|x| \geq \epsilon\} \leq |x|^s \cdot \epsilon^{1-s}$ for $s \geq 1$, we have:

$$\begin{aligned}
\Delta R(g) \leq \frac{2}{K} \cdot \sum_{k=1}^{K} \mathbb{E}\big(|p(\boldsymbol{X}) - \pi_k|^s \cdot \epsilon^{1-s} \cdot \mathbf{I}\{k \in \mathcal{K}\}\big) \\
+ \frac{2\epsilon}{K} \cdot \sum_{k=1}^{K} \mathbb{P}\{k \in \mathcal{K}\}.
\end{aligned}$$

From the proof of Thoerem 4.4 (Section 4.9.5), $|p(\boldsymbol{X}) - \pi_k|^s \leq C^s \Delta Q_p(\delta_k) \leq C^s \Delta Q_p(f)$ for $k \in \mathcal{K}$. Therefore,

$$\Delta R(g) \leq 2\epsilon^{1-s} C^s \Delta Q(f) + \frac{2\epsilon}{K} \cdot \mathbb{P}\{f \neq f^*\}.$$

Combining with the previous bound on $\mathbb{P}\{f \neq f^*\}$,

$$\Delta R(g) \leq 2\epsilon^{1-s} C^s \Delta Q(f) + 2\epsilon \cdot \big((2A)^{1/\alpha} \Delta R(g)\big)^{\beta}$$

Further choosing $\epsilon = \Delta R(g)^{1-\beta}$,

$$\begin{aligned}
\Delta R(g) &\leq 2\Delta R(g)^{(1-\beta)(1-s)} C^s \Delta Q(f) + 2(2A)^{1/\alpha} \Delta R(g) \\
(1 - 2(2A)^{1/\alpha}) \Delta R(g)^{s+\beta-s\beta} &\leq 2C^s \Delta Q(f) \\
\Delta R(g) &\leq \Big(\frac{2C^s}{1 - 2(2A)^{1/\alpha}}\Big)^{1/(s+\beta-s\beta)} \cdot \Delta Q(f)^{1/(s+\beta-s\beta)}
\end{aligned}$$

Letting D denote the exponentiated fraction on the right of the inequality,

$$\Delta R(g) \leq D \cdot \Delta Q(f)^{1/(s+\beta-s\beta)}.$$

4.9.7 Proof of Corollary 4.1

Let φ^Y be a minimally consistent piecewise linear surrogate loss for appropriately defined boundaries $\boldsymbol{\pi}$ in $(0, 1)$ at $\boldsymbol{\delta}$. The φ^Y-optimal margin function, denoted by f^*_φ, is given by:

$$\begin{aligned}
f^*_\varphi(\boldsymbol{X}) &= \operatorname*{argmin}_f \mathbb{E}_{Y|\boldsymbol{X}}\{\varphi^Y(Y f(\boldsymbol{X}))\} \\
&= \operatorname*{argmin}_f \big\{p(\boldsymbol{X})\varphi^+(f(\boldsymbol{X})) + (1 - p(\boldsymbol{X}))\varphi^-(-f(\boldsymbol{X}))\big\} \\
&= \begin{cases}
A^-(\pi_1)/B^-(\pi_1) & \text{if } p(\boldsymbol{X}) \in [0, \pi_1) \\
H^+(\pi_1, \pi_2) & \text{if } p(\boldsymbol{X}) \in (\pi_1, \pi_2] \\
\dots & \\
H^+(\pi_{K-1}, \pi_K) & \text{if } p(\boldsymbol{X}) \in (\pi_{K-1}, \pi_K] \\
A^+(\pi_K)/B^+(\pi_K) & \text{if } p(\boldsymbol{X}) \in (\pi_K, 1]
\end{cases}.
\end{aligned}$$

For any $k \in \{1, \ldots, K\}$,

$$\begin{aligned}
\Delta Q_p(\delta_k) &= Q_p(\delta_k) - Q_p(f^*_\varphi(\boldsymbol{X})) \\
&= p(\boldsymbol{X})(\varphi^+(\delta_k) - \varphi^+(f^*_\varphi(\boldsymbol{X}))) + (1 - p(\boldsymbol{X}))(\varphi^-(-\delta_k) - \varphi^-(-f^*_\varphi(\boldsymbol{X}))) \\
&= p(\boldsymbol{X})B^+(\pi_k)(\delta_k - f^*_\varphi(\boldsymbol{X})) - (1 - p(\boldsymbol{X}))B^-(\pi_k)(\delta_k - f^*_\varphi(\boldsymbol{X})) \\
&= p(\boldsymbol{X})(B^+(\pi_k) + B^-(\pi_k))(\delta_k - f^*_\varphi(\boldsymbol{X})) - B^-(\pi_k)(\delta_k - f^*_\varphi(\boldsymbol{X})) \\
&= p(\boldsymbol{X})(B^-(\pi_k) \cdot \pi_k^{-1})(\delta_k - f^*_\varphi(\boldsymbol{X})) - B^-(\pi_k)(\delta_k - f^*_\varphi(\boldsymbol{X})) \\
&= B^-(\pi_k) \cdot \pi_k^{-1} \cdot (p(\boldsymbol{X}) - \pi_k)(\delta_k - f^*_\varphi(\boldsymbol{X})).
\end{aligned}$$

Since $f^*_\varphi(\boldsymbol{X}) > \delta_k$ when $p(\boldsymbol{X}) > \pi_k$, and similarly $f^*_\varphi(\boldsymbol{X}) < \delta_k$ when $p(\boldsymbol{X}) < \pi_k$, $(p(\boldsymbol{X}) - \pi_k)(\delta_k - f^*_\varphi(\boldsymbol{X})) \leq 0$ must always hold. Therefore,

$$\begin{aligned}
\Delta Q_p(\delta_k) &= -\frac{B^-(\pi_k) \cdot |\delta_k - f^*_\varphi(\boldsymbol{X})|}{\pi_k} \cdot |p(\boldsymbol{X}) - \pi_k| \\
&\geq C^{-1} \cdot |p(\boldsymbol{X}) - \pi_k|,
\end{aligned}$$

where $C = \max\left\{-\frac{\pi_k}{B^-(\pi_k)\cdot|\delta_k - H_j|} : k = 1, \ldots, K;\ j = 0, \ldots, K\right\} > 0$. Letting $s = 1$, the desired bound is achieved.

4.9.8 Proof of Theorem 4.6

Let φ^Y be a minimally consistent piecewise linear surrogate loss for appropriately defined boundaries $\boldsymbol{\pi}$ in $(0, 1)$ at $\boldsymbol{\delta}$. We first show that $\mathcal{H} = \{h_f(\boldsymbol{x}, y) = \varphi^y(yf(\boldsymbol{x})) - \varphi^y(yf^*_\varphi(\boldsymbol{x})) : f \in \mathcal{F}\}$ is a Bernstein class of functions, i.e. that there exists some $B > 1$, $\beta \in (0, 1]$ such that:

$$\mathbb{E}\{h_f(\boldsymbol{X}, Y)^2\} \leq B \cdot \mathbb{E}\{h_f(\boldsymbol{X}, Y)\}^\beta.$$

Then, given that h_f is a Bernstein class, we complete the proof by obtaining a tail bound on $\mathbb{E}h_f(\boldsymbol{X}, Y) - 2\frac{1}{n}\sum_i h_f(\boldsymbol{x}_i, y_i)$. Following the approach of [11], to derive the Bernstein property of h_f, we first show that $\Delta Q_p(f)$ can be bounded below by a pseudo-norm between f and f^*_φ, denoted $\rho_{\boldsymbol{X}}(f, f^*_\varphi)$. Then, we show that $\mathbb{E}\{h_f(\boldsymbol{X}, Y)^2\}$ can bounded above by $\mathbb{E}\{\rho_{\boldsymbol{X}}(f, f^*_\varphi)\}$, and combine

the two results to show the Bernstein property of h_f. Let $\rho_{\boldsymbol{X}}(f, f^*_\varphi)$ be defined as:

$$\rho_{\boldsymbol{X}}(f, f^*_\varphi) = \begin{cases} p(\boldsymbol{X})|f - f^*_\varphi| & \text{if } p(\boldsymbol{X}) < \pi_1,\ f < H_0 \\ (1 - p(\boldsymbol{X}))|f - f^*_\varphi| & \text{if } p(\boldsymbol{X}) > \pi_K,\ f > H_K \\ |f - f^*_\varphi| & \text{otherwise} \end{cases}.$$

Lemma 4.1. *For $p(\boldsymbol{X}) \in [0, 1]$,*

$$\Delta Q_p(f) \geq D^* \cdot \min\{|p(\boldsymbol{X}) - \pi_1|, |p(\boldsymbol{X}) - \pi_K|, (1 - \pi_1), \pi_K\} \cdot \rho_{\boldsymbol{X}}(f, f^*_\varphi).$$

Proof. Since $Q_p(f)$ is convex, $Q_p(f) \geq Q_p(f^*_\varphi) + r \cdot (f - f^*_\varphi)$ for any subgradient, r, of $Q_p(\cdot)$ at f^*_φ. Since φ^Y is piecewise linear, and f^*_φ is as defined above, the set of subgradients are given by:

$$r = \begin{cases} p(\boldsymbol{X})B^+(\pi_1) & \\ \quad \text{and} \quad p(\boldsymbol{X})B^+(\pi_1) + (1 - p(\boldsymbol{X}))B^-(\pi_1) & \text{for } f^*_\varphi = H_0 \\ p(\boldsymbol{X})B^+(\pi_1) + (1 - p(\boldsymbol{X}))B^-(\pi_1) & \\ \quad \text{and} \quad p(\boldsymbol{X})B^+(\pi_1) + (1 - p(\boldsymbol{X}))B^-(\pi_1) & \text{for } f^*_\varphi = H_1, \ldots, H_{K-1} \\ (1 - p(\boldsymbol{X}))B^-(\pi_K) & \\ \quad \text{and} \quad p(\boldsymbol{X})B^+(\pi_K) + (1 - p(\boldsymbol{X}))B^-(\pi_K) & \text{for } f^*_\varphi = H_K \end{cases}.$$

Therefore,

$$\begin{aligned} Q_p(f) &\geq Q_p(f^*_\varphi) + r \cdot (f - f^*_\varphi) \\ \Delta Q_p(f) &\geq r \cdot (f - f^*_\varphi) \\ &\geq \begin{cases} \big(p(\boldsymbol{X})B^+(\pi_1)\big) \cdot \big(f - f^*_\varphi\big) & \text{if } p(\boldsymbol{X}) < \pi_1,\ f < H_0 \\ \big(p(\boldsymbol{X})B^+(\pi_1) - (1 - p(\boldsymbol{X}))B^-(\pi_1)\big) \cdot \big(f - f^*_\varphi\big) & \text{if } p(\boldsymbol{X}) < \pi_1,\ f > H_0 \\ \big(p(\boldsymbol{X})B^+(\pi_k) - (1 - p(\boldsymbol{X}))B^-(\pi_k)\big) \cdot \big(f - f^*_\varphi\big) & \text{if } p(\boldsymbol{X}) \in [\pi_k, \pi_{k+1}),\ f < H_k \\ \big(p(\boldsymbol{X})B^+(\pi_{k+1}) - (1 - p(\boldsymbol{X}))B^-(\pi_{k+1})\big) \cdot \big(f - f^*_\varphi\big) & \text{if } p(\boldsymbol{X}) \in [\pi_k, \pi_{k+1}),\ f > H_k \\ \big(p(\boldsymbol{X})B^+(\pi_K) - (1 - p(\boldsymbol{X}))B^-(\pi_K)\big) \cdot \big(f - f^*_\varphi\big) & \text{if } p(\boldsymbol{X}) > \pi_K,\ f < H_K \\ (1 - p(\boldsymbol{X}))B^-(\pi_K) \cdot \big(f - f^*_\varphi\big) & \text{if } p(\boldsymbol{X}) > \pi_K,\ f > H_K \end{cases}. \end{aligned}$$

Since by definition, $B^+(\pi_1) \leq B^+(\pi_2) \leq \cdots \leq B^+(\pi_K)$ and $B^-(\pi_1) \geq B^-(\pi_2) \geq \cdots \geq B^-(\pi_K)$, we have:

$$-p(\boldsymbol{X})B^+(\pi_k) + (1-p(\boldsymbol{X}))B^-(\pi_k) \geq -p(\boldsymbol{X})B^+(\pi_K) + (1-p(\boldsymbol{X}))B^-(\pi_K)$$
$$p(\boldsymbol{X})B^+(\pi_{k+1}) - (1-p(\boldsymbol{X}))B^-(\pi_{k+1}) \leq p(\boldsymbol{X})B^+(\pi_1) - (1-p(\boldsymbol{X}))B^-(\pi_1).$$

Therefore, the bound on $\Delta Q_p(f)$ may be rewritten as:

$$\Delta Q_p(f) \geq \begin{cases} \left|p(\boldsymbol{X})B^+(\pi_1)\right| \cdot \left|f - f^*_\varphi\right| & \text{if } p(\boldsymbol{X}) < \pi_1,\ f < H_0 \\ \left|p(\boldsymbol{X})B^+(\pi_1) - (1-p(\boldsymbol{X}))B^-(\pi_1)\right| \cdot \left|f - f^*_\varphi\right| & \text{if } p(\boldsymbol{X}) < \pi_1,\ f > H_0 \\ \left|p(\boldsymbol{X})B^+(\pi_K) - (1-p(\boldsymbol{X}))B^-(\pi_K)\right| \cdot \left|f - f^*_\varphi\right| & \text{if } p(\boldsymbol{X}) > \pi_K,\ f < H_K \\ \left|(1-p(\boldsymbol{X}))B^-(\pi_K)\right| \cdot \left|f - f^*_\varphi\right| & \text{if } p(\boldsymbol{X}) > \pi_K,\ f > H_K \\ \min\left\{\left|p(\boldsymbol{X})B^+(\pi_1) - (1-p(\boldsymbol{X}))B^-(\pi_1)\right|,\right. & \\ \quad \left.\left|p(\boldsymbol{X})B^+(\pi_K) - (1-p(\boldsymbol{X}))B^-(\pi_K)\right|\right\} \cdot \left|f - f^*_\varphi\right| & \text{otherwise} \end{cases}.$$

By the consistency of φ^Y, $B^-(\pi_k)/(B^+(\pi_k) + B^-(\pi_k)) = \pi_k$ for all k. Thus, letting $D^* = \min_{k=1,\ldots,K}\{|B^+(\pi_k) + B^-(\pi_k)|\} > 0$, $p(\boldsymbol{X})B^+(\pi_k) - (1-p(\boldsymbol{X}))B^-(\pi_k) = (p(\boldsymbol{X}) - \pi_k)(B^+(\pi_k) + B^-(\pi_k)) \geq D^* \cdot |p(\boldsymbol{X}) - \pi_k|$. Therefore,

$$\Delta Q_p(f) \geq \begin{cases} \left|B^+(\pi_1)\right| \cdot \rho_{\boldsymbol{X}}(f, f^*_\varphi) & \text{if } p(\boldsymbol{X}) < \pi_1,\ f < H_0 \\ D^* \cdot |p(\boldsymbol{X}) - \pi_1| \cdot \rho_{\boldsymbol{X}}(f, f^*_\varphi) & \text{if } p(\boldsymbol{X}) < \pi_1,\ f > H_0 \\ D^* \cdot |p(\boldsymbol{X}) - \pi_K| \cdot \rho_{\boldsymbol{X}}(f, f^*_\varphi) & \text{if } p(\boldsymbol{X}) > \pi_K,\ f < H_K \\ \left|B^-(\pi_K)\right| \cdot \rho_{\boldsymbol{X}}(f, f^*_\varphi) & \text{if } p(\boldsymbol{X}) > \pi_K,\ f > H_K \\ D^* \cdot \min\left\{|p(\boldsymbol{X}) - \pi_1|, |p(\boldsymbol{X}) - \pi_K|\right\} \cdot \rho_{\boldsymbol{X}}(f, f^*_\varphi) & \text{otherwise} \end{cases}.$$

Since $|B^+(\pi_1)| \geq D^* \cdot (1-\pi_1)$, $|B^-(\pi_K)| \geq D^* \cdot \pi_K$, we have for $p(\boldsymbol{X}) \in [0,1]$:

$$\Delta Q_p(f) \geq D^* \cdot \min\left\{|p(\boldsymbol{X}) - \pi_1|, |p(\boldsymbol{X}) - \pi_K|, (1-\pi_1), \pi_K\right\} \cdot \rho_{\boldsymbol{X}}(f, f^*_\varphi).$$

□

Lemma 4.2. *If $|f| < B$ for all $f \in \mathcal{F}$,*

$$\mathbb{E}_{Y|\boldsymbol{X}}\{h_f(\boldsymbol{X}, Y)^2\} \leq L^2(B+M) \cdot \rho_{\boldsymbol{X}}(f, f^*_\varphi)$$

for $L, M \geq 0$.

Proof. We first decompose the conditional expectation as:

$$\begin{aligned}\mathbb{E}_{Y|\boldsymbol{X}}\{h_f(\boldsymbol{X}, Y)^2\} &= \mathbb{E}_{Y|\boldsymbol{X}}\left\{\big(\varphi^Y(Yf(\boldsymbol{X})) - \varphi^Y(Yf(\boldsymbol{X}))\big)^2\right\} \\ &= p(\boldsymbol{X})\big(\varphi^+(f(\boldsymbol{X})) - \varphi^+(f^*_\varphi(\boldsymbol{X}))\big)^2 \\ &\quad + (1-p(\boldsymbol{X}))\big(\varphi^-(f(\boldsymbol{X})) - \varphi^-(f^*_\varphi(\boldsymbol{X}))\big)^2.\end{aligned}$$

Note that if $f(\boldsymbol{X}) \leq H_0$ and $p(\boldsymbol{X}) \leq \pi_1$, then $\varphi^-(f(\boldsymbol{X})) = 0$ and $\varphi^-(f^*_\varphi(\boldsymbol{X})) = 0$. Similarly, if $f(\boldsymbol{X}) \geq H_K$ and $p(\boldsymbol{X}) \geq \pi_K$, then $\varphi^+(f(\boldsymbol{X})) = 0$ and $\varphi^+(f^*_\varphi(\boldsymbol{X})) = 0$. Therefore,

$$E_{Y|\boldsymbol{X}}\{h_f(\boldsymbol{X}, Y)^2\} = \begin{cases} p(\boldsymbol{X})\big(\varphi^+(f(\boldsymbol{X})) - \varphi^+(f^*_\varphi(\boldsymbol{X}))\big)^2 & \text{if } f(\boldsymbol{X}) \leq H_0,\ p(\boldsymbol{X}) \leq \pi_1 \\ (1-p(\boldsymbol{X}))\big(\varphi^-(f(\boldsymbol{X})) - \varphi^-(f^*_\varphi(\boldsymbol{X}))\big)^2 & \text{if } f(\boldsymbol{X}) \geq H_K,\ p(\boldsymbol{X}) \geq \pi_K \\ p(\boldsymbol{X})\big(\varphi^+(f(\boldsymbol{X})) - \varphi^+(f^*_\varphi(\boldsymbol{X}))\big)^2 & \\ \quad + (1-p(\boldsymbol{X}))\big(\varphi^-(f(\boldsymbol{X})) - \varphi^-(f^*_\varphi(\boldsymbol{X}))\big)^2 & \text{otherwise} \end{cases}.$$

Let $L = \max\{B^+(\pi_1), B^-(\pi_K)\}$ denote the Lipschitz constant for φ^Y, and let $M = \max\{|H_0|, |H_K|\}$ denote the bound on f^*_φ, such that $|f^*_\varphi(\boldsymbol{X})| \leq M$ for all $\boldsymbol{X}$. Then,

$$E_{Y|\boldsymbol{X}}\{h_f(\boldsymbol{X}, Y)^2\} \leq L^2(B+M) \cdot \rho_{\boldsymbol{X}}(f, f^*_\varphi),$$

where $\rho_{\boldsymbol{X}}$ is as defined above. □

Lemma 4.3. *If $p(\boldsymbol{X})$ satisfies the margin condition (4.14) at $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$ with parameters A, α, then for any class $\mathcal{F}$ of measurable uniformly bounded functions, the class $\mathcal{H} = \{h_f(\boldsymbol{X}, Y) : f \in \mathcal{F}\}$ is a Bernstein class with exponent $\beta = \alpha/(1+\alpha)$.*

Proof. Let E_1 denote the event that $|p(\boldsymbol{X}) - \pi|$ is the minimizer over the set $\big\{|p(\boldsymbol{X}) - \pi_1|, |p(\boldsymbol{X}) - \pi_K|, (1-\pi_1), \pi_K\big\}$, and let E_2, E_3, E_4 similarly denote the corresponding events for $|p(\boldsymbol{X}) - \pi_K|$, $(1-\pi_1)$ and π_K. Using $\mathbf{I}_E$ to denote the indicator for event E, by Lemma 4.2 we have:

$$\mathbb{E}\{h_f(\boldsymbol{X}, Y)\} \geq D^* \cdot \mathbb{E}\big\{\min\{|p(\boldsymbol{X}) - \pi_1|, |p(\boldsymbol{X}) - \pi_K|, (1-\pi_1), \pi_K\} \cdot \rho_{\boldsymbol{X}}(f, f^*_\varphi)\big\}$$

$$= D^* \cdot \mathbb{E}\big\{\rho_{\boldsymbol{X}}(f, f^*) \cdot \{\mathbf{I}_{E_1} \cdot |p(\boldsymbol{X}) - \pi_1| + \mathbf{I}_{E_2} \cdot |p(\boldsymbol{X}) - \pi_K| + \mathbf{I}_{E_3} \cdot (1 - \pi_1) + \mathbf{I}_{E_4} \cdot \pi_K\}\big\}.$$

Let $t_{\max} = \min_{k=1,\ldots,K+1}\{\pi_k - \pi_{k-1}\}$, where $\pi_0 = 0, \pi_{K+1} = 1$. Given the margin condition, for all k, there exists some $A \geq 0, \alpha \geq 0$ such that for all $t \in [0, t_{\max})$,

$$\mathbb{P}\{|p(\boldsymbol{X}) - \pi_k| \leq t\} \leq At^\alpha,$$

for $k = 1, \ldots, K$. Therefore, letting B and M denote the bounds on f and f_φ^* given in the proof of Lemma 4.2,

$$\begin{aligned}\mathbb{E}\big\{\rho_{\boldsymbol{X}}(f, f_\varphi^*) \cdot |p(\boldsymbol{X}) - \pi_1| \cdot \mathbf{I}_{E_1}\big\} &\geq t \cdot \mathbb{E}\big\{\rho_{\boldsymbol{X}}(f, f_\varphi^*) \cdot \mathbf{I}\{|p(\boldsymbol{X}) - \pi_1| > t\} \cdot \mathbf{I}_{E_1}\big\} \\ &\geq t \cdot \big[\mathbb{E}\big\{\rho_{\boldsymbol{X}}(f, f_\varphi^*) \cdot \mathbf{I}_{E_1}\big\} - (B + M) \cdot At^\alpha\big],\end{aligned}$$

and similarly,

$$\begin{aligned}\mathbb{E}\big\{\rho_{\boldsymbol{X}}(f, f_\varphi^*) \cdot |p(\boldsymbol{X}) - \pi_K| \cdot \mathbf{I}_{E_2}\big\} &\geq t \cdot \big[\mathbb{E}\big\{\rho_{\boldsymbol{X}}(f, f_\varphi^*) \cdot \mathbf{I}_{E_2}\big\} - (B + M) \cdot At^\alpha\big] \\ \mathbb{E}\big\{\rho_{\boldsymbol{X}}(f, f_\varphi^*) \cdot (1 - \pi_1) \cdot \mathbf{I}_{E_3}\big\} &\geq t \cdot \big[\mathbb{E}\big\{\rho_{\boldsymbol{X}}(f, f_\varphi^*) \cdot \mathbf{I}_{E_3}\big\} - (B + M) \cdot \mathbf{I}\{(1 - \pi_1) < t, (1 - \pi_1) \leq \pi_K\}\big] \\ \mathbb{E}\big\{\rho_{\boldsymbol{X}}(f, f_\varphi^*) \cdot \pi_K \cdot \mathbf{I}_{E_4}\big\} &\geq t \cdot \big[\mathbb{E}\big\{\rho_{\boldsymbol{X}}(f, f_\varphi^*) \cdot \mathbf{I}_{E_4}\big\} - (B + M) \cdot \mathbf{I}\{\pi_K < t, \pi_K < (1 - \pi_1)\}\big].\end{aligned}$$

Assume without loss of generality that $\pi_K < (1 - \pi_1)$. Let

$$t = \left(\frac{\mathbb{E}\big\{\rho_{\boldsymbol{X}}(f, f_\varphi^*)\big\}}{C \cdot 2A(B + M)}\right)^{1/\alpha},$$

where $C \geq \max\{2, (2A\pi_K^\alpha)^{-1}\}$. Then, since $\mathbb{E}\{\rho_{\boldsymbol{X}}(f, f_\varphi^*)\} \leq (B + M)$, we have $t < \pi_K$. Combining the above inequalities, we have:

$$\begin{aligned}\mathbb{E}\{h_f(\boldsymbol{X}, Y)\} &\geq D^* \cdot t \cdot \big[\mathbb{E}\big\{\rho_{\boldsymbol{X}}(f, f_\varphi^*)\big\} - (B + M)(2At^\alpha)\big] \\ &\geq D^* \cdot \left(\frac{\mathbb{E}\big\{\rho_{\boldsymbol{X}}(f, f_\varphi^*)\big\}}{C \cdot 2A(B + M)}\right)^{1/\alpha} \big[\mathbb{E}\{\rho_{\boldsymbol{X}}(f, f_\varphi^*)\} - C^{-1}\mathbb{E}\big\{\rho_{\boldsymbol{X}}(f, f_\varphi^*)\big\}\big] \\ &\geq D^* \cdot \left(\frac{C - 1}{C}\right) \cdot \left(\frac{1}{C \cdot 2A(B + M)}\right)^{1/\alpha} \cdot \mathbb{E}\big\{\rho_{\boldsymbol{X}}(f, f_\varphi^*)\big\}^{(1+\alpha)/\alpha} \\ \mathbb{E}\big\{\rho_{\boldsymbol{X}}(f, f_\varphi^*)\big\} &\leq \left[\left(\frac{C}{C - 1}\right) \cdot D^* \cdot (C \cdot 2A(B + M))^{1/\alpha}\right]^{\alpha/(1+\alpha)} \cdot \mathbb{E}\{h_f(\boldsymbol{X}, Y)\}^{\alpha/(1+\alpha)}.\end{aligned}$$

Combining with the result of Lemma 4.2, and noting that $\mathbb{E}\{\rho_{\boldsymbol{X}}(f, f^*_\varphi)\} = \mathbb{E}_{\boldsymbol{X}}\{\rho_{\boldsymbol{X}}(f, f^*_\varphi)\}$, we have:

$$\begin{aligned}
\mathbb{E}\big\{h_f(\boldsymbol{X}, Y)^2\big\} &= \mathbb{E}_{\boldsymbol{X}}\big\{\mathbb{E}_{Y|\boldsymbol{X}}\{h_f(\boldsymbol{X}, Y)^2\} \\
&\leq L^2(B+M) \cdot \mathbb{E}_{\boldsymbol{X}}\big\{\rho_{\boldsymbol{X}}(f, f^*_\varphi)\big\} \\
&\leq L^2(B+M) \cdot \left[\left(\frac{C}{C-1}\right) \cdot D^* \cdot (C \cdot 2A(B+M))^{1/\alpha}\right]^{\alpha/(1+\alpha)} \cdot \mathbb{E}\{h_f(\boldsymbol{X}, Y)\}^{\alpha/(1+\alpha)},
\end{aligned}$$

such that h_f is a Bernstein class. □

Let B' and β be defined such that $\mathbb{E}\big\{h_f(\boldsymbol{X}, Y)^2\big\} \leq B' \cdot \mathbb{E}\{h_f(\boldsymbol{X}, Y)\}^\beta$. Let $\hat{f}_n$ denote the empirical minimizer in $\mathcal{F}$ of $\varphi^y(yf(\boldsymbol{x}))$ over a training sample of size n. We first bound the excess φ-risk by:

$$\begin{aligned}
\Delta Q(\hat{f}_n) &= \mathbb{E}\{h_{\hat{f}_n}(\boldsymbol{X}, Y)\} \\
&= 2\Big(\frac{1}{n}\sum_{i=1}^n h_{\widehat{f}_n}(\boldsymbol{x}_i, y_i)\Big) + \Big(\mathbb{E}\{h_{\widehat{f}_n}(\boldsymbol{X}, Y)\} - 2\big(\frac{1}{n}\sum_{i=1}^n h_{\widehat{f}_n}(\boldsymbol{x}_i, y_i)\big)\Big) \\
&\leq \sup_{f \in \mathcal{F}_B}\Big(\mathbb{E}\{h_f(\boldsymbol{X}, Y)\} - 2\big(\frac{1}{n}\sum_{i=1}^n h_f(\boldsymbol{x}_i, y_i)\big)\Big).
\end{aligned}$$

Note that,

$$\sup_{f \in \mathcal{F}_B}\Big(\mathbb{E}\{h_f(\boldsymbol{X}, Y)\} - 2\big(\frac{1}{n}\sum_{i=1}^n h_f(\boldsymbol{x}_i, y_i)\big)\Big) \leq \frac{3L}{n} + \sup_{f \in \mathcal{F}_n}\Big(\mathbb{E}\{h_f(\boldsymbol{X}, Y)\} - 2\big(\frac{1}{n}\sum_{i=1}^n h_f(\boldsymbol{x}_i, y_i)\big)\Big),$$

where $\mathcal{F}_n$ is a minimal $1/n$-net of $\mathcal{F}_B$. Now applying Bernstein's inequality,

$$\begin{aligned}
&\mathbb{P}\Big\{\sup_{f \in \mathcal{F}_n}\Big(\mathbb{E}\{h_f(\boldsymbol{X}, Y)\} - 2\big(\frac{1}{n}\sum_{i=1}^n h_f(\boldsymbol{x}_i, y_i)\big)\Big) \geq t\Big\} \\
&\qquad \leq N_n \cdot \exp\Big\{-\frac{n(t + \mathbb{E}\{h_f(\boldsymbol{X}, Y)\})^2/8}{\mathbb{E}\{h_f(\boldsymbol{X}, Y)^2\} + (2LB)(t + \mathbb{E}\{h_f(\boldsymbol{X}, Y)\})/6}\Big\}.
\end{aligned}$$

Using the fact that h_f is a Bernstein class, and noting that for $\beta \in [0, 1)$, $z^\beta \leq 1 + z$ for all $z > 0$,

$$\begin{aligned}
\frac{\mathbb{E}\{h_f(\boldsymbol{X}, Y)^2\}}{t + \mathbb{E}\{h_f(\boldsymbol{X}, Y)\}} &\leq B' \cdot \frac{\mathbb{E}\{h_f(\boldsymbol{X}, Y)\}^\beta}{t + \mathbb{E}\{h_f(\boldsymbol{X}, Y)\}}. \\
&\leq B' \cdot \frac{1 + \mathbb{E}\{h_f(\boldsymbol{X}, Y)\}}{t + \mathbb{E}\{h_f(\boldsymbol{X}, Y)\}} \\
&\leq B' \cdot t^{-1}.
\end{aligned}$$

Therefore,

$$\mathbb{P}\Big\{ \sup_{f\in\mathcal{F}_n} \Big(\mathbb{E}\{h_f(\boldsymbol{X},Y)\} - 2\big(\frac{1}{n}\sum_{i=1}^{n} h_f(\boldsymbol{x}_i,y_i)\big)\Big) \geq t\Big\}$$
$$\leq N_n \cdot \exp\Big\{ -\frac{n}{8}\cdot\frac{(t+\mathbb{E}\{h_f(\boldsymbol{X},Y)\})}{B'+\frac{LB}{3}}\Big\}$$
$$\leq N_n \cdot \exp\Big\{ -\frac{nt}{8}\cdot\Big(\frac{B'}{t}+\frac{LB}{3}\Big)^{-1}\Big\}.$$

The proof is complete by noting that the necessary bound holds with probability γ for:

$$t = 4\cdot\frac{LB}{3}\cdot\frac{\log(N_n/\gamma)}{n} + \left(\left(4\cdot\frac{LB}{3}\cdot\frac{\log(N_n/\gamma)}{n}\right)^2 + 8\cdot B'\cdot\frac{\log(N_n/\gamma)}{n}\right)^{1/2}.$$

BIBLIOGRAPHY

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., Powell, J. I., Yang, L., Marti, G. E., Moore, T., Hudson, Jr., J., Lu, L., Lewis, D. B., Tibshirani, R., Sherlock, G., Chan, W. C., Greiner, T. C., Weisenburger, D. D., Armitage, J. O., Warnke, R., Levy, R., Wilson, W., Grever, M. R., Byrd, J. C., Botstein, D., Brown, P. O., and Staudt, L. M. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511.

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology*, 11(10):R106.

Anders, S., Reyes, A., and Huber, W. (2012). Detecting differential usage of exons from RNA-seq data. *Genome Research*, 22(10):2008–2017.

Archangelo, L. F., Gläsner, J., Krause, A., and Bohlander, S. K. (2006). The novel CALM interactor CATS influences the subcellular localization of the leukemogenic fusion protein CALM/AF10. *Oncogene*, 25(29):4099–4109.

Archangelo, L. F., Greif, P. A., Hölzel, M., Harasim, T., Kremmer, E., Przemeck, G. K. H., Eick, D., Deshpande, A. J., Buske, C., de Angelis, M. H., Saad, S. T. O., and Bohlander, S. K. (2008). The CALM and CALM/AF10 interactor CATS is a marker for proliferation. *Molecular Oncology*, 2(4):356–367.

Aronszajn, N. (1950). Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, 68(3):337–404.

Baik, J. and Silverstein, J. W. (2006). Eigenvalues of large sample covariance matrices of spiked population models. *Journal of Multivariate Analysis*, 97(6):1382–1408.

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156.

Bartlett, P. L. and Wegkamp, M. H. (2008). Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9:1823–1840.

Bastien, R. R. L., Rodríguez-Lescure, A., Ebbert, M. T. W., Prat, A., Munárriz, B., Rowe, L., Miller, P., Ruiz-Borrego, M., Anderson, D., Lyons, B., Álvarez, I., Dowell, T., Wall, D., Seguí, M. A., Barley, L., Boucher, K. M., Alba, E., Pappas, L., Davis, C. A., Aranda, I., Fauron, C., Stijleman, I. J., Palacios, J., Antón, A., Carrasco, E., Caballero, R., Ellis, M. J., Nielsen, T. O., Perou, C. M., Astill, M., Bernard, P. S., and Martín, M. (2012). PAM50 breast cancer subtyping by RT-qPCR and concordance with standard clinical molecular markers. *BMC Medical Genomics*, 5:44.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57(1):289–300.

Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E. J., Lander, E. S., Wong, W., Johnson, B. E., Golub, T., Sugarbaker, D. J., and Meyerson, M. (2001). Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences of the United States of America*, 98(24):13790–5.

Blanchard, G., Bousquet, O., and Massart, P. (2008). Statistical performance of support vector machines. *The Annals of Statistics*, 36(2):489–531.

Blekhman, R., Marioni, J. C., Zumbo, P., Stephens, M., and Gilad, Y. (2010). Sex-specific and lineage-specific alternative splicing in primates. *Genome Research*, 20:180–189.

Borysov, P., Hannig, J., and Marron, J. S. (2014). Asymptotics of hierarchical clustering for growing dimension. *Journal of Multivariate Analysis*, 124:465–479.

Bottou, L. and Lin, C.-J. (2007). Support vector machine solvers. In *Large Scale Kernel Machines*, pages 301–320. MIT Press, Cambridge, MA.

Bullard, J. H., Purdom, E., Hansen, K. D., and Dudoit, S. (2010). Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics*, 11(94).

Cabanski, C. R., Wilkerson, M. D., Soloway, M., Parker, J. S., Liu, J., Prins, J. F., Marron, J. S., Perou, C. M., and Hayes, D. N. (2013). BlackOPs: increasing confidence in variant detection through mappability filtering. *Nucleic Acids Research*, 41(19):e178.

Calamai, P. H. and Moré, J. J. (1987). Projected gradient methods for linearly constrained problems. *Mathematical Programming*, 39(1):93–116.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273–297.

Coulombe-Huntington, J., Lam, K. C. L., Dias, C., and Majewski, J. (2009). Fine-scale variation and genetic determinants of alternative splicing across individuals. *PLoS Genetics*, 5(12):e1000766.

Cover, T. M. and Hart, P. E. (1967). Nearest Neighbor pattern classification. *IEEE Transactions on Information Theory*, IT-13(1):21–27.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 1 edition.

Dong, Y., Bui, L. T., Odorico, D. M., Tan, O. L., Myers, S. A., Samaratunga, H., Gardiner, R. A., and Clements, J. A. (2005). Compartmentalized expression of kallikrein 4 (KLK4/hK4) isoforms in prostate cancer: nuclear, cytoplasmic and secreted forms. *Endocrine-Related Cancer*, 12:875–889.

Dong, Y., Kaushal, A., and Brattsand, M. (2003). Differential splicing of KLK5 and KLK7 in epithelial ovarian cancer produces novel variants with potential as cancer biomarkers. *Clinical Cancer Research*, 9:1710–1720.

Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification*. Wiley-Interscience, 2 edition.

Efron, B., Halloran, E., and Holmes, S. (1996). Bootstrap confidence levels for phylogenetic trees. *Proceedings of the National Academy of Sciences of the United States of America*, 93(23):13429–13434.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 95:14863–14868.

Faustino, N. A. and Cooper, T. A. (2003). Pre-mRNA splicing and human disease. *Genes & Development*, 17(4):419–437.

Feng, H., Qin, Z., and Zhang, X. (2012). Opportunities and methods for studying alternative splicing in cancer with RNA-Seq. *Cancer Letters*.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, pages 179–188.

Fix, E. and Hodges, J. L. (1989). Discriminatory analysis-nonparametric discrimination: consistency properties. *International Statistical Review*, 57(3):238–247.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Friedman, J. H., Hastie, T., and Tibshirani, R. (2000). Additive Logistic Regression: a statistical view of boosting. *The Annals of Statistics*, 28(2):337–407.

Grandvalet, Y., Rakotomamonjy, A., Keshet, J., and Canu, S. (2009). Support vector machines with a reject option. In *Advances in Neural Information Processing Systems 21*, pages 537–544. Curran Associates, Inc.

Hall, P., Marron, J. S., and Neeman, A. (2005). Geometric representation of high dimension, low sample size data. *Journal of the Royal Statistical Society: Series B*, 67(3):427–444.

Hartigan, J. A. (1975). *Clustering algorithms*. John Wiley & Sons, Inc.

Hastie, T., Tibshirani, R., and Friedman, J. (2011). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd ed. 2009. corr. 7th printing 2013 edition.

Hayes, D. N., Monti, S., Parmigiani, G., Gilks, C. B., Naoki, K., Bhattacharjee, A., Socinski, M. A., Perou, C. M., and Meyerson, M. (2006). Gene expression profiling reveals reproducible human lung adenocarcinoma subtypes in multiple independent patient cohorts. *Journal of Clinical Oncology*, 24(31):5079–5090.

Herbei, R. and Wegkamp, M. H. (2006). Classification with reject option. *Canadian Journal of Statistics*, 34(4):709–721.

Hoadley, K. A., Yau, C., Wolf, D. M., Cherniack, A. D., Tamborero, D., Ng, S., Leiserson, M. D., Niu, B., McLellan, M. D., Uzunangelov, V., Zhang, J., Kandoth, C., Akbani, R., Shen, H., Omberg, L., Chu, A., Margolin, A. A., van' t Veer, L. J., Lopez-Bigas, N., Laird, P. W., Raphael, B. J., Ding, L., Robertson, A. G., Byers, L. A., Mills, G. B., Weinstein, J. N., Van Waes, C., Chen, Z., Collisson, E. A., Benz, C. C., Perou, C. M., and Stuart, J. M. (2014). Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. *Cell*, 158(4):929–944.

Hu, Y., Huang, Y., Du, Y., Orellana, C. F., Singh, D., Johnson, A. R., Monroy, A., Kuan, P.-F., Hammond, S. M., Makowski, L., Randell, S. H., Chiang, D. Y., Hayes, D. N., Jones, C., Liu, Y., Prins, J. F., and Liu, J. (2013). DiffSplice: the genome-wide detection of differential splicing events with RNA-seq. *Nucleic Acids Research*, 41(2):e39.

Huang, H., Liu, Y., Yuan, M., and Marron, J. S. (2014). Statistical significance of clustering using soft thresholding. *Journal of Computational and Graphical Statistics*, preprint.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer, New York, NY.

Kaufman, L. and Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*, volume 344. Wiley, New York.

Kim, D., Pertea, G., Trapnell, C., Pimentel, H., Kelley, R., and Salzberg, S. L. (2013). TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biology*, 14(4):R36.

Klambauer, G., Unterthiner, T., and Hochreiter, S. (2013). DEXUS: identifying differential expression in RNA-Seq studies with unknown conditions. *Nucleic Acids Research*, 41(21):e198.

Le Cessie, S. and Van Houwelingen, J. C. (1992). Ridge Estimators in Logistic Regression. *Journal of the Royal Statistical Society: Series C*, 41(1):191–201.

Lee, A. H. and Silvapulle, M. J. (1988). Ridge estimation in logistic regression. *Communications in Statistics - Simulation and Computation*, 17(4):1231–1257.

Li, B. and Dewey, C. N. (2011). RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*, 12:323.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., and 1000 Genome Project Data Processing Subgroup (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–2079.

Li, J., Witten, D. M., Johnstone, I. M., and Tibshirani, R. (2012). Normalization, testing, and false discovery rate estimation for RNA-sequencing data. *Biostatistics*, 13(3):523–538.

Lin, Y. (2002). Support vector machines and the Bayes rule in classification. *Data Mining and Knowledge Discovery*, 6:259–275.

Lin, Y. (2004). A note on margin-based loss functions in classification. *Statistics & Probability Letters*, 68(1):73–82.

Lin, Y., Lee, Y., and Wahba, G. (2002). Support vector machines for classification in nonstandard situations. *Machine Learning*, 46:191–202.

Liu, Y., Hayes, D. N., Nobel, A. B., and Marron, J. S. (2008). Statistical significance of clustering for high-dimension, low-sample size data. *Journal of the American Statistical Association*, 103(483):1281–1293.

Liu, Y. and Shen, X. (2006). Multicategory ψ-Learning. *Journal of the American Statistical Association*, 101(474):500–509.

Liu, Y., Zhang, H. H., and Wu, Y. (2011). Hard or Soft Classification? Large-margin Unified Machines. *Journal of the American Statistical Association*, 106(493):166–177.

Mackay, A., Weigelt, B., Grigoriadis, A., Kreike, B., Natrajan, R., A'Hern, R., Tan, D. S., Dowsett, M., Ashworth, A., and Reis-Filho, J. S. (2011). Microarray-based class discovery for molecular classification of breast cancer: analysis of interobserver agreement. *Journal of the National Cancer Institute*, 103(8):662–673.

Macnaughton-Smith, P., Williams, W. T., Dale, M. B., and Mockett, L. G. (1964). Dissimilarity Analysis: a new technique of hierarchical sub-division. *Nature*, 202(4936):1034–1035.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, number 281-297. University of California Press.

Maher, C. A., Kumar-Sinha, C., Cao, X., Kalyana-Sundaram, S., Han, B., Jing, X., Sam, L., Barrette, T., Palanisamy, N., and Chinnaiyan, A. M. (2009). Transcriptome sequencing to detect gene fusions in cancer. *Nature*, 458(7234):97–101.

Mai, Q., Zou, H., and Yuan, M. (2012). A direct approach to sparse discriminant analysis in ultra-high dimensions. *Biometrika*, 99(1):29–42.

Maitra, R., Melnykov, V., and Lahiri, S. N. (2012). Bootstrapping for significance of compact clusters in multidimensional datasets. *Journal of the American Statistical Association*, 107(497):378–392.

Mammen, E. and Tsybakov, A. B. (1999). Smooth Discrimination Analysis. *The Annals of Statistics*, 27(6):1808–1829.

Maniatis, T. and Tasic, B. (2002). Alternative pre-mRNA splicing and proteome expansion in metazoans. *Nature*, 418(6894):236–243.

Mardia, K., Kent, J., and Bibby, J. (1979). *Multivariate Analysis*. Academic Press.

Marioni, J. C., Mason, C. E., Mane, S. M., Stephens, M., and Gilad, Y. (2008). RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, 18(9):1509–1517.

Marron, J. S., Todd, M. J., and Ahn, J. (2007). Distance-Weighted Discrimination. *Journal of the American Statistical Association*, 102(480):1267–1271.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Wiley, New York.

Meinshausen, N. (2008). Hierarchical testing of variable importance. *Biometrika*, 95(2):265–278.

Metzker, M. L. (2010). Sequencing technologies - the next generation. *Nature Reviews. Genetics*, 11(1):31–46.

Meyerson, M., Gabriel, S., and Getz, G. (2010). Advances in understanding cancer genomes through second-generation sequencing. *Nature Reviews. Genetics*, 11(10):685–696.

Monti, S., Tamayo, P., Mesirov, J., and Golub, T. (2003). Consensus Clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52:91–118.

Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., and Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*, 5(7):621–628.

Nilsen, T. W. and Graveley, B. R. (2010). Expansion of the eukaryotic proteome by alternative splicing. *Nature*, 463(7280):457–463.

Ozsolak, F. and Milos, P. M. (2011). RNA sequencing: advances, challenges and opportunities. *Nature Reviews. Genetics*, 12(2):87–98.

Pan, Q., Shai, O., Lee, L. J., Frey, B. J., and Blencowe, B. J. (2008). Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nature Genetics*, 40(12):1413–1415.

Parker, J. S., Mullins, M., Cheang, M. C., Leung, S., Voduc, D., Vickery, T., Davies, S., Fauron, C., He, X., Hu, Z., Quackenbush, J. F., Stijleman, I. J., Palazzo, J., Marron, J. S., Nobel, A. B., Mardis, E., Nielsen, T. O., Ellis, M. J., Perou, C. M., and Bernard, P. S. (2009). Supervised risk predictor of breast cancer based on intrinsic subtypes. *Journal of Clinical Oncology*, 27(8):1160–1167.

Perou, C., Sørlie, T., Eisen, M. B., van de Rijn, M., Jeffrey, S. S., Rees, C. A., Pollack, J. R., Ross, D. T., Johnsen, H., Akslen, L. A., Fluge, O., Pergamenschikov, A., Williams, C., Zhu, S. X., Lønning, P. E., Børresen-Dale, A. L., Brown, P. O., and Botstein, D. (2000). Molecular portraits of human breast tumours. *Nature*, 406(6797):747–52.

Planque, C., Choi, Y.-H., Guyetant, S., Heuzé-Vourc'h, N., Briollais, L., and Courty, Y. (2010). Alternative splicing variant of kallikrein-related peptidase 8 as an independent predictor of unfavorable prognosis in lung cancer. *Clinical Chemistry*, 56(6):987–997.

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74.

Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & Correlation. *Journal of Machine Learning Technologies*, 2(1):37–63.

Prat, A., Parker, J. S., Karginova, O., Fan, C., Livasy, C., Herschkowitz, J. I., He, X., and Perou, C. M. (2010). Phenotypic and molecular characterization of the claudin-low intrinsic subtype of breast cancer. *Breast Cancer Research*, 12:R68.

Qiao, X. and Liu, Y. (2009). Adaptive weighted learning for unbalanced multicategory classification. *Biometrics*, 65(1):159–168.

Ramsköld, D., Wang, E. T., Burge, C. B., and Sandberg, R. (2009). An abundance of ubiquitously expressed genes revealed by tissue transcriptome sequence data. *PLoS Computational Biology*, 5(12):e1000598.

Rao, C. R. (1973). *Linear Statistical Inference and Its Applications*. Wiley, New York.

Robinson, M. D., McCarthy, D. J., and Smyth, G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140.

Robinson, M. D. and Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, 11(3):R25.

Shalev-Shwartz, S., Singer, Y., Srebro, N., and Cotter, A. (2010). Pegasos: primal estimated sub-gradient solver for SVM. *Mathematical Programming*, 127(1):3–30.

Shen, X., Tseng, G. C., Zhang, X., and Wong, W. H. (2003). On ψ-Learning. *Journal of the American Statistical Association*, 98(463):724–734.

Shimodaira, H. (2004). Approximately unbiased tests of regions using multistep-multiscale bootstrap resampling. *The Annals of Statistics*, 32(6):2616–2641.

Steinwart, I. and Scovel, C. (2007). Fast rates for support vector machines using Gaussian kernels. *The Annals of Statistics*, 35(2):575–607.

Sultan, M., Schulz, M. H., Richard, H., Magen, A., Klingenhoff, A., Scherf, M., Seifert, M., Borodina, T., Soldatov, A., Parkhomchuk, D., Schmidt, D., O'Keeffe, S., Haas, S., Vingron, M., Lehrach, H., and Yaspo, M.-L. (2008). A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science*, 321(5891):956–960.

Suzuki, R. and Shimodaira, H. (2006). Pvclust: an R package for assessing the uncertainty in hierarchical clustering. *Bioinformatics*, 22(12):1540–1542.

Talieri, M., Devetzi, M., Scorilas, A., Pappa, E., Tsapralis, N., Missitzis, I., and Ardavanis, A. (2012). Human kallikrein-related peptidase 12 (KLK12) splice variants expression in breast cancer and their clinical impact. *Tumor Biology*, 33(4):1075–1084.

The Cancer Genome Atlas Research Network (2012). Comprehensive genomic characterization of squamous cell lung cancers. *Nature*, 489(7417):519–525.

The Cancer Genome Atlas Research Network (2014). Comprehensive molecular profiling of lung adenocarcinoma. *Nature*, 511(7511):543–550.

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

Tong, P., Chen, Y., Su, X., and Coombes, K. R. (2013). SIBER: systematic identification of bimodally expressed genes using RNAseq data. *Bioinformatics*, 29(5):605–613.

Trapnell, C., Hendrickson, D. G., Sauvageau, M., Goff, L., Rinn, J. L., and Pachter, L. (2013). Differential analysis of gene regulation at transcript resolution with RNA-seq. *Nature Biotechnology*, 31(1):46–53.

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer.

Vapnik, V. (1998). *Statistical learning theory*. Wiley.

Venables, J. P. (2004). Aberrant and alternative splicing in cancer. *Cancer Research*, 64(21):7647–7654.

Verhaak, R. G. W., Hoadley, K. A., Purdom, E., Wang, V., Qi, Y., Wilkerson, M. D., Miller, C. R., Ding, L., Golub, T., Mesirov, J. P., Alexe, G., Lawrence, M., O'Kelly, M., Tamayo, P., Weir, B. A., Gabriel, S., Winckler, W., Gupta, S., Jakkula, L., Feiler, H. S., Hodgson, J. G., James, C. D., Sarkaria, J. N., Brennan, C., Kahn, A., Spellman, P. T., Wilson, R. K., Speed, T. P., Gray, J. W., Meyerson, M., Getz, G., Perou, C. M., and Hayes, D. N. (2010). Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer Cell*, 17(1):98–110.

Wahba, G. (1999). Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. *Advances in Kernel Methods-Support Vector Learning*, 6:69–87.

Wahba, G. (2002). Soft and hard classification by reproducing kernel Hilbert space methods. *Proceedings of the National Academy of Sciences of the United States of America*, 99(26):16524–16530.

Wang, G.-S. and Cooper, T. A. (2007). Splicing in disease: disruption of the splicing code and the decoding machinery. *Nature Reviews. Genetics*, 8(10):749–761.

Wang, J., Shen, X., and Liu, Y. (2008). Probability estimation for large-margin classifiers. *Biometrika*, 95(1):149–167.

Wang, K., Singh, D., Zeng, Z., Coleman, S. J., Huang, Y., Savich, G. L., He, X., Mieczkowski, P., Grimm, S. A., Perou, C. M., MacLeod, J. N., Chiang, D. Y., Prins, J. F., and Liu, J. (2010). MapSplice: accurate mapping of RNA-seq reads for splice junction discovery. *Nucleic Acids Research*, 38(18):e178.

Wang, Z., Gerstein, M., and Snyder, M. (2009). RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews. Genetics*, 10(1):57–63.

Ward, J. H. J. (1963). Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58(301):236–244.

Wegkamp, M. H. and Yuan, M. (2011). Support vector machines with a reject option. *Bernoulli*, 17(4):1368–1385.

Wilkerson, M. D. and Hayes, D. N. (2010). ConsensusClusterPlus: A class discovery tool with confidence assessments and item tracking. *Bioinformatics*, 26(12):1572–1573.

Wilkerson, M. D., Yin, X., Hoadley, K. A., Liu, Y., Hayward, M. C., Cabanski, C. R., Muldrew, K., Miller, C. R., Randell, S. H., Socinski, M. A., Parsons, A. M., Funkhouser, W. K., Lee, C. B., Roberts, P. J., Thorne, L., Bernard, P. S., Perou, C. M., and Hayes, D. N. (2010). Lung squamous cell carcinoma mRNA expression subtypes are reproducible, clinically important, and correspond to normal cell types. *Clinical Cancer Research*, 16(19):4864–4875.

Witten, D. M. (2011). Classification and clustering of sequencing data using a Poisson model. *The Annals of Applied Statistics*, 5(4):2493–2518.

Wu, H., Wang, C., and Wu, Z. (2013). A new shrinkage estimator for dispersion improves differential expression detection in RNA-seq data. *Biostatistics*, 14(2):232–243.

Xi, L., Feber, A., Gupta, V., Wu, M., Bergemann, A. D., Landreneau, R. J., Litle, V. R., Pennathur, A., Luketich, J. D., and Godfrey, T. E. (2008). Whole genome exon arrays identify differential expression of alternatively spliced, cancer-related genes in lung cancer. *Nucleic Acids Research*, 36(20):6535–6547.

Yu, G., Liu, Y., Thung, K.-H., and Shen, D. (2014). Multi-task linear programming discriminant analysis for the identification of progressive MCI individuals. *PLoS ONE*, 9(5):e96458.

Yuan, M. and Wegkamp, M. H. (2010). Classification methods with reject option based on convex risk minimization. *Journal of Machine Learning Research*, 11:111–130.

Zhang, C. and Liu, Y. (2013). Multicategory Large-Margin Unified Machines. *Journal of Machine Learning Research*, 14:1349–1386.

Zhang, C., Liu, Y., and Wu, Z. (2013). On the effect and remedies of shrinkage on classification probability estimation. *The American Statistician*, 67(3):134–142.