2016

Class discovery via feature selection in unsupervised settings

Curtis, Jessica

http://hdl.handle.net/2144/14548
Boston University
BOSTON UNIVERSITY
GRADUATE SCHOOL OF ARTS AND SCIENCES

Dissertation

CLASS DISCOVERY VIA FEATURE SELECTION IN
UNSUPERVISED SETTINGS

by

JESSICA CURTIS
B.S. Mathematics, Fairfield University, 2006
M.S. Mathematical Statistics, Georgetown University, 2007

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
2016
Acknowledgments

My sincerest thanks to my advisor, Dr. Mark Kon, for his patience, guidance, and continued support throughout my graduate studies. His willingness to let me pursue my own interests has allowed me to grow as a graduate student and as a researcher.

I would also like to thank my future husband, Ryan, for allowing me the room to grow and challenge myself, and for supporting me throughout this process with unwavering confidence in my abilities. I could not have done any of this without his love and support.

A special thanks to all of my friends and family, especially my parents, who continue to inspire me to be the best “me” I can be. I love you all.
Identifying genes linked to the appearance of certain types of cancers and their phenotypes is a well-known and challenging problem in bioinformatics. Discovering marker genes which, upon genetic mutation, drive the proliferation of different types and subtypes of cancer is critical for the development of advanced tests and therapies that will specifically identify, target, and treat certain cancers. Therefore, it is crucial to find methods that are successful in recovering “cancer-critical genes” from the (usually much larger) set of all genes in the human genome.

We approach this problem in the statistical context as a feature (or variable) selection problem for clustering, in the case where the number of important features is typically small (or rare) and the signal of each important feature is typically minimal (or weak). Genetic datasets typically consist of hundreds of samples \( n \) each with tens of thousands gene-level measurements \( p \), resulting in the well-known statistical “large \( p \) small \( n \)” problem. The class or cluster identification is based on the clinical information associated with the type or subtype of the cancer (either known or unknown) for each individual. We discuss and develop novel feature ranking methods, which complement and build upon current methods in the field. These ranking methods are used to select features which contain the most significant information for clustering. Retaining only a small set of useful features based on this ranking aids in both a reduction in data dimensionality, as well as the identification of a set of genes that are crucial in understanding cancer subtypes.

In this paper, we present an outline of cutting-edge feature selection methods, and
provide a detailed explanation of our own contributions to the field. We explain both the practical properties and theoretical advantages of the new tools that we have developed. Additionally, we explore a well-developed case study applying these new feature selection methods to different levels of genetic data to explore their practical implementation within the field of bioinformatics.
Contents

1 Introduction 1
   1.1 Feature Selection in Unsupervised Data . . . . . . . . . . . . . . . . . . . . 1
       1.1.1 General Considerations . . . . . . . . . . . . . . . . . . . . . . . . . 3
   1.2 Previous Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
       1.2.1 Classical FS Methods . . . . . . . . . . . . . . . . . . . . . . . . . . 6
       1.2.2 Recently Developed FS Methods . . . . . . . . . . . . . . . . . . . . 11
   1.3 Novel Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

2 Parametric Bimodal Feature Selection 18
   2.1 Model Definition . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
       2.1.1 Bimodal Feature Selection Method: Larkin’s F test . . . . . . . . . . 20
       2.1.2 Thresholding the F test . . . . . . . . . . . . . . . . . . . . . . . . . 22
       2.1.3 Clustering Method - K-Means Clustering . . . . . . . . . . . . . . . 26
   2.2 Extensions to the Multi-Class Setting . . . . . . . . . . . . . . . . . . . . . 26
   2.3 Simulation Study . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
       2.3.1 Simulation - Multi-class Datasets . . . . . . . . . . . . . . . . . . . . 31
   2.4 Case Study . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
   2.5 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38

3 Nonparametric Bimodal Feature Selection 40
   3.1 Model Definition . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
       3.1.1 Kernel Density Estimation . . . . . . . . . . . . . . . . . . . . . . . . . 42
       3.1.2 Ranking and Thresholding . . . . . . . . . . . . . . . . . . . . . . . . . 44
3.1.3 Clustering Method - K-Means Clustering ............................................. 47
3.2 Extension of Bimodal FS to Multi-Class Clustering Problems ............... 47
3.3 Simulation Study ................................................................................. 48
3.4 Case Study ......................................................................................... 51
3.5 Conclusions ......................................................................................... 52

4 Theoretical Asymptotic Performance .......................................................... 54
4.1 K-Means Clustering and the Data Glut ..................................................... 54
4.2 Performing Feature Selection in K-Means Clustering ................................. 59
4.3 K-means Clustering and the Bimodal F test ............................................. 61
  4.3.1 Inference Concerning the Choice of F-test Threshold .......................... 65
4.4 K-means Clustering and the Bimodal KDE Model ................................... 71
  4.4.1 Thresholding the KDE By Modal Identification and Separation .......... 74
  4.4.2 Thresholding Using Higher Criticism ................................................. 76
4.5 Asymptotic Properties of HCT and UHC .................................................. 81
  4.5.1 The Phase Space Boundary ............................................................... 84
4.6 Connecting Phase Diagrams to Clustering Errors ..................................... 88
4.7 From Ordinary HCT to Unsupervised HCT ............................................... 90

5 Feature Selection Applications: Copy Number Data vs. Gene Expression Data ........................................................................................................ 95
5.1 Explanation of Data Types and Integrative Analysis ................................. 96
  5.1.1 Gene Expression Data ........................................................................ 96
  5.1.2 Copy Number Variation Data ............................................................ 97
  5.1.3 The Importance of Integrated Data ..................................................... 98
5.2 Case Study: Breast Cancer Dataset ......................................................... 99
  5.2.1 Results ............................................................................................. 99
5.3 Case Study: Glioma Dataset ..................................................................... 101
  5.3.1 Results ............................................................................................. 101
# List of Figures

1.1 Possible Outcomes Generated from M Total Hypothesis Tests . . . . . . . . 8  
1.2 The Benjamini/Hochberg Method using FDR . . . . . . . . . . . . . . . . 8  

2.1 Empirical CDF for sorted $F, \tilde{F}$ . . . . . . . . . . . . . . . . . . . . . 23  
2.2 Hamming Distance: Uncorrelated Gaussian Noise . . . . . . . . . . . . . 29  
2.3 Hamming Distance: Correlated Noise - Tri-Diagonal Covariance . . . . . 29  
2.4 Hamming Distance: Correlated Noise - Penta-Diagonal Covariance . . . . 29  
2.5 Effect of Signal Rarity and Strength on Clustering Error . . . . . . . . . . 30  
2.6 Effect of Signal Strength on Clustering Error . . . . . . . . . . . . . . . . 31  
2.7 Effect of Signal Sparsity on Clustering Error . . . . . . . . . . . . . . . . 31  
2.8 Effect of Number Impt Features on Bimodal FS Performance . . . . . . . 33  
2.9 Effect of Class Size on Bimodal FS Performance . . . . . . . . . . . . . 34  
2.10 Effect of Cluster Mean Overlap on Bimodal FS Performance . . . . . . . 35  
2.11 Real Datasets: Two-Class Case . . . . . . . . . . . . . . . . . . . . . . . 35  
2.12 Real Datasets: Overview of Performance . . . . . . . . . . . . . . . . . . 36  
2.13 Real Datasets: Full vs. Reduced Clustering Error Comparison . . . . . . 36  
2.14 Real Datasets: IF-PCA vs Bimodal F test . . . . . . . . . . . . . . . . . . 37  
2.15 Real Datasets: Multi-Class Case . . . . . . . . . . . . . . . . . . . . . . . 37  
2.16 Bimodal FS Performance on Real Multi-class Datasets . . . . . . . . . . . 38  

3.1 Gaussian Noise . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49  
3.2 Non-Gaussian Noise . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49  
3.3 Non-Gaussian Noise Comparison of Methods . . . . . . . . . . . . . . . . . 50  
3.4 Effect of Number of Clusters (k) on Clustering Error . . . . . . . . . . . . 51  

x
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td>Real Datasets: Full vs. Reduced Clustering Error Comparison</td>
<td>51</td>
</tr>
<tr>
<td>3.6</td>
<td>Real Datasets: HC vs Bimodal(parametric) vs Bimodal(nonparametric)</td>
<td>52</td>
</tr>
<tr>
<td>3.7</td>
<td>Real Datasets: HC vs Bimodal(parametric) vs Bimodal(nonparametric)</td>
<td>52</td>
</tr>
<tr>
<td>4.1</td>
<td>Distribution of Normalized Bandwidth</td>
<td>80</td>
</tr>
<tr>
<td>5.1</td>
<td>Case Study: KS vs F vs KDE for GE/CNV data</td>
<td>100</td>
</tr>
<tr>
<td>5.2</td>
<td>Case Study: KS vs F vs KDE for GE/CNV data</td>
<td>100</td>
</tr>
<tr>
<td>5.3</td>
<td>Case Study: KS vs F vs KDE for GE/CNV data</td>
<td>102</td>
</tr>
<tr>
<td>5.4</td>
<td>Case Study: KS vs F vs KDE for GE/CNV data</td>
<td>102</td>
</tr>
<tr>
<td>5.5</td>
<td>Case Study: KS vs F vs KDE for METABRIC data</td>
<td>104</td>
</tr>
<tr>
<td>6.1</td>
<td>Iterative Results - Simulation</td>
<td>111</td>
</tr>
<tr>
<td>6.2</td>
<td>Iterative Results - Lung Cancer Dataset</td>
<td>113</td>
</tr>
<tr>
<td>6.3</td>
<td>Iterative Threshold - Lung Cancer Dataset</td>
<td>114</td>
</tr>
<tr>
<td>6.4</td>
<td>Lung Cancer Dataset - Comparison of Methods</td>
<td>114</td>
</tr>
</tbody>
</table>
List of Symbols

\( y \ldots \) an \( n \times 1 \) vector of response variables, typically cancer classes or subtypes

\( X \ldots \) an \( n \times p \) matrix of covariate data, typically gene-expression data

\( \delta \ldots \) a parameter that is used to determine class-size, particularly in a 2-class scenario

\( n \ldots \) the number of samples in the data

\( p \ldots \) the number of features or dimensions in the full dataset

\( k \ldots \) the number of clusters or classes in the data

\( \mu_k \) or \( \bar{x}_k \) the actual or estimated mean of cluster \( k \)

\( \beta \ldots \) a parameter that determines the rarity of a signal

\( \epsilon \ldots \) the measure of signal sparsity

\( r \ldots \) a parameter that determines the signal strength

\( \tau \ldots \) the measure of signal strength

\( \theta \ldots \) a parameter which defines the relationship between \( n \) and \( p \)

\( RW(\ldots) \) or \( ARW(\ldots) \) the (asymptotic) rare-weak setting

\( t \ldots \) the threshold used for feature selection

\( V \ldots \) the projection matrix for feature selection

\( \sigma^2_{uni}(j) \ldots \) the variance of \( x_i \) in dimension \( j \)

\( \sigma^2_{bi}(j) \ldots \) the minimum variance of \( x_i \) when split into two clusters in dimension \( j \)

\( v \ldots \) the number of splits in the construction of \( \sigma^2_{bi}(j) \)
\( F \ldots \) the test statistic associated with \( \sigma^2_{\text{uni}}(j) \) and \( \sigma^2_{b}(j) \)

\( K \ldots \) the kernel to be used in density estimation

\( \hat{h} \) the estimated bandwidth used in density estimation

\( J_K \ldots \) the k-means objective function

\( H(C, C') \ldots \) the Hamming distance between clusters \( C \) and \( C' \)

\( r_t \ldots \) the set of \( r \) features selected based on threshold \( t \)

\( P(X > x) \ldots \) a right-tail p-value based on the distribution of \( X \)

\( \text{SNR} \ldots \) the signal-to-noise ratio

\( \text{snr} \ldots \) the signal-to-noise ratio in a rare-weak setting

\( \rho^*(\beta) \ldots \) the phase space boundary for classification

\( \rho^*_{\text{exact}}(\beta) \ldots \) the phase space boundary for clustering
Chapter 1

Introduction

In recent years there has been overwhelming growth in the amount of accessible biological data in the area of computational biology. New methods of biological data analysis- e.g. DNA and RNA sequence analysis, proteomics, mass and infrarad spectral analysis, and others- produce data with dimension \( p \) much greater than the number of available samples \( n \). A common issue arising from such new data collection methods is known as the “large \( p \), small \( n \)” scenario. This can pose problems in many aspects of analysis, including class discovery and subtype identification, where the goal is to cluster samples into distinct groups. In this type of big data, information for clustering samples becomes spread across multiple dimensions, adding significant noise to data, and causing many classical clustering methods to break down. The process of feature selection (described below) attempts to address this issue.

1.1 Feature Selection in Unsupervised Data

There are two main types of data structures that occur in large datasets. *Supervised* data structures occur when the group or class labels for each data point are known beforehand. *Unsupervised* data occurs when classes are unknown, and the dataset contains only the intrinsic features of each data point. Supervised data problems dealing with class prediction are known as classification problems, while identifications of natural data groupings in an unsupervised setting are known as clustering problems.

Classical classification and clustering methods have assumed that the dimension of a
dataset, \( p \), is significantly smaller than its size, \( n \). This assumption carries over to theoretic and asymptotic analysis. Correspondingly, asymptotic theory in classification and clustering problems has assumed that the sample size grows faster than the feature space (i.e. \( n \to \infty \) faster than \( p \to \infty \)). “Large \( p \), small \( n \)” problems, however, violate this assumption in both static and asymptotic contexts. One solution to such high dimensionality is to reduce the dimension to a smaller subset of relevant features that are important to data grouping- a process called feature selection (FS). This reduction retains a smaller set of features that preserve the original characteristics of the variables. Reducing a dataset to a more manageable dimension subsequently reduces the effect of the violation of the \( n \gg p \) assumption and allows for sound implementation of classical clustering methods.

*Filter methods* are a type of feature selection method that are used before the classification algorithm is implemented to find a relevant subset that is used as input in the classification algorithm. There are two types of filter methods: threshold filter methods, in which features are selected based on whether a calculated criterion for each feature lies above a certain threshold, and ranking filter methods, in which features are selected based on a ranked list of features. The most desirable property of filter methods is that they are performed separately from classification, which reduces computation time and only depends on the intrinsic qualities and properties of the data.

There are several scenarios in which FS is advantageous; particularly when features are highly informative but restricted to a small set; i.e. when useful features are rare and their signals are weak. Accordingly, FS has been studied extensively in a supervised context. In contrast, unsupervised selection has seen little attention until recently, due in part to the complexities of clustering vs. classification problems. In the case of cancer subtype identification and discovery, unsupervised data is common, while supervised contexts are rarely seen in practice. This is because the main goal in these studies is to use gene-level data to identify and predict new subtypes of cancer, rather than verify known subtypes. Therefore, the extension of supervised approaches to unsupervised clustering has become a particular point of interest for researchers in this field of study.
Feature selection has many advantages in statistical modeling. Ideally, reducing data dimensionality leads to faster model training and a reduction of run time and computational cost, while maintaining good class prediction. In extreme cases, particularly where useful features are rare or their signals are weak, FS can result in smaller classification errors [47]. However, FS should also avoid typical problems that occur in data analysis, such as overfitting, ignoring crucial features, or poorly altering the original representation of the data. The latter is particularly important because keeping the data intact allows for more interpretable results. The subsequent chapters will demonstrate the notion that successful unsupervised FS methods are very useful in practice, because they have the potential to improve clustering methods dramatically in the case of high dimensional data.

1.1.1 General Considerations

Our specific focus is feature selection for improvement of unsupervised learning, and more specifically clustering. Given a dataset with \( n \) observations, each of which contains \( p \) measurements or features, the goal of clustering is to determine a natural partitioning of the dataset into a (usually predetermined) number of parts.

Formally, we begin by considering a dataset \( \{x_i, y_i\} \) where \( x_i = (x_{i1}, x_{i2}, \ldots x_{ip}) \in \mathbb{R}^p \) is a vector containing \( p \) measurements for each individual \( i \). The value \( y_i \) represents the cluster to which instance \( i \) should belong, which is assumed to be fixed, but unknown. It may represent a hidden class, for instance, one that has been heretofore unrecognized and unlabeled. Note that if only 2 clusters are assumed, typically \( y_i = \pm 1 \); if there are \( k > 2 \) clusters, typically \( y_i \in \{1, 2, \ldots k\} \).

Based on the hypothesis of the existence of clusters, we assume that the dataset \( x_i \) is constructed as follows:

\[
x_i = \sum_k \mu_k I(y_i = k) + z_i. \tag{1.1}
\]

Here, \( \mu_k \) represents the contrast mean for each cluster, which we assume satisfies the constraint \( \sum_k \mu_k = 0 \) (without loss of generality). The indicator variable \( I(y_i = k) \) represents
the true class assignment of each data point, which we are trying to estimate. The values $z_i \in \mathbb{R}^p$ form a vector composed of entries coming from a zero-mean known or unknown distribution, representing measurement error. Note that the distribution of errors is the same for each class.

It is important to emphasize that actual cluster assignments $y_i$ are based on fundamental characteristics, but are unknown in practice. For example, in bioinformatics, the clusters may be based on differing cancer subtypes in a dataset of tissue samples. Our goal is to reduce the dimension of $x_i$ using feature selection, and to identify a clustering consistent with the classes defined by the $y_i$ with minimal error.

### 1.1.1.1 Simplifications

In the two-class or two-cluster scenario, we sometimes redefine the label vector $y = \pm 1$ so that:

$$y_i = \begin{cases} 
(1 - \delta) & \text{with probability } \delta \\
-\delta & \text{with probability } (1 - \delta) 
\end{cases} \quad (1.2)$$

Under this definition, $\delta$ represents the fraction of samples in each cluster. This notation is developed in [48, 89, 90] and, while unconventional, allows the label vector to be proportional to a Bernoulli($\delta$) distribution, making it particularly useful in simulating random datasets of unequal class sizes.

Similarly, we adopt a typical assumption that the mean signal for each of $j$ features, $\mu^2(j)$, can be represented by two values: $\mu^2(j) = \omega_0$, typically just 0 (the signal for noisy features) or $\mu^2(j) = \omega_{r_p}$, a non-zero constant (the constant signal for important features). Note that this is a common assumption in the context of signal detection [18, 19, 21, 23, 46, 51]. A rescaling of the mean is appropriate under this assumption, which is indicative of a rare-weak setting:

$$\mu^2(j) \sim (1 - \epsilon_p)\omega_0 + \epsilon_p \omega_{r_p} \quad (1.3)$$

Here, $\epsilon_p$ represents a measure of the signal sparsity - the number of features with relevant
class information. Additionally, $\omega_0$ represents a distribution with a point mass at 0- shrinking the signal for noisy features- and $\omega_{\tau_p}$ represents the distribution with a point mass at some non-zero constant- the signal of each important or selected feature. We can then choose to define the signal strength $\tau_p$ and signal sparsity $\epsilon_p$ to reflect a rare-weak scenario, such that

$$\tau_p = \sqrt{2r \log p} \quad \text{and} \quad \epsilon_p = p^{-\beta}$$

Under this parameterization, $r \in (0, 1)$ and $\beta \in (0, 1)$ are used to set and identify signal sparsity and signal strength, respectively. Therefore, knowing the values of $\beta$, $r$, and $\delta$ allows us to fully identify the rare-weak model scenario: $RWM(\beta, r, \delta)$. Note that through this parametrization, signal strength and sparsity are determined by the number of features, $p$. As $p \to \infty$, $\epsilon_p \to 0$, reflecting a sparse signal. Additionally, when signals are sparse, the strength of each signal tends to fall on the order of $O(\sqrt{\log p})[46]$, and thus $\tau_p \to \infty$, albeit very slowly. If the signal strength is strictly larger than $O(\sqrt{\log p})$, then feature selection is easy, and classical clustering methods perform adequately with minimal error. When the signal is significantly less than $O(\sqrt{\log p})$, however, adequate signal detection is nearly impossible. Authors in [21, 23, 47, 48] argue that feature selection is most useful in the case where relevant features are rare and weak and defined by the parameterization above. Consequently, the methods developed here are particularly applied to situations that reflect this setting. For simplicity, we typically refer to the parameters $\epsilon_p$ and $\tau_p$ as just $\epsilon$ and $\tau$, with the assumption that both parameters are related to dimension $p$ by the relationships given above.

Finally, while we typically use the term “features” to describe actual columns of the data matrix $X$, we note here that it is also possible to consider features as transformations or linear combinations of the data matrix as well. In most cases, we assume that features are strictly data matrix columns, although we recognize that the results presented in this thesis can easily be extended to incorporate more complicated features arising from transformations of the data matrix.
1.2 Previous Work

1.2.1 Classical FS Methods

1.2.1.1 The ANOVA Model

A common approach to feature selection is based on the ANOVA model. Authors in [4] describe the ANOVA model in the context of rare-weak signal settings. The classical ANOVA setup considers the relationship between $X$ and $y$ to be defined as:

$$y_j = \beta_j X + z_j, \quad j = 1, \ldots, p$$

where $\beta_j$ captures the group effect for the $j$-th group, and $z_j \sim N(0, \sigma^2)$ represent iid measurement errors. Within the context of supervised data and classification, the value of $\beta_j$ refers to the average expression level for gene $j$ across all $n$ samples.

The basic problem for signal detection can be expressed by the ANOVA model as an overall global testing problem:

$$H_0 : \beta_1 = \beta_2 = \ldots = \beta_p = 0$$

$$H_A : \text{at least 1 } \beta_j \neq 0$$

Using this setup, the ANOVA method will allow us to determine whether any of the feature expression values can be assumed to be nonzero, indicating that the feature $j$ is useful for classification. However, this particular setup will only identify whether features can be eliminated from the classification algorithm; it does not identify which $\beta_j$'s should be included and which can be ignored.

To alleviate this issue, a variation of the ANOVA method known as the “Max Test” was developed to consider many individual ANOVA tests, one for each $\beta_j$, and then combine them into a multiple testing problem by applying a Bonferroni-type correction. Formally, this is done by considering the test:

$$H_{0,j} : \beta_j = 0 \quad H_{A,j} : \beta_j \neq 0$$
Then, a minimum p-value is computed and compared with an appropriate threshold. This is equivalent to rejecting the null when \( \max_j |\beta_j| \geq A \) for some \( A > 0 \), where \( A \) is the desired feature selection threshold.

The value of the threshold, \( A \), is determined by the asymptotic theory of these two methods. Specifically, if the errors \( z_j \) are known or assumed to be normally distributed, then the ANOVA method requires the threshold value \( A \) to be on the order of \( o(p^{1/4}) \), whereas the Max method requires \( A \) to be on the order of \( o((2 \log p)^{1/2}) \) \[4\]. When \( p \) is small, these values are very similar; however, when \( p \) is large, as is the case when dealing with genomic data, these values can differ dramatically.

Using the detection boundary in the rare, weak setting, the threshold \( A \) is chosen as \( A = \sqrt{2 \rho^*(\alpha) \log(p)} \), where \( \rho^*(\alpha) \) is the signal detection boundary, which is defined exactly as \( \rho^*(\beta) \) in the phase-space discussion present in Chapter 4. This threshold for \( A \) is known as the “sparse detection threshold.” \[4\]

1.2.1.2 False Discovery Rates

Another classical thresholding method is the False Discovery Rate (FDR) approach to classification. The theoretical background behind this method is given in \[43\].

Similar to the ANOVA testing setup, we consider the problem of feature selection as a multiple testing problem. For each feature \( j \), we consider the hypotheses:

\[
H_{0,j} = \text{classification } (Y) \text{ and feature } j \ (X_j) \text{ are unrelated}
\]

\[
H_{A,j} = \text{classification } (Y) \text{ and feature } j \ (X_j) \text{ are related}
\]

This test is simply a t-test for each feature \( j \). As usual, we reject \( H_{0,j} \) at level \( \alpha \) when the p-value for gene \( j \), \( p(j) \) is small, i.e. when \( p(j) < \alpha \). Within the context of multiple testing, we must consider an aggregated error measure, known as the False Discovery Rate (FDR), defined as the expected proportion of significant genes that are actually insignificant. Consider figure 1.1, which shows the possible outcomes from \( M \) hypothesis tests (in the case of genes, \( M = p \) = total number of genes).
Figure 1.1: Possible Outcomes Generated from M Total Hypothesis Tests

<table>
<thead>
<tr>
<th></th>
<th>Called Significant</th>
<th>Called Not Significant</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_0$ True</td>
<td>$U$</td>
<td>$V$</td>
<td>$M_0$</td>
</tr>
<tr>
<td>$H_0$ False</td>
<td>$T$</td>
<td>$S$</td>
<td>$M_1$</td>
</tr>
<tr>
<td>Total</td>
<td>$M - R$</td>
<td>$R$</td>
<td>$M$</td>
</tr>
</tbody>
</table>

Based on this figure, the false discovery rate can be calculated as $FDR = \mathbb{E}\left[\frac{V}{R}\right]$.

The use of the FDR as a testing procedure was developed by authors in [6], and is described as an algorithm in [43]. The summary of this algorithm is given in the following figure.

**Algorithm 18.2 Benjamini-Hochberg (BH) Method.**

1. Fix the false discovery rate $\alpha$ and let $p(1) \leq p(2) \leq \cdots \leq p(M)$ denote the ordered $p$-values.
2. Define
   \[L = \max\left\{ j : p(j) < \alpha \cdot \frac{j}{M} \right\},\]
   \[(18.44)\]
3. Reject all hypotheses $H_0j$ for which $p_j \leq p(L)$, the BH rejection threshold.

Figure 1.2: The Benjamini/Hochberg Method using FDR.

In bioinformatics, the method described above finds the genes with $p$-values that fall below the given $p$-value threshold, $p_L$. Under the assumption of independent tests (meaning the genes themselves are independently related to the classification, a strong yet useful and common assumption), the bound of the FDR is given $FDR \leq \frac{M_0}{M} \alpha \leq \alpha$. Therefore, we proceed by choosing a limiting $p$-value that corresponds to the limit of the FDR. From this explanation, it is not clear how to find the appropriate thresholding value $p_L$; in practice, finding the appropriate threshold is actually quite difficult. In many cases, therefore, the
bound is calculated using a plug-in approach for the FDR threshold. The plug-in approach has been proven to be equivalent to the method above, with a major advantage: this approach deals solely with test statistics instead of p-values.

The plug-in estimate is based on the approximation $\hat{FDR} = \frac{E(V/R)}{E(R)} \approx E\left(\frac{V}{R}\right)$ and in general $\hat{FDR}$ is a consistent estimate of FDR.

### 1.2.1.3 Principal Component Analysis

Principal Component Analysis (or PCA) is a method that is used to reduce the dimensionality of a given dataset. It works by using a transformation to project the data points onto orthogonal curves or lines. The function of the curve/line (called a principal component) is then used to describe the data structure and its variability rather than the individual data points. The process of using PCA is explained in [18].

More formally, we are given a set of data vectors, $X = \{x_i\}_{i=1}^N$. Our goal is to determine the function $f(\lambda)$ that projects points $x_i$ to points $p_{x_i}$ that fall on the line. In particular, we are looking to determine the function:

$$f(\lambda) = E(X|P(X) = f(\lambda))$$

where $E(X|P(X) = f(\lambda))$ is the expectation (average) of all points $x_i$. This function typically represents a similarity measure that determines a graphical representation of the data, as in, it projects the data matrix $X$ onto a graph with vertices determined by data points and edges determined by the function $f(\lambda)$.

Note that $f(\lambda) = (f_1(\lambda), f_2(\lambda), \ldots, f_c(\lambda))$ will have coordinates consisting of the $c$ principal components, where each principal component measures and explains as much variation in the data as possible and is orthogonal to all other components. Essentially, this process is similar to k-means clustering, with the constraint that the means (cluster centers) all lie on a given curve or line. Thus, this process effectively reduces data dimension since the number of principal components $c$ is always less than or equal to the number of original
variables \( p \).

1.2.1.4 Linear Discriminant Analysis

Linear Discriminant Analysis (or LDA) is a method developed in machine learning to find a linear combination of features that best separates two or more classes. The method works similarly to linear least squares regression, with the distinction that it is developed to handle categorical response data. LDA is closely related to PCA, but operates on determining the differences between classes rather than similarities within, and therefore classifies data points based on linear combinations rather than principal components.

In practice, LDA looks to determine the \( k \) linear discriminant functions \( f_k(\delta) \) that act as decision boundaries separating the \( k \) classes. The process assumes the probability that sample \( x_i \) belongs to class \( k \) is normally distributed with equal or common covariance, i.e., \( P(x_i|Y_i = k) \sim N(\mu_k, \Sigma) \). Under these assumptions, it looks to maximize the functions

\[
 f_k(\delta) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k
\]

where \( \mu_k \) is the \( k \)th cluster mean, \( \Sigma \) is the common covariance for each cluster \( k \), and \( \pi_k \) represents the proportion of samples in cluster \( k \), estimated by \( n_k/N \).

These functions are based on Fisher’s linear discriminant, which measures class separation as a ratio of between-class variance to within-class variance. Fisher’s linear discriminant \( S \) is formally defined between class \( k \) and class \( l \) as:

\[
 S = \frac{w^T (\mu_k - \mu_l)^2 w}{w^T \Sigma w}
\]

where \( w \) represents a linear combination of \( x_i \). Thus, similar to PCA, the data points \( x_i \) are projected onto \( w \) through linear combinations, and the line which best separates the points (perpendicular to \( w \)) is determined. The separation \( S \) is maximized when \( w \propto \Sigma^{-1}(\mu_k - \mu_l) \). Consequently, maximizing \( S \) is an identical process to maximizing \( f_k(\delta) \) under the assumptions highlighted above.
1.2.2 Recently Developed FS Methods

1.2.2.1 Higher Criticism

Higher criticism thresholding (HCT) is one of the more prominent methods of feature selection which was developed recently in [Donoho/Jin]. This method is a supervised filter feature selection method in which an optimal feature subset is determined by thresholding.

Suppose we are given a set of labeled training samples \((X_i, y_i), i = 1...n\) such that \(X_i \in \mathbb{R}^p\) is a feature vector with \(p\) elements and \(y_i = \pm 1\) denotes the classification of sample \(i\).

The assumptions given in this method are as follows: (1) the training set contains an equal number of 1’s and -1’s; (2) the feature vectors \(X_i\) are distributed as \(X_i \sim N(Y_i\mu, \Sigma), i = 1...n\) for an unknown mean vector \(\mu \in \mathbb{R}^p\) with the feature covariance matrix \(\Sigma\); (3) feature correlations can be ignored, i.e. \(\Sigma\) is a diagonal matrix; (4) features are standardized to variance 1.

From the data \((X_i, y_i)\) authors in [22] define a vector of Z-scores such that:

\[
Z(j) = n^{(-1/2)}\Sigma_i y_i X_i(j) \quad \text{for } j = 1,...,p
\]

These correspond to the hypothesis test for a relation between each feature and the class variable- they represent the Z-score for testing the null hypothesis \(H_{0,j} : \text{Cov}(Y, X(j)) = 0\) against a two-sided alternative. The theoretical intuition is that relevant features with nonzero values for \(\mu(j)\) will also have nonzero values of \(Z(j)\), and therefore testing the above hypothesis is functionally equivalent to testing \(H_{0,j} : \mu(j) = 0\). The assumptions of the higher criticism approach mentioned above imply that \(Z \sim N(\sqrt{n}\mu, I_n)\).

We proceed by determining the p-values for each feature, such that:

\[
p(j) = \text{Prob}\{|N(0, 1)| > |Z(i)|\}, \text{ for } j = 1,...,p
\]
We then arrange and assign a new label $\pi(j)$ to the sorted p-values such that

$$\pi(1) \leq \pi(2) \leq \pi(3) \leq \ldots \leq \pi(p)$$

Note that the p-values are considered to be uniformly distributed under the null ($p(j) \sim U[0,p]$), and therefore the ordered p-values should be asymptotically normally distributed (i.e. $\pi(i) \sim_{approx} N(i/p, i/p(1-i/p))$). This asymptotic distribution is the theoretical basis for higher criticism thresholding. The higher criticism objective is then defined as:

$$HC(j; \pi(j)) = \sqrt{p} \frac{j/p - \pi(j)}{\sqrt{j/p(1-j/p)}} \quad (1.4)$$

This quantity can be thought of effectively as the “Z-score of p-values”, in which the HC statistic measures the discrepancy between the global null hypothesis (that the ordered p-values are distributed uniformly) and the two-sided alternative. Define $\alpha \in [0, 1]$. Then the HC test statistic is given by $HC^* = \max_{1 \leq i \leq \alpha p} HC(i; \pi(i))$, and the threshold is chosen based on the feature whose score achieves this maximization $HC^*$.

In other words, let $HC^*$ be achieved at index $i$. Then the higher criticism threshold is given by: $\hat{t}^{HC} = |Z_{(i)}|$. The HC method selects only features with Z-scores that exceed $\hat{t}^{HC}$.

Because this test is based on a global null hypothesis, authors in [22] argue that the HC statistic and threshold “reflect the shift in emphasis from single test results to the whole collection of tests. The HC test statistic was developed to detect the presence of a small fraction of non-null hypothesis among many truly null hypotheses.” [22] Thus, we can use the standardized distribution of the p-values to find an appropriate cutoff for the most important features to be added to our classification model.
1.2.2.2 Comparison of FDR to Higher Criticism

Authors Donoho and Jin [21] and Klaus and Strimmer [51] both argue that the HCT method is more advantageous in a rare-weak data setting than other methods. In particular, Donoho and Jin specifically state that “the HCT-based feature selection classifiers are radically simpler than all of the other methods being considered [including Bagboost, LogitBoost, SVM, Random Forests, PAM, and the classical methods DLDA and KNN]” [21]. This is due to the fact that the HC method does not require tuning or cross-validation to achieve similar performance to the other methods. Additionally, based on numerical and empirical evidence, authors in [21] argue HCT will outperform the FDR methods in rare-weak data settings because the HCT method will have:

- a lower threshold,
- a higher false-feature discovery rate (FDR) than other approaches,
- a lower missed feature detection rate (MDR) than other approaches,
- a better misclassification rate (MCR) than other approaches.

Based on these advantages, it is clear that HCT is an extremely useful FS method.

1.2.2.3 Unsupervised FS methods: IF-PCA and Nonparametric FS

While unsupervised FS is fairly new, some FS schemes have been proposed. Authors Jin and Wang [49] has developed an unsupervised method which they refer to as IF-PCA. This method operates almost identically to the HCT method described above, with one major difference: their p-values are determined based on the Kolmogorov-Smirnov (KS) score for the distribution of each feature, rather than depending on measuring a relation between $X$ and $y$. The $KS$ statistic is defined by the distance between the empirical CDF $F_n(x(j))$ of the feature distribution, and the theoretical CDF $F(x(j))$, which authors in [49] assume to be Gaussian:

$$KS(j) = \sup_x |F_n(x(j)) - F(x(j))|$$ (1.5)
Feature selection is done in the $IF - PCA$ method using a modified version of higher criticism thresholding for an unsupervised context, which we refer to as unsupervised higher criticism ($UHC$):

$$UHC(j, \pi(j)) = \frac{j/p - \pi(j)}{\sqrt{\max\{(j/p - \pi(j)), 0\}} + j/p}$$ (1.6)

[Jin/Wang] argue that the $UHC$ functional acts similarly to the HCT in a supervised context. Their results are shown in the context of spectral k-means clustering, in which samples are clustered based on the eigenvectors of the covariance matrix rather than the data itself. This process is further discussed in Chapter 4.

The $KS$ statistic acts as a measure of deviation between the empirical feature distribution and the theoretical distribution of the feature under the null, thus providing an adequate measure of feature importance for clustering. However, this test requires a theoretical assumption for the distribution of noise in the data, which is typically unknown in practice. Additionally, spectral clustering is based on eigenvalues instead of data values, which could hinder the interpretations of resulting cluster means.

Geng et al. [33] have also tapped into a nonparametric version of unsupervised FS, selecting features using wavelet-based spike sorting in the context of neurological data. They particularly use kernel density estimation to determine a nonparametric estimate for the distribution of each feature, and then use that estimate to determine whether the feature carries clustering information or pure noise. Their feature selection is based on an ad-hoc method of visually identifying peaks and separation between peaks in the univariate feature distributions.

The methods that we propose here attempt to extend the ideas of both of these existing methods, with the particular advantage that the new method can be applied to higher level clusterings (with multiple classes) and provide similar and sometimes even superior performance. The application of the method proposed here has a great deal of potential in many important biological fields, including cancer subtype identification and discovery.
1.3 Novel Contributions

To discover new and better ways to address the problem of unsupervised feature selection in “big data” scenarios, we have developed a novel approach that is both simple and effective. The new models presented here are generated based on the assumption that features which contain pertinent clustering information will have univariate distributions across samples that are markedly different from distributions of noisy features. Results demonstrate that this model works remarkably well for particularly rare-weak scenarios, i.e., large datasets where significant features are rare and their signals are weak. We develop this main idea in the context of a few different types of models, and explore both the theoretical advantages of such a model and the usefulness of the model in particular bioinformatics applications.

The contributions are thus as follows:

1. We develop a parametric model for feature selection based on the bimodality of feature distributions under a two-class clustering scheme. Under certain standard assumptions for feature selection problems, we rank features based on a novel “measure of bimodality,” and implement a variety of thresholding techniques to select the top important features.

2. Mirroring the first model, we develop a nonparametric version of the parametric model, with a nonparametric bimodality measure used in feature ranking. This second model is particularly useful because it allows the relaxation of distribution assumptions on the data, therefore making it particularly useful in the case of highly skewed feature distributions, or largely unequal cluster sizes. The thresholding done here is also modified to incorporate fewer model assumptions, i.e. is particularly developed in the nonparametric setting.

3. We extend the ideas present in both 1) and 2) to multi-class cluster settings, i.e. problems where more than two clusters are known to be present. This addition is particularly useful as it allows these methods to be applied to situations in which
more than two clusters are found.

4. Within the context of bioinformatics, we explore the use of different common data types for the feature selection problem. In particular, we compare the usefulness of the novel feature selection methods in two particular data types - gene expression data, and copy number variation data. We discuss the performance of these methods and the future directions of feature selection as integrated data platforms become available.

5. We explore the theoretical advantages of the methods described above. In particular, we discuss the inference concerning the thresholding method in each situation, which is the critical parameter in this type of problem. We also discuss in detail what is known as the “phase space boundary,” beyond which significant features are so rare and/or their signals are so weak that effective clustering is impossible. Within this discussion, I develop the argument surrounding the natural benefits of using a particular thresholding method which mirrors this theoretical boundary.

6. Finally, we discuss a novel iterative approach to the unsupervised feature selection problem. This method iterates between clustering and feature selection, and takes advantage of supervised feature selection methods. This is particularly useful since the field of supervised feature selection is so widely studied, and many models have been developed - the iterative nature of the method allows us to take advantage of these supervised models in an unsupervised setting. The method allows clustering and feature selection to be performed numerous times until an ideal clustering is found, i.e. until the method converges. This alleviates the thresholding problem in a filtered and ranked feature selection scenario, which is known to be the more difficult problem in an unsupervised setting.

The parametric feature selection model is described in Chapter 2, for both two-cluster and multi-cluster scenarios. A similar discussion of the nonparametric model is outlined
in Chapter 3. Chapter 4 contains the theoretical arguments for the models developed in Chapters 2 and 3, as well as a discussion of the asymptotic properties of these models. In Chapter 5, we discuss the differences in model performance within bioinformatics based on genetic data type. Finally, in Chapter 6, we explain the novel iterative feature selection/clustering method, and both its advantages and disadvantages compared to a ranking filter method of feature selection.
Chapter 2

Parametric Bimodal Feature Selection

The first new feature selection method proposed here is a novel filter and ranking method that takes advantage of the implied univariate distribution of each feature. This method is based on the notion that features containing cluster information will have a markedly different univariate distribution of the \( n \) samples than features that are irrelevant in the clustering scheme. This parametric model assumes the distribution of noise and the level of importance of features in the model, both of which are common assumptions in practice. A description of the model is given below. The model is tested and evaluated in both simulated and real-world scenarios, and conclusions based on the performance of the model are discussed toward the end of this chapter.

2.1 Model Definition

The new feature selection algorithm proposed here capitalizes on the theoretical notion that features that are important to clustering will have bimodal marginal distributions (in the case of 2 clusters) or multimodal marginal distributions (in the case of more than two clusters). Each mode of the marginal distribution represents an average feature value for each class. Contrastingly, noise features that carry little class information are expected to follow a unimodal distribution. In general, considering a dataset \( X \) consisting of \( n \) observations and \( p \) features, the algorithm selects features in the following manner:

1. For each feature \( 1 \leq j \leq p \), determine the empirical univariate distribution of the \( n \) values (e.g. generate a histogram of \( n \) values for feature \( j \));
2. determine the likeliness that the distribution is unimodal vs. bimodal using a “bimodality measure”;

3. rank the features based on the bimodality measure, from “most likely bimodal/least likely unimodal” to “least likely bimodal/most likely unimodal”;

4. determine a threshold cutoff to select \( r \) features with strongest bimodal values;

5. generate new reduced dataset of size \([n \times r]\);

6. perform clustering on reduced dataset using classical clustering approaches;

7. assign classes based on resulting clustering.

The algorithm we propose uses Larkin’s F test [53] as a bimodality measure in step 2. This test generates a value \( F \) by comparing the ratio of the unimodal variance of the marginal distribution of the feature to the minimum average variance of a bimodal version of the marginal distribution of the same feature. Large values of \( F \) correspond to likely bimodal distributions. We consider two thresholding methods for the F test in step 4 based on whether cluster assignments are entirely unknown, or known but hidden.

2.1.0.4 Assumptions in the Model

In order to implement this test, a few assumptions must be made. In particular, we assume that the data follows a Gaussian mixture model; i.e. \( X \sim \sum_{k=1}^{k} N(\mu_k, \Sigma_k) \). We assume \( \mu_k \in \mathbb{R}^p \) is a vector representing the contrast means, and \( \Sigma_k \) represents measurement noise. The assumption of a Gaussian mixture model, while somewhat restrictive, is also a common assumption among the literature. [22, 21, 23, 46, 47, 90] In most cases, we assume \( \Sigma_k = I_p \), meaning that features are normalized to have variance 1. Additionally, we assume features are uncorrelated and the feature relevance for clustering is somehow captured in the marginal distribution of the features. The latter is a powerful yet appropriate assumption in the case of big data, mainly because estimation of covariances is extremely difficult, and
because significant features are assumed to be so rare and their signals so weak that we expect only very weak correlations between features to exist, if at all. We recognize that this assumption restricts the application of the method, without further exploration of the performance in the context of correlated features.

2.1.1 Bimodal Feature Selection Method: Larkin’s F test

2.1.1.1 The Test

Larkin’s F test [53] was developed to determine whether the distribution of a continuous univariate statistic was unimodal or bimodal based on a ratio of variance. It directly compares the variance of the distribution assuming it is a unimodal distribution ($\sigma^2_{uni}$), to the minimum variance of the distribution when it is considered to be a concatenation of two unimodal distributions ($\sigma^2_{bi}$). This test works under the assumption that the noise in the data is normally distributed, a common assumption in practice. Consequently, this test compares the calculation of variance based on the assumption of a single Gaussian distribution vs. the assumption of an underlying Gaussian mixture. Ultimately, the normality assumption implies that each variance measure will follow a chi-squared distribution and ultimately result in an F test for bimodality. The resulting F ratio can be seen as a quantitative measure of bimodality, which is large if the distribution is bimodal, and small if it is unimodal.

For each feature $j$, we define an estimate for unimodal and bimodal variance measure, $\sigma^2_{uni}(j)$ and $\sigma^2_{bi}(j)$. The computation of $\hat{\sigma}^2_{uni}(j)$ is straightforward and similar to the marginal variance of the feature, simply given by the formula:

$$\hat{\sigma}^2_{uni}(j) = \frac{\sum_{i=1}^{n} (x_i(j) - \bar{x}(j))^2}{n - 1}$$

The computation of $\sigma^2_{bi}(j)$, however, is somewhat more complicated. The distribution of each feature $j$ is divided into two parts repeatedly (for a total of $v$ times), such that there
is at least one data point in each part. For example, in the case of \( n \) observations, we expect part 1 to contain \( n_1(v) \) observations and part 2 to contain \( n_2(v) \) observations, such that \( n_2(v) = n - n_1(v) \). Then the mean and variance based on the distribution for each part separately is computed, and the average of the variances is found; the value of \( \sigma_{bi}^2(j) \) is the minimum of these averages. Formally, for each feature \((j)\):

- Generate \( v \) splits; i.e., determine a point at which to “split” the univariate distribution such that \( n_1 \) points lie to the left of the split, and \( n_2 \) points lie to the right of the split, noting that \( n_1 + n_2 = n \).

- For each of \( v \) splits, find \( \mu_{n_1}(v) \), \( \mu_{n_2}(v) \), \( \sigma_1^2(v) \) and \( \sigma_2^2(v) \), such that:

\[
\mu_{n_1}(v) = \frac{1}{n_1} \sum_{i=1}^{n_1} x_{ij} ; \quad \mu_{n_2}(v) = \frac{1}{n_2} \sum_{i=n_1+1}^{n} x_{ij} \\
\sigma_1^2(v) = \frac{1}{n_1 - 1} \sum_{j=1}^{n_1} (x_{ij} - \mu_{n_1})^2 ; \quad \sigma_2^2(v) = \frac{1}{n_2 - 1} \sum_{j=n_1+1}^{n} (x_{ij} - \mu_{n_2})^2
\]

- For each of \( v \) splits, find the average variance, \( \sigma^2(v) = \frac{1}{v} (\sigma_1^2(v) + \sigma_2^2(v)) \).

- Then \( \sigma_{bi}^2(j) = \min_v (\sigma^2(v)) \).

The resulting \( F \) test is given by the ratio:

\[
F = \frac{\sigma_{uni}^2}{\sigma_{bi}^2}
\]

By construction, unimodal distributions will have significantly lower values of \( \sigma_{uni}^2 \) compared to other features. Similarly, they will most likely also have higher values of \( \sigma_{bi}^2 \) compared to other features. Note that these will lead to small \( F \) ratios for unimodal distributions. Conversely, bimodal distributions will have large numerator variance and smaller denominator variance, leading to higher \( F \) ratios. Thus, we argue that this test provides satisfactory ranking for feature importance for clustering.
2.1.2 Thresholding the F test

In developing this method, Larkin explains “F ratios near 4.0...is roughly the value at which bimodality appears by eye, and [is significant] at the 1% level in actual samples of size 30-100.” [53] However, this thresholding method is a rough estimate for determining bimodal vs. unimodal distributions. For our purposes, we prefer a more data-dependent threshold.

In our case, we considered and compared two main thresholding methods. These methods are referred to as the empirical threshold, which is useful in what we call semi-supervised data realms, where the true classes are known but hidden during feature ranking. The second method is referred to as the HC-F threshold, which is an extension of the HCT as set forth in section 1.2.2, and can be used in completely unsupervised settings.

2.1.2.1 Empirical Threshold

Once features are ranked based on the $F$ statistic, we look to find a cutoff threshold $t$ such that choosing a feature $j$ where $F(j) \geq t$ is equivalent to selecting features which will, by construction, be the dimensions in which bimodality is strongest. We would then cluster on a set of features $\hat{S}_t = \{1 \leq j \leq p : F(j) \geq t\}$, which would ideally lead to optimal cluster discrimination.

We begin by ranking features by the $F$-statistic as defined above; in particular let $\tilde{F}$ represent the distribution of the ordered $F$ values such that $\tilde{F}(1) \geq \tilde{F}(2) \geq ... \geq \tilde{F}(p)$. A visual example of the graph of $\tilde{F}$ is given in the following figure. Note the shape of the distribution is such that the values of $\tilde{F}$ drop drastically at first, and then level out. This indicates that there are few features that contain significant clustering information, followed by many that contain an equal amount of information for clustering purposes.
We generate a list of $m$ possible thresholds $T_m = \{t_1, t_2, \ldots, t_m\}$ based on the quantiles of this empirical distribution function. In particular, we choose possible thresholding values in the list of $T_m$ such that only a small percentage of features (less than 20%) are kept. Once a list of possible thresholds has been identified, we choose an ideal threshold in the context of semi-supervised data, so that our final threshold $t^*$ is the $F$ value for which the selected features $\hat{S}$ produce minimal clustering error when compared to the true clusters.

There is a caveat to using the heuristic approach for thresholding, namely that this approach produces a list of possible thresholds and, thus, generates a set of possible reduced feature sets. The ideal threshold is then determined as the choice that produces the smallest difference between the assigned clusters $C$ and the true underlying clusters $\tilde{C}$. Therefore, in order to implement this thresholding method, we must know the true underlying class assignment for each sample.

Accordingly, we consider implementing this thresholding method in a semi-supervised context, in which the true classes are known, but hidden, and consequently can be used to determine the accuracy of the clustering. Identifying target features in using this method can be considered equally as informative as discovering useful features in a supervised context. Thus, the advantage of this thresholding method is that it can be used in both
supervised and semi-supervised contexts, and that it allows for the direct comparison of results from both types of data.

2.1.2.2 Unsupervised Higher Criticism Threshold

The second thresholding technique is based on the notion of higher criticism thresholding. This technique has been fully developed in the supervised case, as explained in chapter 1. In our new method, we capitalize on the theoretical advantages of supervised higher criticism, and follow the arguments from [49, 89, 90] to translate the HCT functional for use in a supervised context into an unsupervised setting. This method works in a completely data-dependent manner and does not require any prior knowledge of the class or cluster assignments and, thus, can be used in an entirely unsupervised context.

Recall that the higher criticism threshold is generated as a “z-score for p-values”, where the p-values represent the level of importance of a feature for classification purposes. In our case, we generate p-values based on the $F$ test described above, and then follow a modified version of supervised HCT to find a natural cutoff in our ranking for including features.

In general, the unsupervised higher criticism thresholding method (UHC) works as follows:

1. For each feature $j$, determine a p-value $p(j)$ for the feature based on the value of the bimodal statistic $F(j)$.

2. Order the p-values from smallest to largest, such that $\pi(1) \leq \pi(2) \leq ... \leq \pi(p)$.

3. determine the value of the unsupervised higher criticism functional for each feature; i.e. calculate

$$UHC(j) = \frac{j/p - \pi(j)}{\sqrt{\max(\sqrt{n}(j/p - \pi(j)), 0) + j/p}}$$

4. Fix a level $\alpha_0$ such that $\alpha_0 \in (0, 1)$. Choose $UHC^* = \max_{1 \leq j \leq n_0}(UHC)$. 
5. The resulting threshold is given by \( t = F(j : UHC(j) = UHC^*) \).

The explanation behind this thresholding choice is fully explained in Chapter 4. For now, we assume that this particular threshold choice is useful in this context. We choose the parameter \( \alpha_0 = 0.1 \) to be consistent with the supervised analysis of higher criticism, particularly in rare-weak settings. However, the choice of \( \alpha_0 \) is somewhat arbitrary, and the success of the method is fairly resistant to any choice of \( \alpha_0 \) such that \( \alpha_0 < p/2 \), as explained in [22, 21, 23].

We consider a strategy to calculate the p-values to be used in this thresholding method based on the assumptions of the \( F \) statistic itself, namely that by construction it should follow an noncentral \( F \) distribution. We consider only the right tail of the distribution of the \( F \) statistic, because this tail will contain the features of interest.

By construction, the right tail of the \( F \) statistic follows a non-centered \( F \) distribution with \( df_1 = n - 1 \) and \( df_2 = n - 2v - 2 \) and non-centrality parameter \( 1/\lambda \), where \( v \) represents the number of splits that are generated in the calculation of \( \sigma^2_{bi} \). We thus calculate p-values for each statistic by considering the following:

\[
p(j) = P(F > F_{n-1,n-2v-2}(1/\lambda))
\]

The argument for this distribution choice is considered in Chapter 4. These p-values are then ranked based on significance, and a threshold for FS is found using the UHC method outlined above.

2.1.2.3 Modifications

Errors of this \( F \) test are known to occur in highly skewed and strongly leptokortic distributions, which lead to misleadingly high \( F \) values. [53] A slight modification can be added to the original \( F \) test, which includes a test for skewness as well as a test for kurtosis, and eliminates features with highly skewed or strongly leptokortic distributions. We choose not to explore the \( F \) test modifications here, and instead rely on the nonparametric modifi-
cations of the entire test as laid out in Chapter 3 to account for scenarios in which the bimodal $F$ test violates these assumptions.

### 2.1.3 Clustering Method - K-Means Clustering

While multiple clustering methods were tested, the clustering method that we focus on in step 6 is mainly k-means clustering. We found that k-means produces lower or equivalent errors compared with other methods, such as hierarchical or spectral clustering. Additionally, we chose to focus on k-means clustering because the method is widely used and accepted as a powerful and theoretically-advantageous clustering method.

### 2.2 Extensions to the Multi-Class Setting

The strongest advantage to this FS method is that it can be naturally extended to a multi-class scenario with very few modifications. In contrast, most existing feature selection methods are restricted to exhibiting good performance in the case of $k = 2$ defined clusters. Very few methods address situations with more than two clusters, and even fewer FS methods perform well in such settings. Our ranking method, however, can be extended to consider cases in which more than two classes are assumed to be present, i.e. for $k \geq 3$.

Our method considers a slight modification to the calculation of the $F$ statistic used for ranking in the case of more than two clusters. Recall that in a simple two-class scenario, the $F$ statistic compares the variance of the univariate distribution assuming the feature was unimodal, to the variance under the assumption that it comes from a bimodal mixture model. When multiple classes are considered, we thus compare the variance of the feature distribution based on the unimodal assumption to the minimum variance of the distribution under a *multimodal* assumption. To calculate the $F$ statistic in a multimodal setting, we compute $\sigma_{uni}^2$ as above, and compute $\sigma_{multi}^2$ in the following manner (mirroring the process of computing $\sigma_{bi}^2$):

- Generate $v$ splits, and then define $\{v_1, v_2, v_{k-1}\} \in V$; i.e. For $\omega \in \{1, 2, \ldots, k\}$,
determine \((\omega - 1)\) points at which to “split” the univariate distribution such that \(n_\omega\) points lie to the left of the split \(v_\omega\), with \(n_k = n - \sum_{\omega=1}^{k-1}(n_\omega)\).

- For each combination \(V\) of possible splits, find the mean and variance of each section of the univariate distribution, \(\mu_{n_\omega}(V)\) and \(\sigma^2_{n_\omega}(V)\), similar to the bimodal calculation above.

- For each combination \(V\), find the average variance \(\sigma^2(V) = \frac{1}{k} \sum_{\omega=1}^{k} \sigma^2_{n_\omega}(V)\).

- Then \(\sigma^2_{\text{multi}}(j) = \min_{\omega}(\sigma^2(V))\).

The resulting F test is given by the ratio:

\[
F = \frac{\sigma^2_{\text{uni}}}{\sigma^2_{\text{multi}}}
\]

The multi-class method works best in the case where \(k > 2\) and yet still small, i.e. \(k < 5\). Additionally, performance of this method varies directly based on sample size. From the construction of this statistic, it should be clear that this method is not ideal when \(k\) is large and \(n\) is small, as there will not be enough samples to accurately represent many clusters even in a marginal distribution.

What is interesting about this particular test, however, is that in most cases, the bimodal statistic is sufficient in detecting important features for clustering, even when the number of clusters is greater than two. We have found that there is very little difference in ranking of features when using the bimodal or the multi-modal F statistic, and that this difference tends to 0 as \(k\) increases for a constant \(n\). Additionally, in the case of many clusters, the multi-modal F statistic becomes cumbersome, and its computational burden starts to outweigh the benefits for using this measure versus the bimodal statistic. Therefore, while we consider the multi-modal F test useful for 3-4 clusters, we typically refer back to the bimodal statistic when \(k > 4\).
2.3 Simulation Study

We conducted a simulation study to test the performance of this new feature selection method based on a few parameters - the number of significant features (determined by $\beta$), the strength of the signal from significant features (determined by $\tau$), and the effect of a correlated covariance structure. In all simulations, we assumed that there were two classes of equal sizes (each cluster containing exactly 1/2 of the $n$ samples), with $n \approx 100$ and $p = 1 \times 10^4$. The data was generated in the following manner:

1. Set the parameters $(\beta, \tau)$ such that $\epsilon = p^{-\beta}$ is the desired sparsity, and $\tau$ is the desired signal strength.

2. Generate a $p$-length vector $b$, such that each element $b_i \sim \text{Bernoulli}(\epsilon)$. Set $\mu = b\sqrt{\tau/\sqrt{n}}$.

3. Generate a simulated $n$-length class/cluster vector $C$ such that each element $C_i \sim \text{Bernoulli}(1/2) - 1/2$. (This assumes $\delta = 1/2$, i.e. each cluster contains 1/2 of $n$ samples.)

4. Generate a $n \times p$ matrix $Z$ where each column $z_i \sim N(0, \omega)$, where $\omega$ represents the desired covariance structure.

5. Set working data matrix $X = C\mu' + Z$.

For each setting, the Hamming distance, measuring the fraction of misclassified samples, was recorded. The simulation was conducted for 30 times for each scenario, and results reported reflect the average proportion of misclassified samples, based on the Hamming distance measure, over 30 runs of the experiment. Also included is the average percentage or number of features kept in each scenario. Clustering was done using the classic k-means clustering method in each simulation.
Based on these results, it is clear that feature selection and data reduction minimizes classification errors in most cases, even in the presence of correlated noise. Errors are drastically minimized in the case of uncorrelated noise, particularly because the feature
selection is done in a univariate manner and works best under the assumption that noise in the data is uncorrelated. Nevertheless, the algorithm shows superior clustering performance compared with clustering on the entire dataset, even in the case of correlated noise. Additionally, the algorithm allows the data to cluster in a more efficient manner given the reduced size of the dataset.

We then chose to explore the relationship between the effectiveness of feature selection using the bimodal F test based on the parameters for signal strength (tau) and signal sparsity (beta). We compare the errors based on the fraction of misclassified samples in the case of k-means clustering with no FS implemented, and then when bimodal FS using the F test was applied. Comparing the errors in the following chart, we see that FS improves clustering errors, but provides the largest improvement in the case where features are drastically rare (i.e. \( \beta \) is large).

![Figure 2.5: Effect of Signal Rarity and Strength on Clustering Error](image)

We also compared the effects of signal strength and signal rarity on the performance of the parametric model, and benchmarked them against the performance of the IF-PCA algorithm discussed in chapter 1. We rename our method in the following graphics as “UHC-F” to indicate we have used the \( F \) test for ranking, and thresholding was done based on using the p-value from the \( F \) distribution and applying “unsupervised” higher criticism. The results show that our method produces datasets containing fewer features (with percentages shown in the legend) and are more accurate in almost all cases when
compared to IF-PCA, and show drastic improvement over the full clustering model. The
parametric model developed here performs similarly to IF-PCA, which is still a drastic
improvement over full clustering, but does so with closer to 2% of features retained (vs.
5% in IF-PCA).

Figure 2.6: Effect of Signal Strength on Clustering Error

![Figure 2.6: Effect of Signal Strength on Clustering Error](image)

Figure 2.7: Effect of Signal Sparsity on Clustering Error

![Figure 2.7: Effect of Signal Sparsity on Clustering Error](image)

2.3.1 Simulation - Multi-class Datasets

While there are many filter methods that have been developed for both supervised and
unsupervised feature selection, this method is one of the very few that have been developed
to handle only two cases. Furthermore, the application of this method in a multi-class setting retains its easy implementation and interpretability, and results in better clustering in rare and weak model scenarios.

An extended simulation study was used to determine the performance of the new method on multi-class datasets. The parameters that were assessed in performance were similar to those tested in the two-class case; namely, the number of significant features, the strength of the signal from significant features, the effect of sample size on appropriate feature selection, and the amount of separation between cluster means. In all simulations, we assumed that the classes were of equal sizes (each simulated dataset was made up of 3 class clusters, each containing exactly 1/3 of the n samples). The data was generated based on the following parameter values:

1. Total features (fixed): \( p = 1 \times 10^4 \).
2. Number of classes and/or clusters (fixed): \( k = 3 \).
3. Number of samples per class and/or cluster (variable): \( n_k \in \{20, 30, 40\} \).
4. Fraction of “important” features needed for cluster assignment (variable, determined by parameter \( \beta \)): \( .65 < \beta \leq .75 \).
5. Signal strength of important features (variable, determined by parameter \( \tau \): \( \tau \in \{10, 12, 14\} \)).
6. Distance between cluster means in “important” feature dimensions (variable, determined by specification of vector \( \mu \)).

For each scenario, the simulation was conducted for 30 times, and results reported reflect the average error (Hamming distance) over 30 runs of the experiment. Clustering was done using the classic k-means clustering method in each simulation. We compared the errors based on the Hamming distance (proportion of misclassified samples) using the full dataset and k-means clustering, to performing k-means clustering on a dataset that was
reduced based on the bimodal $F$ test. The percent of features that were kept for clustering was also recorded. Here, we see that using bimodal feature selection and the $F$ test leads to a reduction in clustering error and does so based on a significantly smaller dataset.

Figure 2.8: Effect of Number Impt Features on Bimodal FS Performance

Naturally, as the signal strength increases in the important features, both the reduced-clustering and full-clustering methods perform better. Similarly, increasing the percentage of important features also increases the performance of the full and reduced clustering
methods. However, regardless of the number of important features, results from these simulations indicate that the reduced dataset generated from bimodal feature selection reduces the error by at least 50% in all cases. The case where bimodal feature selection has the most significant advantage is where there are fewer important features with a stronger signal.

We also consider the effect of the class size for $k = 3$ clusters on the k-means clustering error. In the following figure, we fix the parameters for signal sparsity and strength, and consider the effect of increasing the class size. These results demonstrate that while the performance of k-means clustering depends on the sample size, the bimodal clustering performs with a steady error rate, indicating that its performance is resistant to sample size.

![Figure 2.9: Effect of Class Size on Bimodal FS Performance](image)

Changing the distance between cluster centers has a significant effect on clustering performance, in both the full k-means and reduced k-means cases. Clusters that have a larger amount of overlap create more of a harder clustering problem, while clusters that are farther away in the multi-dimensional space will be clearly defined and easily separable. Based on these results, there is a clear “sweet spot” for the bimodal feature selection method, in which using the reduced dataset results in a clustering that provides significantly fewer errors.
2.4 Case Study

In addition to testing this method on numerous simulated datasets, we also tested our bimodal feature selection method on standard cancer datasets, and compared our results with other methods. These are datasets that are commonly used in supervised feature selection discussions - they are datasets in which feature selection proves useful in enhancing class discovery. We strip the supervised classes from these datasets, and treat the data in a semi-supervised context as explained section 2.1.2. The two-class datasets that we have chosen, along with their corresponding original sources, are displayed in the following table.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Source</th>
<th>n</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leukemia (partial)</td>
<td>Golub et. al (1999)</td>
<td>38</td>
<td>7129</td>
</tr>
<tr>
<td>Leukemia (full)</td>
<td>Golub et. al (1999)</td>
<td>72</td>
<td>7129</td>
</tr>
<tr>
<td>Lung</td>
<td>Gordon et. al (2002)</td>
<td>181</td>
<td>12533</td>
</tr>
<tr>
<td>Prostate</td>
<td>Singh et. al (2002)</td>
<td>102</td>
<td>6033</td>
</tr>
</tbody>
</table>

An overview of the performance of our method is given in the following figure, where the fraction of misclassified samples is shown *after* bimodal feature selection was performed. Results using 2 clustering methods, spectral clustering and k-means clustering, are also
compared here. This demonstrates that using bimodal feature selection to select important features before clustering is implemented leads to small errors in most cases, when either spectral clustering or k-means clustering is used.

Figure 2.12: Real Datasets: Overview of Performance

<table>
<thead>
<tr>
<th>Dataset</th>
<th># samples</th>
<th># features</th>
<th># (%) retained</th>
<th>error (spectral)</th>
<th>error (k-means)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leukemia (partial)</td>
<td>38</td>
<td>7129</td>
<td>1257 (17.63)</td>
<td>4/38</td>
<td>0/38</td>
</tr>
<tr>
<td>Leukemia (full)</td>
<td>72</td>
<td>7129</td>
<td>236 (3.91)</td>
<td>7/72</td>
<td>7/72</td>
</tr>
<tr>
<td>Lung</td>
<td>181</td>
<td>12533</td>
<td>18 (0.14)</td>
<td>14/181</td>
<td>38/181</td>
</tr>
<tr>
<td>Colon</td>
<td>62</td>
<td>2000</td>
<td>90 (4.50)</td>
<td>7/62</td>
<td>8/62</td>
</tr>
<tr>
<td>Prostate</td>
<td>102</td>
<td>6033</td>
<td>40 (0.56)</td>
<td>41/102</td>
<td>32/102</td>
</tr>
</tbody>
</table>

To prove that our method was in fact performing better than the full clustering performance, we compared our error rates using a reduced dataset (generated from our bimodal feature selection algorithm) to the error rates (fraction of misclassified samples) for the clustering methods for the entire dataset. In every case, our results demonstrate the bimodal feature selection method and a reduced dataset minimized errors vs. clustering with all of the data. In this setting, we used k-means clustering to cluster the data in the reduced dataset setting.

Figure 2.13: Real Datasets: Full vs. Reduced Clustering Error Comparison

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Full dataset</th>
<th>Reduced Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Kmeans</td>
<td>Hier</td>
</tr>
<tr>
<td>Leukemia</td>
<td>20/72</td>
<td>26/72</td>
</tr>
<tr>
<td>Lung</td>
<td>22/181</td>
<td>32/181</td>
</tr>
<tr>
<td>Colon</td>
<td>23/62</td>
<td>24/62</td>
</tr>
<tr>
<td>Prostate</td>
<td>50/102</td>
<td>50/102</td>
</tr>
</tbody>
</table>

We also compared our results with another unsupervised feature selection method called IF-PCA, presented by Jin and Wang [49], which performs feature selection based on the Komolgorov-Smirnov ranking of features, and use HCT in the context of spectral clustering.
to generate their reduced dataset. Our results indicate that there are cases in which the bimodal F-test outperforms IF-PCA, and other cases in which IF-PCA performance is slightly superior.

![Figure 2.14: Real Datasets: IF-PCA vs Bimodal F test](image)

2.4.0.1 Performance on Real Multi-class Datasets

We also tested our bimodal feature selection method on standard cancer datasets that are common in a multi-class setting. Similar to the two-class case, we select datasets that are common in feature selection discussions, particularly in a multi-class setting. The list of datasets and their corresponding papers are given in the following table.

![Figure 2.15: Real Datasets: Multi-Class Case](image)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Source</th>
<th>K</th>
<th>n</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leukemia</td>
<td>AML/ALL</td>
<td>3</td>
<td>72</td>
<td>7129</td>
</tr>
<tr>
<td>Leukemia</td>
<td>Golub et. al (1999)</td>
<td>6</td>
<td>248</td>
<td>12625</td>
</tr>
<tr>
<td>Breast</td>
<td>DFHCC</td>
<td>3</td>
<td>115</td>
<td>54675</td>
</tr>
<tr>
<td>Breast</td>
<td>Natrajan et. al (2009)</td>
<td>3</td>
<td>53</td>
<td>48701</td>
</tr>
<tr>
<td>Breast</td>
<td>MAQC2</td>
<td>3</td>
<td>290</td>
<td>22283</td>
</tr>
<tr>
<td>Brain</td>
<td>Tumor</td>
<td>5</td>
<td>90</td>
<td>7129</td>
</tr>
</tbody>
</table>

An overview of the performance of our method is given in the following figure. In particular, the beauty of this method is that it is easily adaptable to the case where more than 2 classes exist. To our knowledge, other unsupervised methods such as IF-PCA which are particularly useful in 2-class cases, have not yet been extended to the situation where $k > 2$ (although the extension of IF-PCA is natural as well). As you can see, using bimodal
feature selection to select important features before clustering is implemented leads to small errors in most cases, and does so even with incorporating significantly fewer features than the entire dataset.

Figure 2.16: Bimodal FS Performance on Real Multi-class Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of Classes</th>
<th>Number of Samples</th>
<th>Number of Features</th>
<th>Kmeans Error (%)</th>
<th>Bimodal Kmeans Error (%)</th>
<th>Number Features (%) Retained</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lukemia</td>
<td>AML/ALL</td>
<td>3</td>
<td>72</td>
<td>28 (39%)</td>
<td>13 (18%)</td>
<td>40 (.56%)</td>
</tr>
<tr>
<td>Leukemia</td>
<td>AML/ALL</td>
<td>6</td>
<td>248</td>
<td>12625 (188%)</td>
<td>80 (32%)</td>
<td>94 (.74%)</td>
</tr>
<tr>
<td>Breast</td>
<td>DFHCC</td>
<td>3</td>
<td>115</td>
<td>54675 (37%)</td>
<td>21 (18%)</td>
<td>850 (1.6%)</td>
</tr>
<tr>
<td>Breast</td>
<td>IRB</td>
<td>3</td>
<td>129</td>
<td>54675 (35%)</td>
<td>34 (26%)</td>
<td>200 (0.37%)</td>
</tr>
<tr>
<td>Breast</td>
<td>DFHCC2</td>
<td>3</td>
<td>84</td>
<td>54675 (5%)</td>
<td>8 (9.5%)</td>
<td>1200 (2.2%)</td>
</tr>
<tr>
<td>Breast</td>
<td>HLP</td>
<td>3</td>
<td>53</td>
<td>48701 (17%)</td>
<td>9 (17%)</td>
<td>400 (.82%)</td>
</tr>
<tr>
<td>Breast</td>
<td>MAQC2</td>
<td>3</td>
<td>230</td>
<td>22283 (117%)</td>
<td>58 (25%)</td>
<td>50 (.22%)</td>
</tr>
<tr>
<td>Brain</td>
<td>Tumor</td>
<td>5</td>
<td>90</td>
<td>7129 (49%)</td>
<td>42 (47%)</td>
<td>1000 (14%)</td>
</tr>
</tbody>
</table>

2.5 Conclusions

Parametric bimodal FS is a novel method that takes advantage of the underlying distribution of features that will be important in clustering. The model is based on the concept that important features will be distributed differently than non-informative features across samples. Exploiting the difference in this behavior demonstrates that this model provides an effective way to search for target or marker features that are crucial in cluster identification for samples.

Simulated and real-world situations demonstrate that the new method performs better than existing clustering methods where no feature selection is considered, and performs at a similar level compared to other feature selection methods. Furthermore, this method performs well on significantly fewer features than are contained in the original dataset - over 95% of the features are removed before clustering.

We have also seen that this method performs well even in the presence of correlated noise. This result indicates the validity of implementing the assumption of uncorrelated noise without loss of generality and without loss of performance. Moreover, in simulated
studies involving more than two clusters, this method depends heavily on the signal strength and univariate separation between clusters.

Real-world data scenarios present the most promising aspect of these results. Here, we see that datasets are reduced significantly and result in minimal clustering errors. This suggests that this bimodal F test will be useful in identifying target or drives genes that are specifically related to the proliferation of certain cancers and their subtypes in the context of gene-expression data. We have found that while the method works best in the case of two-class scenarios, it still performs useful feature selection even in the case of multi-class settings. In multi-class settings, however, the performance of the method seems to decrease as the number of clusters increases.

It is perhaps interesting to note that while the normality assumption might initially seem to be too restrictive, this method seems to perform well in many scenarios. However, it does seem that this restriction could be causing larger errors in the case where these distributions are highly skewed. Accordingly, we develop a similar method that relaxes this assumption in Chapter 3.
Chapter 3

Nonparametric Bimodal Feature Selection

In many cases, a few assumptions must be made in order to perform feature selection. This is because many filter methods require the use of a parametric model. As discussed in Chapter 2, the most common assumption that is used is one of normally-distributed noise, which is a typical and reasonable assumption. However, because it is always advantageous to consider statistical methods that require the fewest assumptions possible, in this chapter we consider a nonparametric version of the bimodal feature selection model developed in Chapter 2.

Our approach selects features based on the nature of their univariate distributions, which are estimated in a nonparametric setting. We show that the nonparametric FS method improves the performance of the k-means clustering method dramatically in the case of high dimensional data, in both simulated and actual data scenarios. We also compare our results from this method to other parametric FS models, and see that in some cases this new method outperforms existing parametric models, particularly in the case where the parametric assumptions are violated.

3.1 Model Definition

There are a few common ways to model a distribution in a nonparametric environment. One of the most common nonparametric methods is known as kernel density estimation (KDE). In our model, we estimate the univariate distribution of each feature using KDE, and then determine the number of modes in the distribution based on this estimate. The
smoothness of the distribution is crucial in determining the number of modes identified in each case. In fact, the smoothing parameter $h$ is specifically designed as a parameter that can be tuned to accentuate or eliminate “bumps” in the graph of a particular function. Therefore, determining how to set and use the parameter $h$ to investigate the number of modes present in a distribution is crucial in the context of this model. This notion is discussed in detail in a subsequent section of this chapter.

Once the smoothness parameter $h$ has been set, modal identification using the kernel-smoothed function becomes fairly easy. After identifying modes, feature selection is performed in a univariate framework, based on the understanding that features carrying little pertinent clustering information will be distributed unimodally, while critically useful features will follow a bimodal, or in some cases multi-modal distribution, depending on the number of classes or clusters present. This is similar to the parametric model presented in Chapter 2. However, due to the flexible nature of the density estimate, this method has the added advantage of capturing and accurately identifying modes in highly skewed distributions. It is also particularly useful for detecting modes where the cluster sizes are uneven, or when significantly fewer samples contribute to one modal value versus another.

In general, the steps to performing feature selection in a nonparametric setting are as follows:

1. For each feature $j$, perform kernel density estimation (KDE) to generate a density estimate $\hat{f}_n(x(j))$ for each feature distribution based on $n$ data values. Note that in performing KDE, an ideal bandwidth, $h(j)$, and a smoothing kernel, $K$, must be selected. In our case, the kernel used will be what is known as the biweight kernel. The bandwidth $h(j)$ is selected based on the common “rule of thumb” estimate as in [Silverman].

2. Use the density estimate to identify the number of significant modes present in the distribution.

3. Rank features by importance, with features exhibiting strongly bimodal distributions
ranking higher in classification importance.

4. Select features based on resulting ranking, eliminating features with unimodal distributions and keeping only those that exhibit bimodal tendencies, based on a thresholding measure.

5. Cluster data based on the reduced set of features using k-means clustering, and use the results to assign data values to clusters.

A detailed explanation of how these steps are executed follows.

3.1.1 Kernel Density Estimation

Consider the distributions of samples in a given feature vector \( j \), \( \{x_1(j), x_2(j), \ldots, x_n(j)\} \). When considering each feature separately, this should be a one-dimensional list of feature values, with one value for each sample.

KDE is used to estimate and smooth the data density (denoted \( \hat{f}_n \)) by centering a kernel function \( K(x) \) at each data point \( x_i(j) \) such that:

\[
\hat{f}_n(x(j)) = \frac{1}{nh} \sum_{i=1}^{n} K\left( \frac{x(j) - x_i(j)}{h} \right)
\] (3.1)

Note that the this estimate depends on the kernel function \( K \) and the bandwidth parameter \( h \).

In our case, our choice of kernel \( K \) the biweight kernel, for two reasons. First, the biweight kernel is compactly supported, ensuring that only local information is used surrounding each data point (a characteristic that is important when constructing a density estimate when \( n \) is relatively small). Second, due to the compact support, the biweight kernel is also computationally less demanding. The biweight kernel function is given by the formula:

\[
K(x_j) = \frac{15}{16} (1 - u^2)^2 \mathbb{1}_{|u| \\leq 1}
\] (3.2)

The support of the kernel is \(-1 \leq u \leq 1\), which implies that \( x_{ij} - h \leq x_j \leq x_{ij} + h \), and,
thus, the support of the kernel is also determined by the chosen bandwidth, \( h \). Thus, the choice of \( h \) is critical.

To determine the value of \( h \), our smoothing parameter, we ideally choose the bandwidth that minimizes the asymptotic mean integrated squared error, which is a linear combination of the squared-bias and the variance. This is a common criterion for bandwidth selection, and it is the value \( h \) that minimizes the function:

\[
AMISE(\hat{f}_n(x(j))) = \frac{5}{7n}h + \frac{1}{28}h^4\|f''\|_2^2
\]  

(3.3)

In lieu of this calculation, we choose an estimate for the optimal bandwidth based on the “rule of thumb” presented in [76]:

\[
\hat{h} = \hat{\sigma}(j)C_v(K)n^{-1/(2v+1)}
\]  

(3.4)

Where \( \sigma \) is the sample standard deviation, \( C_v(K) \) is a constant dependent on the choice of kernel (in the case of a biweight kernel, \( C_v(K) = 2.78 \)), and \( v \) is the order of the kernel (in our case, \( v = 2 \)). This particular estimate is known to behave well as a bandwidth estimate in a variety of situations, and is thus a common choice in practice. While a comparison of bandwidth selection was briefly considered, most analysis was done using the standard bandwidth choice as described above.

### 3.1.1.1 Bandwidth Comparison

We compared bandwidth selections by simulation. Note that while the bandwidth selection can alter KDE in the case of highly skewed distributions, it is shown to have little effect on more regularized distributions. We considered the performance of FS using the rule of thumb bandwidth described above, as well as a bandwidth defined by medians, namely:

\[
\hat{h} = \left(\frac{4}{3n}\right)^{1/5}\hat{\sigma}_p
\]
where
\[ \tilde{\sigma}_p = \frac{\text{median}(|X[i, j] - \text{median}(X[i, j])|)}{.6745}. \]

Note the only major difference between this construction of \( h \) and the bandwidth based on the rule-of-thumb measure is the construction of the estimate of \( \sigma \). However, in the case of feature selection, it seems that our models are somewhat resistant to the choice of our estimate for \( \sigma \) and, therefore, are resistant to the choice of bandwidth. Performance on both simulated and real world datasets is rarely changes based on the bandwidth definition. Accordingly, we choose the typical bandwidth definition defined in [76] for use in our model, although we recognize other bandwidth definitions could be considered.

### 3.1.2 Ranking and Thresholding

Two ranking and thresholding methods were considered in the case of nonparametric feature selection, one based on semi-supervised data (in which true classes are known, but hidden) and one data-dependent threshold that is applicable in an entirely unsupervised context. We explain both methods in detail below.

#### 3.1.2.1 Thresholding Based on Modal Intensity and Separation

In a semi-supervised context, features are ranked such that those features with strongly bimodal distributions will rise to the top, while features with unimodal distributions are treated as noisy non-informational features and therefore will be discarded before classification or clustering. This ranking is performed by calculating the number of significant modes in the data, and subsequently measuring the distance between identified modes. The explanation of this process follows.

Once the density estimate is found using KDE, we normalize each distribution to have a maximum of 1. We then consider all modes greater than 0.15 to be potentially significant, and treat them as such. This process is known to be effective, for reasons similar to those presented in [33].
To find potential modes, we determine any local peaks and spikes that are present in the data; a data value $y_i$ is tagged as a potential mode $m$ if $y_{i-1} < y_i$ and $y_{i-1} < y_i$. Once potential modes have been tagged, we determine which of the local modes should be considered global modes based on their neighborhood values. To do so, we measure the strength/frequency of the neighborhood surrounding each mode (the value of $y_i$ itself), and identify points where $y_i > 0.15$. We also determine the Euclidean distance between potential modes (the absolute distance $|m_a - m_b|$ for all modes $a$ and $b$ that have been identified). Features with more than one substantial mode, based on this criteria, were considered “bimodal” vs unimodal, and were classified accordingly. Ranking was first done based on the number of identified modes, and then subsequently on the distance between modes. Reduced feature sets for clustering were found by implementing a threshold, with the final feature set chosen as the set that resulted in minimum clustering error using k-means clustering. This characteristic is easily determined in a semi-supervised context where the true classes are fully known.

### 3.1.2.2 Alternative Thresholding using Bandwidth HC

In addition to a ranking and thresholding method based on modal location and intensity, we also consider an alternative method to incorporate the choice of bandwidth and capitalize once again on the idea of unsupervised higher criticism thresholding. We exploit the concept that features which tend to exhibit bimodal distributions will require a larger bandwidth in order to obtain a “smoother” estimate. Therefore, one would naturally expect features with the largest estimated bandwidth (i.e., those features which required the largest smoothing parameter) to be exactly the features that will benefit the most from clustering.

We construct this thresholding method in the following manner:

1. For each feature $j$, generate a density estimate $\hat{f}_n(x(j))$, and record the bandwidth estimate $\hat{h}(j)$ based on the estimate generated.
2. Standardize the bandwidth values based on the arguments for large-testing problems presented in [26]; namely determine the “z-score” of each estimated bandwidth $\hat{h}(j)$ as follows:

$$z(j) = \frac{\hat{h}(j) - \bar{h}}{\sigma(\hat{h})}$$

where $\bar{h}$ and $\sigma(\hat{h})$ are the empirical mean and standard deviation of the distribution of $\hat{h}$, respectively.

3. Find a right-tailed p-value associated with the normalized bandwidth; i.e. determine

$$p(j) = P(Z > z(j))$$

for $j = 1, \ldots, p$.

4. Use the sorted p-value $\pi(j)$ to determine a proper threshold based on the unsupervised higher criticism thresholding functional (UHC), as outlined in Chapter 2.

For reference, we will refer to this method of generating p-values as the “KDE variance” method. This method is extremely effective in finding and selecting useful features for clustering. Because this model incorporates higher criticism, it also has many useful theoretical properties associated with thresholding selection- it captures precisely the significant features and determines a threshold that approaches an ideal asymptotic threshold. We explore the theoretical reasoning behind this method in Chapter 4.

Finding a p-value in the above fashion works well in our models because of the assumed behavior of our model - namely that useful features are rare and weak, and that the noise follows a common unknown zero-mean distribution. When this assumption is known to be violated, particularly when the variances are skewed or are vastly different across dimensions, construction of the p-value in the method above is no longer valid. In this case, we find a p-value by replacing steps 1-3 above with the following steps:

1*. From the data $x_1(j), x_2(j), \ldots, x_n(j)$, find $h_{crit}$, the minimal bandwidth estimate for which the smoothed distribution of $x(j)$ is unimodal, and use it to determine $\hat{f}_{crit}$, the extreme unimodal density function of the data with bandwidth $h_{crit}$.
2*. Re-sample $x^{*}_{1}(j), x^{*}_{2}(j), ..., x^{*}_{n}(j)$ from $\hat{f}_{crit}$, and determine the number of modes based on $h_{crit}$. Repeat this step many ($B$) times.

3*. Find an associated p-value based on the empirical proportion of samples that yield multimodality.

Details that relate to the determination of $h_{crit}$ in step 1* and determining multimodality based on $h_{crit}$ in step 3* follow the discussion in [77] and are outlined in Chapter 4. It is also possible to use the determination of $h_{crit}$ itself to calculate a p-value for each feature $j$: we determine $h_{crit}(j)$ for each feature and then use the distribution of the p estimates of $h_{crit}(j)$ to find an approximate p-value. This method works well in establishing a p-value, although the threshold using this method is more ambiguous - higher criticism thresholding does not work appropriately in this case. We instead choose the features that seem to be moderately significant, with a threshold between 0.05 and 0.1. For reference, we will refer to this method of generating p-values as the “KDE critical bandwidth” method.

3.1.3 Clustering Method - K-Means Clustering

The clustering method that we focus on here again is k-means clustering. The reasoning for this selection is similar to the reasoning outlined in the parametric method, as explained in Chapter 2.

3.2 Extension of Bimodal FS to Multi-Class Clustering Problems

The nonparametric bimodal FS method described here is capable of searching for any number of clusters (so long as they are well-defined), and does so consistently with powerful performance. This method can also be applied in a multi-class setting and retains its simplicity and interpretability. Most importantly, this new FS method results in better clustering, particularly in rare and weak model scenarios where many other methods fail.

In the case of UHC in completely unsupervised settings, no modifications are made when it is applied to multi-class settings. This is because the process is only dependent on
the value of the bandwidth estimate for each feature $\hat{h}(j)$. This is a significant advantage of this method - it is easily applied to situations where more than two clusters are known to exist.

We have also developed a non-parametric multi-modal feature selection model in a semi-supervised realm, similar to the above two-class discussion with the following modification: we locate the number of modes in each density estimate using the modal identification procedure as outlined above, ranking features with the ideal number of modes ($\geq 2$) as most important, and unimodal feature distributions as least important. Modal separation is then considered based on the difference between furthest modes in the univariate distribution.

3.3 Simulation Study

We performed a simulation study to determine the performance of the new nonparametric feature selection algorithm. A few specific parameters of interest were determined - the number of significant features (determined by $\beta$), the strength of the signal from significant features (determined by $\tau$), and the effect of the distribution of noise in the data. In all simulations, we assumed that there were two classes of equal size, with each cluster containing exactly 1/2 of the $n$ samples and with $n \approx 100$ and $p = 1 \times 10^4$. The data was generated in the following manner:

1. Set the parameters $(\beta, \tau)$ such that $\epsilon = p^{-\beta}$ is the desired sparsity, and $\tau$ is the desired signal strength.

2. Generate a p-length vector $b$, such that each element $b_i \sim Bernoulli(\epsilon)$. Set $\mu = b\sqrt{\tau/\sqrt{n}}$.

3. Generate a simulated n-length class/cluster vector $C$ such that each element $C_i \sim Bernoulli(1/2) - 1/2$. (This assumes $\delta = 1/2$, i.e., each cluster contains 1/2 of $n$ samples.)

4. Generate a nxp matrix $Z$ where the distribution of each column $z_i$ represents either
Gaussian or Non-Gaussian noise structure. In the case of Gaussian noise, \( z_i \sim N(0, 1) \). For non-gaussian noise, we used a highly skewed and translated form of the beta distribution \( z_i \sim \text{Beta}(\alpha = 1, \beta = 5) \).

5. Set working data matrix \( X = C\mu' + Z \).

For each scenario, the simulation was conducted 30 times, and results reported reflect the average error (Hamming distance) over 30 runs of the experiment. The p-values used for feature ranking were generated using the KDE-variance method, and clustering was performed using the classic k-means clustering method in each simulation.

Figure 3.1: Gaussian Noise

| \( \beta = 0.6 \) | \( \tau = 6 \) | 0.4259 | 0.0854 | 0.2661 |
| \( \tau = 10 \) | 0.2460 | 0.0411 | 0.3852 |
| \( \tau = 14 \) | 0.1803 | 0.0159 | 0.4104 |

| \( \beta = 0.65 \) | \( \tau = 6 \) | 0.4210 | 0.1142 | 0.3194 |
| \( \tau = 10 \) | 0.3738 | 0.0528 | 0.2193 |
| \( \tau = 14 \) | 0.2793 | 0.0366 | 0.2176 |

| \( \beta = 0.7 \) | \( \tau = 6 \) | 0.4395 | 0.1900 | 0.2135 |
| \( \tau = 10 \) | 0.4039 | 0.3233 | 0.1859 |
| \( \tau = 14 \) | 0.3786 | 0.0744 | 0.1426 |

Figure 3.2: Non-Gaussian Noise

| \( \beta = 0.6 \) | \( \tau = 6 \) | 0.4401 | 0.1126 | 0.4560 |
| \( \tau = 10 \) | 0.3282 | 0.0570 | 0.2594 |
| \( \tau = 14 \) | 0.2696 | 0.0298 | 0.0663 |

| \( \beta = 0.65 \) | \( \tau = 6 \) | 0.4398 | 0.1641 | 0.4684 |
| \( \tau = 10 \) | 0.4074 | 0.0822 | 0.3720 |
| \( \tau = 14 \) | 0.3505 | 0.0482 | 0.2827 |

| \( \beta = 0.7 \) | \( \tau = 6 \) | 0.4440 | 0.2184 | 0.3120 |
| \( \tau = 10 \) | 0.4505 | 0.1382 | 0.2124 |
| \( \tau = 14 \) | 0.4061 | 0.0500 | 0.0794 |
Based on these results, it is clear that feature selection and data reduction minimizes classification errors when compared to clustering on the full dataset. Reduced datasets constructed using nonparametric feature selection all contain less than 1% of their original dimensions, and the resulting clusterings are at least twice as effective as clustering on the entire dataset. Errors are drastically reduced in the case of non-Gaussian noise, which shows the advantages of using a nonparametric method to model univariate distributions in the case of complicated noise structures.

When comparing the performance of the nonparametric method to others in the case of non-Gaussian noise, the advantage becomes even more clear - the nonparametric method clearly performs equally well regardless of the distribution of the noise in the data, which shows its flexibility in application. We also looked at the effect of the number of classes on performance. While it seems as if IF-PCA struggles with a larger number of clusters, performance of the methods outlined in Chapters 2 and 3 remain consistent, regardless of the number of clusters assumed in the underlying dataset. In these two simulated studies, we use the KDE critical bandwidth construction of the p-value with a threshold of 0.07 as a cutoff for the nonparametric procedure. In these two figures, nonparametric bimodal FS refers to this method.

![Figure 3.3: Non-Gaussian Noise Comparison of Methods](image-url)
3.4 Case Study

Our method was tested on a number of actual cancer datasets that are widely used for comparison in the field, as first introduced in Chapter 2. In most cases, results from clustering on the reduced dataset after nonparametric bimodal feature selection was performed led to a decrease in clustering error compared to clustering that was performed using all features. In each study, we used the KDE variance method for our feature ranking, and found a threshold based on unsupervised higher criticism. Clustering was done using k-means clustering. In this section, nonparametric bimodal FS refers to this method.

Results were also compared to the parametric version of bimodal feature selection as presented in Chapter 2, and the IF-PCA feature selection method. Results show that para-
metric models are advantageous in most cases. However, in the instances where parametric feature selection fails, however, the nonparametric method shows increased improvement in clustering.

Figure 3.6: Real Datasets: HC vs Bimodal(parametric) vs Bimodal(nonparametric)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>IF-PCA</th>
<th>Bimodal (F)</th>
<th>Bimodal (KDE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leukemia</td>
<td>1/72</td>
<td>7/72</td>
<td>20/72</td>
</tr>
<tr>
<td>Lung</td>
<td>0/181</td>
<td>14/181</td>
<td>1/181</td>
</tr>
<tr>
<td>Colon</td>
<td>19/62</td>
<td>7/62</td>
<td>24/62</td>
</tr>
<tr>
<td>Prostate</td>
<td>41/102</td>
<td>32/102</td>
<td>26/102</td>
</tr>
</tbody>
</table>

The nonparametric FS method was also tested in the multi-class datasets first seen in Chapter 2; results are given in the following table.

Figure 3.7: Real Datasets: HC vs Bimodal(parametric) vs Bimodal(nonparametric)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>K</th>
<th>Full Error</th>
<th>Bimodal (F) Error</th>
<th>% Retained Features</th>
<th>Bimodal (KDE) Error</th>
<th>% Retained features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leukemia AML/ALL</td>
<td>3</td>
<td>28/72</td>
<td>13/72</td>
<td>40 (0.56)</td>
<td>23/72</td>
<td>1628 (22)</td>
</tr>
<tr>
<td>Leukemia AML/ALL</td>
<td>6</td>
<td>188/248</td>
<td>80/248</td>
<td>94 (0.74)</td>
<td>33/248</td>
<td>93 (0.74)</td>
</tr>
<tr>
<td>Breast   DHFCC</td>
<td>3</td>
<td>37/115</td>
<td>21/115</td>
<td>850 (1.60)</td>
<td>21/115</td>
<td>365 (0.67)</td>
</tr>
<tr>
<td>Breast   HLP</td>
<td>3</td>
<td>17/53</td>
<td>9/53</td>
<td>400 (0.82)</td>
<td>9/53</td>
<td>85 (0.17)</td>
</tr>
<tr>
<td>Breast   MAQC2</td>
<td>3</td>
<td>117/230</td>
<td>58/230</td>
<td>50 (0.22)</td>
<td>81/230</td>
<td>140 (0.66)</td>
</tr>
<tr>
<td>Brain    Tumor</td>
<td>5</td>
<td>49/90</td>
<td>42/90</td>
<td>1000 (34)</td>
<td>30/90</td>
<td>87 (1.20)</td>
</tr>
</tbody>
</table>

From these results, it is clear that both bimodal FS methods outperform clustering on the full dataset; there are also datasets where nonparametric FS outperforms the bimodal $F$ test as given in Chapter 2.

3.5 Conclusions

The nonparametric bimodal FS technique, similar to the parametric model, is a novel method that ranks features for clustering importance based on the shape of their underlying distribution. The model presented in this chapter is more flexible than the previous model,
as it relies on fewer distribution assumptions for the data. The nonparametric model allows for quick and useful investigation into identifying features that carry significant clustering information, and naturally separates them from seemingly insignificant features.

This method outperforms the parametric model in both simulated and real-world situations, because of its flexibility. Moreover, this method provides superior performance to the parametric model in the case of highly skewed underlying distributions, and clustering problems with significantly uneven class sizes. It is clear that this method, like the parametric method, also significantly reduces data dimension by removing over 95% of the features before clustering, while still maintaining or even improving clustering accuracy. In the presence of non-Gaussian noise, the performance of the model remains unaffected. In fact, results indicate this method works extremely well in the case of non-Gaussian noise.

Real-world results also tend to favor the nonparametric model over the parametric method in some cases. It is interesting to note, however, that in cases where the nonparametric model does not perform well, the parametric model significantly reduces data dimension and clustering error. This indicates that these two methods are somewhat complementary, and that together these methods provide a more wholistic approach to the feature selection problem. This method also performs exceptionally well when applied to multi-class datasets, which are notoriously tricky to divide into multiple subgroups. These are precisely the situations where most other methods fail.
Chapter 4

Theoretical Asymptotic Performance

In this chapter, we discuss the theoretical concepts underlying the models developed in Chapters 2 and 3. We begin with a brief discussion of the k-means clustering problem, and the challenges of k-means clustering in the case of large datasets with $p \gg n$. The conversation is extended to discuss why feature selection is useful in the context of the k-means clustering algorithm. We argue that the FS methods outlined in Chapters 2 and 3 are useful in minimizing k-means clustering errors in parametric and nonparametric settings. We discuss the two thresholding methods in detail, outlining the reasoning behind the semi-supervised thresholding methods and arguing the theoretical advantages to using unsupervised higher criticism thresholding (UHC) in a completely unsupervised context. Finally, we explore the properties of using UHC in rare-weak scenarios, and how this threshold tends toward the ideal thresholding choice within asymptotic rare-weak unsupervised settings.

4.1 K-Means Clustering and the Data Glut

We now discuss the general principles behind k-means clustering. The general objective of k-means clustering can be constructed as follows, as developed in [43]: consider the points $\{x_1, x_2, ... x_n\} \in \mathbb{R}^p$. Our goal is to generate a k-partition $C = \{C_1, C_2, ... C_k\}$ such that the partition $C$ covers the parameter space in $p$ dimensions. Define $\mu_k$ to be the centroid of each part of the partition:

$$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} X_i$$  \hspace{1cm} (4.1)
The k-means clustering method looks to find the optimal k-partition \( C^* = \{C^*_1, C^*_2, ... C^*_k\} \) that minimizes the objective function:

\[
J_K = \sum_{i=1}^{n} ||x_i - \mu_k(x_i)||_2^2 = \sum_{k=1}^{K} \sum_{i \in C_k} ||(x_i - \mu_k)||_2^2
\]  

(4.2)

Here, \( \mu_k \in \mathbb{R}^p \) represents the \( p \)-dimensional cluster mean vector, and \( \cdots ||_2 \) represents the standard \( L^2 \) norm. We define our optimal k-partition \( C^* \) as:

\[
C^* = \arg \min_C (J_K)
\]  

(4.3)

Iterating between finding cluster means based on current cluster assignments, and then rearranging clusters to minimize total cluster variance results in each cluster \( C_k \) being well-defined by its cluster center, \( \mu_k \), which varies until a minimum is reached at convergence. In other words, technically we have:

\[
\mu_k(i) = \arg \min_{\bar{x}_k} \sum_{i \in C_k} ||(x_i - \bar{x}_k)||_2^2
\]

and:

\[
C_k(i) = \arg \min_{1 \leq k \leq K} ||(x_i - \bar{x}_k)||_2^2
\]

Thus, we combine these criteria and iterate through these steps until clusters stabilize, i.e., until convergence is reached. The ideal cluster partition \( C^* \) is then found by optimizing the following function:

\[
C_K^* = \arg \min_{C_1, \ldots, C_k, \{\bar{x}_k\}_1^k} \sum_{k=1}^{K} \sum_{i \in C_k} ||(x_i - \bar{x}_k)||_2^2.
\]

An equivalent form of the objective function is introduced by replacing the \( L^2 \) norm by a sum of squared Euclidean distance over \( p \) dimensions:

\[
J_K = \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} (x_i(j) - \bar{x}_k(j))^2
\]  

(4.4)
And similarly, the ideal cluster set will be given by:

\[ C^* = \min_{C \{\bar{x}_k\}_k} \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} (x_i(j) - \bar{x}_k(j))^2 \]  

(4.5)

To simplify, this algorithm can also be written in matrix form. First, we define the matrix \( X_{n \times p} \) such that each data point \( x_i \) is the \( i \)th row of \( X \). Then we define a clustering or k-partition \( C_k \) by creating a matrix \( C_{n \times k} \) so that the \( j \)th column represents the \( j \)th cluster, and the \( i \)th row represents the cluster assignment of data point \( x_i \). In other words, define \( C \) as a matrix with orthonormal columns containing \(|C_k|\) nonzero elements (indicating the points included in cluster \( C_k \)), such that:

\[
C_{ij} = \begin{cases} 
1 \sqrt{|C_k|} & \text{if } x_i \in C_k \\
0 & \text{otherwise}
\end{cases}
\]

Using the matrix definition, we can re-define the objective function in matrix form to be the following:

\[ J_K = ||X - CC^T X||_2^2 \]  

(4.6)

Note that the value \( CC^T X \) corresponds precisely to the appropriate mean value for cluster \( k \), \( \mu_k(x_i) \), which allows this equivalent matrix formation of the k-means objective function. Thus, or resulting optimal clustering \( C^* \) is the cluster assignment matrix that satisfies:

\[ C^* = \arg \min_{C \in \mathbb{R}^{n \times k}} ||X - CC^T X||_2^2 \]  

(4.7)

In order to determine a clustering using k-means, we minimize the objective function, which results in a clustering where each data point is assigned to a cluster based on the squared Euclidean distance from its assigned cluster mean. The complexity of this process is of order \( o(np^2ik) \), where \( p = \# \) dimensions, \( k = \# \) clusters (assumed to be either known or determined in advance of clustering), \( n = \# \) data points, and \( i = \# \) iterations. \[45\]
As \( p \to \infty \), this algorithm will have long computational run times, even if accuracy is not affected. Additionally, the features that contain no clustering information will add noise and, therefore, could lead to suboptimal performance of the k-means clustering algorithm, as we discuss in the next section. Accordingly, FS should be strongly considered as a useful solution to high dimensionality in k-means.

### 4.1.0.1 The Usefulness of Feature Selection in K-Means Clustering

Suppose it is known that an ideal clustering exists in \( r \) dimensions; it is a well-known problem that adding any additional dimensions (\( s \) dimensions which are pure noise) will lead to a degenerating objective function, and consequently a sub-optimal clustering. This argument proceeds as follows:

Consider a clustering in \( r \)-dimensional space; \( C^*_r = (C^r_1, C^r_2) \) which is ideal in \( r \) dimensions. That is:

\[
C^*_r = \arg \min_{C, \{m_k\}_1^K} \sum_{k=1}^K \sum_{i \in C_k} \sum_{j=1}^r (x_i(j) - \bar{x}_k(j))^2 = \arg \min_{C, \{m_k\}_1^K} J^r_K
\]  \hspace{1cm} (4.8)

Suppose we add \( s \) new dimensions (such that \( r + s = p \)), which are all known to be noisy dimensions and therefore have no significance for clustering. We claim that as the number of non-signal variables \( s \) becomes sufficiently large (as \( s \to \infty \)), the resulting clustering becomes random. Specifically, the distribution of clusterings becomes random, in that for any two data points \( x_1 \) and \( x_2 \), the probability of being in the same cluster becomes equal to any other pair (i.e., tends to \( 1/k \)).

Our new clustering in \( p \) dimensions will minimize the following objective function:
If we assume that all informative features have a zero-mean distribution, \( \bar{x}_{jk} = 0 \) and therefore the objective function simplifies to:

\[
J_{kp}^* = J_{rk}^* + C
\]  

The last added term \( C \) will be such that \( C \geq 0 \), by construction. Assuming these features are unimportant, we can treat the value of \( C \) as a constant; as the number of unimportant features grows, however, the entire objective function will monotonically increase with increasing dimension. Therefore, minimizing \( J_{kp}^* \) is dependent solely on minimizing \( J_{rk}^* \).

This implies that not only will \( C_r^* \) provide an \( r \)-dimensional clustering solution for the data, but this \( r \)-dimensional solution will provide a similar, if not better clustering for the \( p \)-dimensional problem with added noise, since the information regarding the true cluster separation is contained solely within the \( r \) feature dimensions.

We consider a more precise argument for why this must be true, and develop our argument in the following theorem:

**Theorem 1.** Given a collection of ordered variables \( x_1, x_2, \ldots \), any continuous invariant clustering method \( \chi \) benefits from variable selection, in that with sufficiently many noise variables, clustering is improved if some noise variables are eliminated. More specifically, in the above model, if sufficiently many irrelevant (noise) variables are added, any clustering
**algorithm becomes random with probability 1.**

Without loss of generality we will assume that the data matrix $X$ has the property that all data matrices sufficiently near it have the same clustering. We divide the matrix $X = [X_1X_2]$ into two matrices. The first is an $n \times r$ matrix whose rows consist of the original dataset with $r$ important features. The second matrix $X_2$ has rows containing $s$ entries that consist only of the additional noise features.

We standardize the data vectors so that just the vectors in $X_2$ have a standard normal distribution, i.e., $x_{i2} \to \frac{1}{\sqrt{s}} x_{i2}$. As $s \to \infty$, the vectors in $X_2$ maintain a $N(0, I_s)$ distribution, while the vectors in $X_1$ (with a fixed number of entries) converge to 0 in distribution. We restrict the data vectors (which without loss are assumed linearly independent) to the subspace $S = \text{span}\{x_i\}_{i=1}^n$. We note that in this (now fixed) dimensional subspace, the joint distribution of the data vectors approaches a standard normal joint distribution, i.e., the distribution of the majority $s$ components. It follows easily from this (since clustering in this subspace is the same as in the full space) that:

$$P(C(x_i) = C(x_j)) \to \frac{1}{k}$$

This result implies that adding noisy dimensions will cause us to lose discrimination between clusters. Consequently, clustering methods based on the Euclidean distance (such as k-means) will degenerate, eventually assigning all observations at random. In other words, including all $p$ dimensions spreads out the signal of the cluster assignments. Therefore, adding enough noise will eventually overpower the signal so that it is unintelligible in $p$ dimensions, even if it is clear in $r$ dimensions. Accordingly, we consider implementing feature selection in the case of high dimensional data.

### 4.2 Performing Feature Selection in K-Means Clustering

The goal of feature selection for k-means clustering is to create a reduced matrix $R_{n \times r}$, such that $R$ is generated by selecting $r$ columns of the data matrix $X$. We then use the
reduced matrix to find a resulting clustering $C^*_r$ that minimizes the reduced optimization function:

$$C^*_r = \arg \min_{C_r \in \mathbb{R}^{n \times r}} ||R - C_r C_r^T R||_2^2$$  \hspace{1cm} (4.12)$$

Here, we attempt to find a reduced feature space that recovers true underlying clusters at least as well as the full-dataset clustering. Consider the set of true (possibly hidden) classes, $\hat{C}$. In a semi-supervised context, these true classes could be known, yet hidden; however, $\hat{C}$ is completely unknown in strictly unsupervised settings. We choose a set of $r$ features such that the resulting clustering minimizes the Hamming distance ($H$) compared to the true classes, i.e.,

$$\left\{ r : H(\hat{C}, C_r) = \inf_{C_r} H(\hat{C}, C_r) \right\}$$  \hspace{1cm} (4.13)$$

where:

$$H(\hat{C}, C_r) = \sum_{i=1}^{n} 1\{\hat{C}(i) \neq C_r(i)\}$$  \hspace{1cm} (4.14)$$

The Hamming distance can be considered as a measurement of error that represents the number of misclassified samples. In scenarios where signals are strong and there are many significant features, feature selection may not lead to a better clustering. In such cases, the Hamming distance for both the full feature set and the reduced feature set are equivalent. When features are rare and their signals are weak, however, a significant number of noisy features results in an increase in the Hamming distance; it is under these precise conditions where feature selection may be valuable.

We know that in rare-weak scenarios it is possible to find a set of features $r$ such that:

$$H(\hat{C}, C_r) \leq H(\hat{C}, C_p)$$  \hspace{1cm} (4.15)$$

as long as the signal is strong enough to detect. This argument is based on the properties of what are known as phase diagrams which are discussed in the last section of this chapter.

Note that feature selection can be considered in this context by using a projection matrix $V \in \mathbb{R}^{p \times r}$ such that $XV \in \mathbb{R}^{n \times r}$ projects the dataset $X$ onto the $r$-dimensional
subset of features. In the context of our algorithm, the matrix $V$ is determined via hard thresholding. We create a matrix $V$ that selects only features considered to be the most important for clustering, i.e., features with bimodal measures above a certain threshold. The resulting matrix based on this projection is defined by $XV = R$. Formally, let $p' \in \mathbb{R}^{1 \times r}$ be such that $p'$ is a subset of the $p$ features containing only $r$ significant features selected, i.e., $p' = \{p'_1, p'_2, ..., p'_r\}$. Our projection matrix is simply defined by $V_{p \times r}$ where:

$$V[i,j] = \begin{cases} 
1 & \text{if } i = p'(j) \\
0 & \text{otherwise}
\end{cases}$$

This projection matrix simply selects the columns of $X$ that are considered useful for clustering. A scaling matrix $A_r \in \mathbb{R}^{r \times r}$ could also be defined, such that $XVA \in \mathbb{R}^{n \times r}$ would select and scale the data matrix into the $r$-dimensional feature space. Here, we assume no scaling occurs, i.e., $A_r = I_r$.

### 4.3 K-means Clustering and the Bimodal F test

Assume that an ideal clustering exists in $r$ dimensions—how do we locate the exact $r$ features to be selected in order to achieve optimal clustering? We consider the theoretical arguments surrounding the choice of the $F$ test for feature ranking, and how it is designed to locate features that contain a marginal signal, thereby improving the overall clustering. The arguments presented here are for the simplest case, and involve some standard initial conditions; nevertheless, it is clear that these results would hold in more general scenarios.

We assume the following, for simplicity:

1. Two clusters are present. ($k=2$)

2. Important features will have equal signal strength, and cluster labels of $\pm 1$; i.e., cluster means in $j$th dimension will have value $\bar{x}_k(j) = \pm \mu_k(j)$, if $j$ is an important feature.
3. Non-important features will be distributed as zero-mean Gaussian noise $N(0, \sigma^2)$; i.e., cluster means in $j$th dimension will have value $\bar{x}_k(j) = 0$, if $j$ is a noisy or unimportant feature.

We argue that the $F$ test allows us to select $r$ important features and to discard $s$ unimportant features. By using the $F$ test for FS, clustering error decreases when non-informative features are removed. Consequently, some minimum clustering error can be found by thresholding the $F$ test to include only the most important features.

Recall that the objective function is written in terms of within-cluster distance:

$$J_K = \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} (x_i(j) - \bar{x}_k(j))^2$$

For notational simplicity, let $d(C_k, C_l) = \sum_{i \in C_k} \sum_{i' \in C_l} \sum_{j=1}^{p} (x_i(j) - x_{i'}(j))^2$.

Then we can simplify the objective function:

$$J_K = \sum_{k=1}^{K} d(C_k, C_k)$$

To minimize this objective function, because all terms are additive by dimension, we can define a dimension-wise objective function for each dimension $j$:

$$J_K(j) = \sum_{k=1}^{K} d_j(C_k, C_k)$$ (4.16)

where $d_j$ is defined as the univariate distance between points in a single dimension; i.e., $d_j(C_k, C_l) = \sum_{i \in C_k} \sum_{i' \in C_l} (x_i(j) - x_{i'}(j))^2$. While minimizing $J_K$ is not technically equivalent to minimizing $\sum_{j=1}^{p} J_K(j)$, considering a univariate minimization will allow us to search for a subset of $r$ features successfully for which a global minimization (and therefore ideal clustering) is relevant under certain conditions. This will be true largely when the features are uncorrelated, or weakly correlated, as in the case of high-dimensional data where important features are both rare and weak-signaled, and when marginal distributions of
features exhibit clustering information in that dimension. This is not an extremely restrictive assumption, especially under the rare-weak signal setting. Under these conditions, we can assume that an equal or better clustering found by minimizing $J_K$ can be found by investigating $\min \sum_{j=1}^{p} J_K(j)$. In other words, we can rearrange the sums in the objective function, under the conditions listed above, without loss of generality. We present this result in the following theorem.

**Theorem 2.** Assume a setting as defined by $RW(\beta, r)$, in which features that contain significant clustering information are rare (determined by $\beta$) and weak (determined by $r$). Additionally, under this regime, assume significant features for clustering are uncorrelated, and are such that the clustering signal present in each dimension is reflected in the marginal distribution of the feature. Under these conditions:

$$\min J_K \geq \sum_{j=1}^{p} \min J_K(j)$$

and, therefore,

$$\min \frac{1}{C_{\frac{1}{2}}} \sum_{k=1}^{K} d(C_k, C_k) \geq \sum_{j=1}^{p} \min \frac{1}{C_{\frac{1}{2}}} d_j(C_k, C_k)$$

This argument is a direct result from the connection between SVD and k-means clustering, which implies that if there is a reduced feature space in which an optimal clustering exists, then trying to cluster with added noise will always result in a larger objective function. Consequently, we argue that features which contribute the most in minimizing the overall objective function will subsequently have univariate distributions that benefit from clustering.

We now turn to the $F$ test. Recall that the F test is a ratio of two values, namely $\sigma_{uni}^2(j)$ and $\sigma_{bn}^2(j)$, that are calculated in every dimension. They are defined as follows, presented here again for reference:

$$\sigma_{uni}^2(j) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i(j) - \bar{x}(j))^2$$
\[ \sigma_{bi}^2 = \frac{1}{2} \min \left[ \frac{1}{n_1 - 1} \sum_{i=1}^{n_1} (x_i(j) - \mu_{n_1})^2 + \frac{1}{n_2 - 1} \sum_{i=n_1+1}^{n} (x_i(j) - \mu_{n_2})^2 \right] \]

By definition, \( \sigma^2(j) \approx \hat{\sigma}^2(j) = \sigma_{uni}^2(j) \). This is a static value and cannot be minimized through clustering, as it represents the univariate variance of the feature. Additionally, note that \( \sigma_{bi}^2 \), which is already defined as a minimum, is approximately equivalent to finding the minimum-within cluster distance in one dimension, assuming the existence of clusters. In particular, the calculation of \( \sigma_{bi} \) is directly related to the calculation of our marginal objective function \( J_K(j) \) when \( k = 2 \), i.e., \( J_K(j) = d(C_1, C_1) + d(C_2, C_2) \). In other words:

\[ \sigma_{bi}(j) = \min \left[ \frac{1}{n_1 - 1} \sum_{i=1}^{n_1} (x_i(j) - \mu_{n_1})^2 + \frac{1}{n_2 - 1} \sum_{i=n_1+1}^{n} (x_i(j) - \mu_{n_2})^2 \right] \]

\[ \propto d_j(C_1, C_1) + d_j(C_2, C_2) = \min[J_K(j)] \quad (4.17) \]

This implies that features with significantly small values of \( \sigma_{bi} \) will correspond with features that carry pertinent clustering information in that dimension.

Recall that the \( F \) statistic is generated as:

\[ F = \frac{\sigma_{uni}^2}{\sigma_{bi}^2} \]

where, again, \( \sigma_{uni}^2 \) is considered constant with respect to dimension \( j \). By construction, the largest values of \( F \) will likely correspond to features that have relatively small values of \( \sigma_{bi}^2 \) (i.e., dimensions that will benefit the most from clustering), and consequently will correspond to dimensions with the smallest \( J_K(j) \). Thus, features with large \( F \) values are precisely the features we wish to select when clustering the data, as they are most likely to exhibit some natural separation in the \( j \)th dimension. This is exactly the process of FS by ranking the \( F \) test; if we sort the \( F \) ratio from largest to smallest, dimensions which exhibit minimal \( J_K(j) \) rise to the top, and are more likely to be selected. In other words, we want to choose features that benefit the most from considering a bimodal versus unimodal model; these features will be precisely the ones with the maximum change between \( \sigma_{uni} \)
4.3.1 Inference Concerning the Choice of F-test Threshold

In this section, we consider the two natural thresholding methods that intuitively provide the ideal cutoff for feature selection. These methods consist of a thresholding method in the case of semi-supervised data, which is based on the empirical CDF of the ranked $F$ test statistic itself. We also consider a more data-dependent threshold based on implementing higher criticism using a p-value of the $F$ statistic, which can be used in the case of completely unsupervised data. We first present arguments based on thresholding using the empirical CDF of the $F$ test statistic for $p$ features.

4.3.1.1 Thresholding the F Statistic Based on Empirical CDF

Here, we use an ad-hoc method of thresholding that is based on quantiles related to the sorted $F$ statistic. We construct an empirical distribution of the ranked $F$ statistic, and consider possible thresholds based on the values of different quantiles. A detailed explanation of this method follows.

First, denote $\tilde{F}$ as the descending sorted values of our calculated $F$ statistic, such that $\tilde{F}(i) \geq \tilde{F}(i+1)$, and note that $\tilde{F}$ is monotonically decreasing. Choosing a cutoff threshold $t$ such that $F \geq t$ is equivalent to selecting features which will by construction be the dimensions in which bimodality is strongest. We then cluster on a set of features $rt$ consisting only of features for which $\tilde{F} \geq t$, which would ideally lead to optimal cluster discrimination.

We generate a list of $m$ possible thresholds $T_m = \{t_1, t_2, ... t_m\}$ based on the quantiles of this empirical distribution function. Namely, we choose $m$ possible thresholds such that:

$$t_m = (1 - \alpha_m)\tilde{F}(1)$$

where $0 < \alpha_m < 1$ is small and represents the proportion of features kept, and the quantity $(1 - \alpha_m)$ allows us to choose a threshold equal to the $(1 - \alpha_m)$th quantile of $\tilde{F}$. In practice,
we typically choose $\alpha_m \in \{.2,.15,.1,.075,.05,.025,.01,.005\}$.

Once a list of possible thresholds has been identified, we choose an ideal threshold in the context of semi-supervised data. Our final threshold $t^*$ is the $F$ value for which the selected features have $F > t^*$ and produce minimal clustering error. Thus, for each threshold value $t_m$, we generate a set of features $S_t$ to be kept, and construct a reduced data matrix $X_t$ projected into $p'$ dimensions based on $S_t$, with $p' < p$. We then perform $k$-means clustering for each reduced dataset, and a cluster assignment $C_t$ is found.

The error is thus defined as the Hamming distance between the cluster assignments $C_t$ and the true, yet hidden classes $\tilde{C}$:

$$H(\tilde{C}, C_t) = \sum_{i=1}^{n} 1\{\tilde{C}(i) \neq C_t(i)\}$$

Let $H^*(\tilde{C}, C_t)$ represent the minimum value of this error. Our threshold is chosen as the maximum value of $t_m$ that results in $H^*(\tilde{C}, C_t)$. In other words, we choose our threshold based on the smallest reduced dataset that provides minimum error. Formally:

$$t^* = \max_{t_m} \{H(\tilde{C}, C_{t_m}) = H^*(\tilde{C}, C_{t_m})\}$$

The ideal feature set, denoted $S^*_t$, contains only features that have $F$ statistics above the given threshold $t^*$, and result in a $r$-dimensional dataset that provides minimum clustering error. Our final chosen subset of features $S^*_t$ will be:

$$S^*_t = \{1 \leq j \leq p : F(j) > t^*\}$$

Note that in practice, the size of $S^*_t$ is relatively small compared to $p$. Typically in rare-weak scenarios, less than 10% of the features are necessary and should retained in any given dataset. This is a fairly conservative estimate. In most cases, fewer than even 5% of features are retained. Accordingly, we restrict the list of potential thresholds so that the resulting reduced matrices will contain less than 20% of the data, a conservative estimate.
in practice. We argue that performing clustering on multiple feature sets before finding the optimal set is appropriate, because the final feature set only includes an extremely small proportion of features (especially in the rare-weak model scenario), and because clustering on these reduced datasets require very little run time and small computational cost.

In the context of rare-weak conditions, clearly the rarity and weakness of the signal are directly related to the number of features retained. Thus, increasing the rarity and weakness of significant features will lead to choosing a lower threshold, thereby instituting a necessary increase in the number of features selected. This makes sense intuitively, because weak signals will be harder to discriminate compared to noisy features. This method works in what we have termed a “semi-supervised” context, in which the actual cluster assignments are known but have been stripped from the data. In particular, we use supervised datasets that have been acquired from retrospective studies, where class assignment information is known. We proceed by stripping the class assignments $\tilde{C}$ from the dataset, thus creating a matrix $X$ of size $n \times p$ representing “unsupervised” data.

This semi-supervised approach is restrictive in its use, because it requires prior knowledge of classes. Nevertheless, it proves to be a particularly advantageous approach for a number of reasons. The most useful of these reasons is that it provides a “gold-standard” clustering assignment with which we can compare our clustering results. Using semi-supervised data allows us to find the exact clustering error for each method, and, thus, to determine areas of the data-verse for which this method thrives. Using a semi-supervised context allows us a direct comparison between the accuracy and predictive quality of our models, showing that they provide nearly-ideal clusters in practice. Consequently, this semi-supervised context allows us to consider the application of this method in a supervised context, even though it is particularly generated to be useful in unsupervised settings.

4.3.1.2 Thresholding Using Higher Criticism

While the method of threshold selection described above can be particularly advantageous in a semi-supervised context, there can be significant drawbacks. The most obvious
drawback is that choosing an ideal threshold using a minimum hamming distance is not possible when the true classes are completely unknown as opposed to hidden. For example, in prospective studies where the data is entirely unsupervised, thresholding using this method would provide a list of possible thresholds with no accurate way to determine which threshold is in fact the ideal. Therefore, we explore other thresholding options to optimize data clustering algorithms.

In particular, we consider using the notion of unsupervised higher criticism thresholding (UHC) based on a p-value associated with both the parametric and nonparametric feature rankings, and show that this thresholding measure leads to optimal clustering in practice. The advantage of this measure is it has been theoretically proven to mimic the ideal classification boundaries in an asymptotic context - even in the case where the underlying classes are completely unknown. For now, we assume that UHC is an appropriate threshold, and discuss finding a p-value from the $F$ test to use in UHC. Later on in this chapter, we specifically discuss the theoretical advantages to UHC, namely how it maximizes the signal-to-noise-ratio and, ultimately, provides the ideal clustering. We also establish the connection between UHC and how it subsequently minimizes the Hamming distance and increases cluster quality in rare-weak settings.

4.3.1.3 Tail Probability for the $F$ Distribution

What remains to be discussed is the calculation of the survival function of the $F$ distribution, i.e., the right tail probability (p-value) of the $F$ statistic. By construction of the $F$ test, we expect the tail probability of interest (i.e., $P(F > F_n)$) to follow a (non-central) $F$ distribution. Let $F_{1/\lambda}$ represent the a $F$ statistic adjusted by the non-centrality parameter $1/\lambda$. Then, we argue that:

$$F(j) \sim F_{n-1,n-2v-2}(1/\lambda) \quad (4.20)$$
Where \( n \) represents the number of samples, and \( v \) represents the number of splits that are generated in the construction of \( \sigma_{bi} \). In other words, we argue that the \( F \) statistic will follow a non-central \( F \) distribution. This claim is supported in the following argument.

Again, consider the \( F \) statistic:

\[
F = \frac{\sigma_{uni}^2}{\sigma_{bi}^2}
\]

Note that under the assumption of Gaussian noise, we expect \( \sigma_{uni}^2 \) to represent an estimate for the variance in dimension \( j \) and will, therefore, follow a chi-squared distribution, i.e.:

\[
\sigma_{uni}^2 \sim \chi^2_{n-1}
\] (4.21)

Consider the construction of \( \sigma_{bi}^2 \), and recall that it is defined as a minimum average over the left and right sides of the distribution, given the ideal split. Let \( L(j) \) represent the set of points to the left of the split, and \( R(j) \) represent the set of points to the right of the split. Under the Gaussian mixture distribution assumption, we expect the variance of \( L(j) \) (which we will denote \( \hat{\sigma}_{L}^2(j) \)) and the variance of \( R(j) \) (\( \hat{\sigma}_{R}^2(j) \)) to be distributed as follows:

\[
\hat{\sigma}_{L}^2(j) \sim \chi^2_{n_1-1} (\lambda_1) \quad \text{and} \quad \hat{\sigma}_{R}^2(j) \sim \chi^2_{n_2-1} (\lambda_2)
\] (4.22)

Where \( \lambda_1 \) and \( \lambda_2 \) represent non-centrality parameters. In other words, we expect the left and right pieces of the overall distribution to follow non-central chi-squared distributions. Consequently, we argue:

\[
\sigma_{bi}(j) = \min \frac{1}{2} \left( \hat{\sigma}_{L}^2(j) + \hat{\sigma}_{R}^2(j) \right) \sim \chi^2_{n-2v-2} (\lambda)
\] (4.23)

In other words, we expect \( \sigma_{bi}^2 \) to also follow a non-central chi-squared distribution (as it is a minimum of the sum of two non-central chi-squared distributions), with centrality parameter \( \lambda \) that is dependent on \( \lambda_1 \) and \( \lambda_2 \). Under normality conditions we assume the non-centrality parameters will be simply the means of the left and right partial distribu-
tions; i.e., \( \lambda_1 = \mu_L^2 \) and \( \lambda_2 = \mu_R^2 \), and consequently \( \lambda = \frac{1}{2}(\mu_L^2 + \mu_R^2) \).

Thus, the ratio of \( \sigma_{uni}^2 \) and \( \sigma_{bi}^2 \) in the \( F \) test is a ratio of a central and non-central \( \chi^2 \) distribution, and thus will follow a non-central \( F \) distribution as defined in the beginning of this section. This result is implied by the work in [53], where this statistic is known to have a non-central \( F \) distribution in the case of testing whether a distribution has one vs. two modes. Once this distribution is clearly defined, finding p-values associated with the test statistic becomes fairly straightforward. We construct a p-value based on the CDF of the non-central \( F \) distribution:

\[
p(j) = P(F > F_{n-1,n-2v-2}(1/\lambda))
= 1 - \sum_{l=0}^{\infty} \left( \frac{1}{l!} \exp^{-\lambda/2} \right) \mathcal{I}_{(n-1)x+n-2v-2} \left( \frac{n-1}{2} + l, \frac{n-2v-2}{2} \right)
\]

(4.24)

Where \( \mathcal{I} \) represents the regularized incomplete beta function, as defined for the CDF of a non-central \( F \) distribution with \( n \) representing the number of samples, and \( v \) representing the number of splits used in the calculation of \( \sigma_{bi} \). In particular, \( \mathcal{I} \) is defined in this case as:

\[
\mathcal{I} = \frac{B\left(\frac{(n-1)x}{(n-1)x+n-2v-2}, \frac{n-1}{2} + l, \frac{n-2v-2}{2}\right)}{B\left(\frac{n-1}{2}, \frac{n-2v-2}{2}\right)}
\]

(4.25)

where:

\[
B\left(\frac{(n-1)x}{(n-1)x+n-2v-2}, \frac{n-1}{2} + l, \frac{n-2v-2}{2}\right) = 
\int_{0}^{\left(\frac{n-1}{2}+l\right)x+n-2v-2} t^{\frac{n-1}{2}+l-1}(1-t)^{\frac{n-2v-2}{2}-1} dt
\]

and \( B(\frac{n-1}{2}, \frac{n-2v-2}{2}) \) is a constant, dependent on \( n \) and \( v \). There are many numeric procedures that estimate this tail probability well based on the \( F \) distribution that are inherent to statistical packages. Therefore, we simply refer to this as the typical p-value based on the non-central \( F \) distribution with non-centrality parameter \( 1/\lambda \) and with \( n-1 \) and
Based on this general construction, with reasonable sample size, this test will have an error on the order of $o(1/\sqrt{n})$, because the distance between the mean of the distribution and the ideal cluster split will be $o(1/\sqrt{n})$. Consequently, the error in the test will approach 0 with increasing $n$ under given normality conditions.

Note that this $F$ distribution holds particularly in the case with $k = 2$, i.e., when we are using the $F$ test to specifically determine whether a distribution is unimodal vs. bimodal. However, in the case of testing multimodality, the distribution of this statistic is unclear - it is not certain that the resulting statistic will always follow an $F$ distribution. In the case of the multimodal $F$ test, therefore, we argue that the best p-value to use in the construction of the higher criticism threshold is one based on the empirical distribution of the test statistic, $F$.

4.4 K-means Clustering and the Bimodal KDE Model

We make a similar argument for the usefulness of feature ranking and selection in a non-parametric context. This argument takes place in two different scenarios - one which assumes semi-supervised data, and another which assumes completely unsupervised data. Both arguments follow a heuristic approach to informally explain how the construction of a KDE provides sufficient information for feature ranking and thresholding.

Recall that a kernel density estimate is generated for each univariate feature distribution as:

$$
\hat{f}_n(x(j)) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x(j) - x_i(j)}{h}\right)
$$

Here, the kernel of interest is the biweight kernel:

$$
K(u) = \frac{15}{16}(1 - u^2)^2\mathbb{I}_{\{|u|<1\}}
$$

Due to kernel density estimation theory, the choice of the bandwidth $h$ is crucial in this
estimate. Ideally, one would consider the asymptotic mean-integrated squared error:

$$AMISE(\hat{f}_n(x(j))) = \frac{R(K)}{nh} + \frac{1}{4} m_2^2(K) h^4 \|f''\|_2^2$$ (4.28)

where $R(K) = \int K^2(u)du$ and $m_2^2(K) = \int u^2 K(u)du$. We choose the bandwidth $h$ such that this criterion is minimized, namely:

$$h_{AMISE}(j) = \left[ \frac{R(K)}{nm_2^2(K)\|f''\|_2^2} \right]^{1/5}$$ (4.29)

For the biweight kernel, it is known that $R(K) = 5/7$ and $m_2^2(K) = 1/7$. Thus these equations for AMISE and subsequent optimal $h$ are simplified for the biweight kernel as:

$$AMISE(\hat{f}_n(x(j))) = \frac{5}{7nh} + \frac{1}{28} h^4 \|f''\|_2^2$$ (4.30)

and:

$$h_{AMISE}(j) = \left[ \frac{5}{n\|f''\|_2^2} \right]^{1/5}$$ (4.31)

In this context, the optimal bandwidth estimate is dependent on the second derivative of the unknown true density function $f$. Accordingly, we consider the rule-of-thumb choice of bandwidth in each feature dimension $j$, as developed in [76]:

$$h(j) = \hat{\sigma} C_v(K) n^{-1/5}$$

Where $C_v(K)$ is a known constant based on the choice of kernel, and $\hat{\sigma}$ represents an estimate for the standard deviation of the distribution of feature $j$. Based on our kernel selection, $C_v(k) = 2.78$. This bandwidth selection is a common choice in practice because it is determined in a data-dependent manner, and is known to be a good estimate for the optimal bandwidth when the true underlying density (and its subsequent derivatives) is unknown [76].

We now establish the connection between our use of KDE for feature ranking and k-
means clustering. Recall that in the context of k-means clustering, our objective is to minimize the marginal optimization function in each feature dimension:

\[ J_K(j) = \sum_{k=1}^{K} d_j(C_k, C_k) \]

In other words, we aim to minimize the within-cluster distance or, equivalently, the difference between total variation and between-cluster variation. Accordingly, we can rewrite our objective function as:

\[ J_K(j) = \sum_{k=1}^{K} d(C_k, C_k) = n\sigma^2(j) - \sum_{k=1}^{K} \sum_{l \neq k} d_j(C_k, C_l) \]

Intuitively, we want to investigate features that benefit the most from clustering; these features have larger values of \( \sigma(j) \), but also have the property that \( J_k(j) \) is significantly reduced when clustering is implemented (i.e. they will also have large values of \( d_j(C_k, C_l) \) with \( k \neq 1 \) and small values of \( d_j(C_k, C_k) \)).

In our semi-supervised nonparametric bimodal test, our ranking system is intuitive; we rank features first by the number of identified modes, and secondly by the distance between modes. Recall that this is done by first normalizing the density estimate to have a maximum of 1, identifying local modes, and then calculating modal separation based on Euclidean distance. Higher-ranked features should be those dimensions for which more than one mode is identified and distance between modes is large, as these will be features which exhibit bimodal distributions and are therefore more likely to contain pertinent cluster information. It is clear that features or dimensions that are ranked high on this list will be exactly those features with large total and large between-cluster variance; i.e., those features with the smallest \( J_k(j) \).

In a completely unsupervised context, this argument requires further explanation, however. We argue that features with the largest values of \( \hat{h}(j) \) will benefit the most from clustering and, consequently, contain pertinent clustering information. This argument is
based computationally on the bandwidth construction. It is known that larger values of \( \hat{h} \) are associated with “smoother” density estimates. In particular, due to the direct relationship between the bandwidth value \( \hat{h}(j) \) and the variance estimate \( \hat{\sigma}^2(j) \), we know that features containing a larger total variance will also be features that have a larger associated bandwidth estimate. Therefore, features exhibiting larger bandwidth estimations \( \hat{h}(j) \) contain data that are more spread apart, and therefore have the potential to benefit the most from clustering. Accordingly, we argue that ranking the features based on the critical bandwidth will be analogous to finding features that minimize \( J_k(j) \). The proper thresholding choices for each ranking method are considered in the following section.

4.4.1 Thresholding the KDE By Modal Identification and Separation

In this section, we discuss the thresholding method of the nonparametric algorithm in a semi-supervised context, i.e. when the true clusters are known but hidden, and our goal is to recover them with minimal error.

As explained in Chapter 3, each feature density was normalized to have a maximum of 1. Potential modes were identified as peaks which were greater than 0.15. If more than one potential mode is identified, the Euclidean distance between modes is measured. The features are then ranked based first by the number of modes, secondly on modal intensity, and thirdly on the distance between modes.

In considering thresholding options for this type of ranking, we establish a list of potential thresholds, and generate reduced datasets based on features whose ranking lies above these thresholds. Samples are then clustered based on each reduced dataset, and the threshold that corresponds to the reduced dataset that provides minimum clustering error when compared to the true classes is considered ideal.

The list of possible thresholds, \( t \), is identified based modal intensity and location in the density estimate. Therefore, we choose different possible thresholds based on changing these parameters. Namely, let \( M(j) \) represent the number of potential modes in the density estimate of feature \( j \), and \( S_m(j) \) represent the distance between modes in the density
estimate of feature $j$. Note that if $M(j) = 1$, $S_M(j) = 0$. If $M(j) = 2$, $S_M(j) > 0$, and if $M(j) > 2$, we let $S_M(j)$ represent the distance between the furthest two modes in the density estimate. The most intuitive threshold here is to choose features that exhibit more than one mode, i.e.:

$$S_t = \{1 \leq j \leq p : M(j) > 1\}$$ (4.32)

In practice, however, we find that this thresholding method is too conservative; this method will keep too many features, some of which may ultimately be uninformative. This happens based on fluctuations in the density estimate near the true mode itself; potential modes that are identified close together could represent fluctuations in measurement noise rather than true separate modes in the distribution. Thus, we add an additional constraint to the threshold, one that requires $M > 1$ and a large separation between modes. Possible threshold choices are based on the amount of separation allowed between modes. In particular, if we let $s_m = \{0, 0.5, 1, 1.5, 2, 2.5, 3\}$, we find our list of possible thresholds as

$$t(i) = \{1 < j < \alpha_0 p : M(j) > 1 \text{ and } s_M(j) > s_m(i)\}$$ (4.33)

This limits the number of possible thresholds to the length of $s_m$. For each possible threshold choice $t(i)$, a reduced matrix $X_{t(i)}$ is generated, such that all features that rank above the rank of $t(i)$ are included. For each reduced matrix, clustering is performed, and a cluster assignment $C_{t(i)}$ is found. The error is again defined as the Hamming distance between the cluster assignments and the true, yet hidden classes:

$$H(\tilde{C}, C_t) = \sum_{i=1}^{n} 1\{\tilde{C}(i) \neq C_t(i)\}$$ (4.34)

and our threshold is chosen as:

$$t^* = \min_t H(\tilde{C}, C_t) = H^*(\tilde{C}, C_t)$$ (4.35)
The ideal feature set, denoted $S^*_t$, contains only features that rank above the given threshold $t^*$, which is similar to the $F$ test ranking and will produce minimum clustering error by construction.

This feature selection process contains the same limitations as the parametric setting - namely that the ideal threshold is chosen based on comparing the clustering assignments to the known yet hidden true class values. Therefore, while this method works well in the case of semi-supervised data, the problem of choosing a data-dependent threshold in a completely unsupervised context still remains.

### 4.4.2 Thresholding Using Higher Criticism

In a completely unsupervised setting, we assume that UHC is appropriate, and discuss finding a p-value from the KDE to use in UHC. We discuss the reasoning behind our choice of UHC for thresholding in the last sections of this chapter.

While finding a p-value for the $F$ statistic is straightforward, finding a p-value in a nonparametric setting is challenging. However, if a p-value can be generated based on the KDE of each univariate distribution, then UHC can be implemented easily. We argue how to generate a p-value for the nonparametric setting, and how to use in UHC to find the appropriate threshold in a completely unsupervised context.

The two main factors are necessary to determine a density estimate; the choice of kernel used, and the bandwidth. As explained previously, the choice of kernel is the biweight kernel, which is useful because it offers compact support of the density estimate resulting in short computational run time. The use of local information only is also a desired quality for a kernel choice in “bump-hunting,” or investigating the modality of a density estimate. Accordingly, the biweight kernel is a rather standard kernel choice when investigating modality.

The other critical parameter in KDE is the bandwidth, as it determines the number of modes present in a density estimate. In the case of the Gaussian kernel, Silverman presents a particularly useful test for unimodality, based on the monotonicity property of
the bandwidth [76]. The intuition behind using the bandwidth in a discussion of modality is that “if the data are strongly bimodal, a large value of $h$ will be needed to obtain a unimodal estimate.” Therefore, to test the hypothesis:

$$H_{0,j} : \text{the density } \hat{f}(j) \text{ has exactly } k \text{ modes}$$

$$H_{A,j} : \text{the density } \hat{f}(j) \text{ has more than } k \text{ modes}$$

(with typically $k = 1$), the critical window width can be constructed such that:

$$h_{\text{crit}} = \inf \{ h : \hat{f}(., h) \text{ has at most } k \text{ modes} \}$$

This criteria can be used to ultimately determine whether $\hat{f}(., h)$ has more than $k$ modes. More specifically, $h_{\text{crit}}$ represents the smallest value of $h$ for which the estimated density has at most $k$ modes. Therefore, the null hypothesis that $\hat{f}$ has exactly $k$ modes can be rejected if and only if $h < h_{\text{crit}}$.

While this argument is strong, this test is specifically designed for the Gaussian kernel, and is only applicable when modality is known to be a monotone function of bandwidth. In the case of non-Gaussian kernels (including the biweight kernel), however, there is a strong non-monotonicity property that holds, which would imply that Silverman’s modality test based on bandwidth would not be appropriate here. Hall et. al. argue the following theorem:

**Theorem 3** ([40]). Suppose $K$ is a symmetric and strictly unimodal probability density with support equal to $I = [-1,1]$, continuous on the real line, four times continuously differentiable on $I$, and with the property that $K''_{\psi}(0) > 0$, $K'_{\psi}(\eta) > 0$, and $K''_{\psi}(\eta) > 0$ for some $\psi, \eta \in (0, 1)$.

Then if $n \geq 3$, and if $X = \{X_1, ..., X_n\}$ denotes a random sample drawn from a continuous distribution, then with strictly positive probability the number of modes of $\hat{f}_h$ is
not monotone in \( h \).

This theorem holds for the biweight kernel, because any kernel of the form \( K(x) = C_\theta(1 - x^2)^\theta \) on \( I \) where \( 5/2 \leq \theta \leq 11/2 \) satisfies the conditions in the above theorem. [40]

The result is that \( h_{\text{crit}} \) can be determined explicitly in a Gaussian kernel, but it is ambiguous in non-Gaussian kernels. In a Gaussian density, the number of modes is known to be a monotone non-increasing function of the bandwidth. Therefore, for any given dataset, there is a bandwidth below which all density estimators will have at least two modes, and above which all estimators will only have one mode [73]. However, non-Gaussian kernels (including the biweight kernel) result in a strictly non-monotonic relationship between the number of modes in the density and the bandwidth.

Although this appears to be an immediate problem, this non-monotonicity is not debilitating. In fact, if we ignore the tail effects of the non-Gaussian KDE, we can perform a similar test for unimodality in the case of non-Gaussian kernels [40]. In particular, it is possible to use a similar test for modality “by decreasing through bandwidths for which \( f \) is unimodal, although it is generally not possible to define a critical bandwidth by increasing through bandwidths for which \( f \) is multimodal.” This claim is supported by bootstrapping, as illustrated in the following theorem, first presented in [40].

**Theorem 4** ([40]). Let \( X^*_1, ..., X^*_n \) be a resampling with replacement conditioned on \( X \) from the distribution with density \( f \), and define

\[
\hat{f}^*(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X^*_i}{h}\right)
\]

and let \( h^*_{\text{crit}} \) be the critical bandwidth analogous to \( h_{\text{crit}} \) in this context. Then given a nominal level \( \alpha \) for the test of unimodality, when the null hypothesis is false and as \( n \to \infty \),

\[
P(h^*_{\text{crit}}/h_{\text{crit}} \leq 1|X) \to 1
\]

and therefore the probability that the test leads to an appropriate rejection converges to 1.
The result of this theorem implies that we can use re-sampling methods such as those described in the Gaussian kernel case in [77] in order to estimate a p-value for use in higher criticism.

The most rate-limiting step to this approach, however, is the determination of \( h_{\text{crit}} \) in practice. In most cases, the value of \( h_{\text{crit}} \) must be uncovered through re-sampling methods such as bootstrapping or other computationally intense manners. We have found that in the case where we can safely assume that the noise in each dimension acts in a common Gaussian or near-Gaussian manner, it is possible to forgo this additional computation, and instead base our test on the relative comparison of bandwidths among all \( p \) features. This process involves finding an “ideal” smoothing bandwidth parameter based on the distribution of all bandwidth values in a large-testing problem, and then uses these values to perform an analogous test to the one presented above, which requires significantly reduced computation time.

In order to do so, we extend the ideas of [76, 40] and others by creating an alternate version of a p-value based on the biweight kernel using the idea of large-test scaling presented by Efron [26]. We capitalize on the principle that larger ideal bandwidth values are strongly associated with bimodally or multimodally distributed features, because these distributions will be naturally “bumpier” and will require a larger value of the smoothing parameter. These are exactly the features which will contain pertinent information for clustering.

We consider the problem of constructing p-values in the case of large scale simultaneous hypothesis testing. Given a collection of hypothesis tests \( H_1, \ldots, H_p \) with associated test statistics \( q_1, \ldots, q_p \), which may or may not be independent, associated p-values for each test statistic \( P(q_1 > Q_1), \ldots, P(q_p > Q_p) \), and \( p \) relatively large (i.e. \( p > 100 \)), authors in [26] conclude that “large scale testing situations permit empirical estimation of the null distribution.” Capitalizing on the large number of features, we claim that the asymptotic distribution of the standardized p bandwidths as \( n \to \infty \) should be normally distributed. We employ the standardization approach proposed in [26] to normalize the bandwidth
distribution:

\[ h_s(j) = \frac{h(j) - \bar{h}}{s(h)} \]  \hspace{1cm} (4.36)

where \( h(j) \) represents the bandwidth of the \( j \)th density estimate, \( \bar{h} \) represents the average of all \( p \) bandwidths, and \( s(h) \) represents the standard deviation of all bandwidths. In other words, by translating the test statistic as above so that the mean and variance of the empirical null fit with the data, then the asymptotic distribution of our test statistics \( q_1, \ldots, q_p \) (or in our case, the bandwidth estimates \( \hat{h}(j) \)) as \( p \to \infty \) will be iid zero-mean normals. Consequently, p-values associated with these test statistics can be found based on the standard normal distribution.

Authors in [26] suggest that this simple standardization method works exceptionally well in practice; in fact, the empirical distribution of the standardized \( h_s(j) \) can be shown to follow the theoretical standard normal distribution, i.e. \( h_s(j) \sim N(0,1) \).

This claim is supported in the following figure, which provides a histogram of all normalized bandwidths \( h_s(j) \) for \( j = 1, \ldots, p \) genes based on simulated data, with \( p = 1e4 \) and \( n \approx 100 \). The red line represents the standard normal distribution, and supports the claim that this distribution is appropriate.

\[ \text{Figure 4.1: Distribution of Normalized Bandwidth} \]

Based on this result, we conclude that the best way to assign a p-value to the bandwidth is to treat the value \( h_s(j) \) as a “z-score for the bandwidth of feature \( j \)”, and use this
information to determine a p-value for the density estimate of each feature. Accordingly, we re-define the test statistic for bimodality (i.e. testing whether a feature contains \(k\) modes versus the alternative that it contains more than \(k\) modes) based on the normalized bandwidth distribution, by determining a p-value:

\[
p(j) = P(\text{density estimate has more than } k \text{ modes}) \\
= P(h < h_{crit}) \\
= P(h_s(j) > Z_\alpha(h_{crit}))
\]  

(4.37)

This is the p-value calculated treating \(h_s(j)\) as a z-score and computing the area to the right of this value, based on the standard normal distribution. The p-value is, by design, meant to test specifically whether the density has exactly \(k\) modes, versus the alternative that the density has more than \(k\) modes, or equivalently \(h > h_{crit}\) vs \(h < h_{crit}\).

This method of constructing a p-value works extremely well in practice. When the p-value defined above is used in conjunction with UHC thresholding, our results demonstrate that our method offers optimal clustering performance with minimal error based on significantly reduced feature spaces. Note that the construction of the p-value is resistant to the method by which the optimal bandwidth is acquired. In our simulations, we found that the choice of the definition for \(\hat{\sigma}\) (which ultimately affects the choice of \(h\)) does not overwhelmingly affect the features selected. Accordingly, the “rule of thumb” bandwidth is sufficient in all cases.

### 4.5 Asymptotic Properties of HCT and UHC

We now consider the theoretical advantages to thresholding our new bimodal FS methods using higher criticism. Authors in [22] explain that threshold classifiers are advantageous feature selection models. The literature suggests that there are theoretical advantages to using the higher criticism method for supervised feature selection. Authors in [51] further
explain theoretically how the “decision threshold provided by HC may also be viewed as an approximation to a natural class boundary (CB) in two-class discriminant analysis.”

Extensive work has been done regarding the use of higher criticism as a thresholding method; in this section we discuss the asymptotic relevance of using higher criticism for feature selection, which is used as a thresholding method in both the parametric and non-parametric modal feature selection models presented in Chapters 2 and 3. We argue that similar asymptotic properties hold for both supervised (HCT) and unsupervised (UHC) thresholding using higher criticism. We also argue that UHC thresholding provides a natural and data-dependent threshold FS, and mimics the ideal threshold for significant features under mild conditions.

For simplicity, we again consider the two-class situation as outlined in [22, 90] and others:

\[ X = y\mu' + z \]

, where \( y \) represents the hidden or unknown class assignment (i.e. \( y = \pm 1 \)), \( \mu \) represents the signal of the cluster, and \( z \) represents a matrix containing iid entries from a zero-mean normal distribution, \( N(0, 1) \), which indicate measurement noise. Additionally, we assume the clustering information in each dimension is contained in the marginal distribution of that feature, and that features are uncorrelated. Note that the properties shown here also exist when the assumptions regarding the noise distribution and correlation between features are relaxed, as shown in [21, 23, 47, 48]. We additionally assume both \( n \to \infty \) and \( p \to \infty \), under the relation \( n = p^\theta \) for some \( 0 < \theta < 1 \), which reflects a \( p \gg n \) setting. Our goal is to generate \( p \) hypothesis tests of the form:

\[
H_0(j) : x(j) \sim N(0, 1) \quad \text{vs.} \quad H_1(j) : x(j) \sim N(\mu(j), 1)
\]  

(4.38)

where \( \mu(j) > 0 \).

Due to the particular setting where signals are rare and the number of features are weak, we expect the null hypothesis to be true in almost all cases, with the exception of a
small fraction of tests. Using the parametrization of [21, 46], where $\epsilon_p$ represents the rarity of a signal (dependent on $p$), and $\mu(j) = \mu_{\tau_p}$ for all $j$, i.e., each important feature carries an equal signal (dependent on $p$ and strength $\tau$), we have the following global hypothesis tests based on all $p$ tests:

$$H_0 : x(j) \sim i.i.d. N(0, 1) \quad 1 \leq j \leq p$$

$$H_1 : x(j) \sim (1 - \epsilon)N(0, 1) + \epsilon N(\mu_{\tau_p}, 1) \quad 1 \leq j \leq p$$

Note that $H_0$ represents the global null, but $H_1$ represents a specifically-valued alternative of interest. When the values of $\tau_p$ and $\epsilon_p$ are precisely known and fixed, the optimal test for this setup is the likelihood ratio test (LRT). [?] However, the performance of the LRT is directly connected to the rarity and weakness of the signal. This problem becomes very hard (but not quite impossible), even in the case where these parameters are fully known, when the values $\tau$ and $\epsilon$ [23] are such that:

$$\tau > \sqrt{2 \log p} \quad \text{and} \quad \frac{1}{p} < \epsilon < \frac{1}{\sqrt{p}}$$

If we re-parametrize $\tau_p = \sqrt{2r \log p}$ and $\epsilon_p = p^{-\beta}$, the rare-weak scenario is exactly when $0 < r < 1$ and $1/2 < \beta < 1$. [Donoho/Jin] Thus, we refer to the model as $ARW(r, \beta, \theta)$ which fully defines the signal strength and sparsity of the model, where $\theta$ represents the relationship between the size of $n$ and $p$; for a more full definition, we consider $ARW(r, \beta, \theta) = ARW(r, \beta, c, \gamma, \delta)$ which considers the class size $\delta$, and the relationship and speed at which $n$ and $p$ tend to infinity by the relationship $n \sim c(\log p)^\gamma$ for some constants $(c, \gamma)$. However, $r$ and $\beta$ are the primary parameters of interest.

Authors in [23, 51, 48] and others argue that boundaries exist based on values of the parameters $r$ and $\beta$, as defined above, for which signals are undetectable, detectable, and recoverable, even in the case where $r$ and $\beta$ are fully known. This theoretical boundary is known as the phase space boundary, and exists for any thresholding method applied to the
ARW regime. The details of the boundary are explored in the following section.

4.5.1 The Phase Space Boundary

We begin with a discussion of the optimal phase space boundaries in the region $r, \beta \in (0, 1)^2$, based on FS given an ideal threshold in an supervised context, i.e. when $r$ and the true class for each sample is fully known. The following theorem is proved in [23], as well as subsequent papers:

**Theorem 5 ([23]).** Consider the function:

$$
\rho^*(\beta) = \begin{cases} 
0 & 0 < \beta \leq 1/2 \\
\beta - 1/2 & 1/2 < \beta \leq 3/4 \\
(1 - \sqrt{1 - \beta})^2 & 3/4 < \beta < 1 
\end{cases}
$$

The success region is then given by the region $r > \rho^*(\beta)$, with $0 < \beta < 1$. Consequently, in the interior of the complementary region $r < \rho^*(\beta)$, $1/2 < \beta < 1$, even an ideal threshold cannot successfully identify separation of classes as $n, p \to \infty$.

This theorem identifies an exact boundary for signal recovery, and consequently an area of the phase space region for which feature selection proves useful. Therefore, for any values of $r, \beta$ such that $r < \rho^*(\beta)$, any feature selection tests (i.e., tests of the form $H_0(j)$: feature $j$ is useless vs. $H_1(j)$: feature $j$ is useful), including the idea likelihood ratio test LRT (which would be used when $r$ and $\beta$ are fully known) will have the sum of the Type I and Type II errors tending to 1 as $p \to \infty$. Conversely, when $r > \rho^*(\beta)$, then there exists a sequence of tests for which the sum of the Type I and Type II errors tends to 0 as $p \to \infty$, and the alternative can be detected reliably by the LRT. The proof is laid out for the simplist case in which features are uncorrelated ($\Sigma = I_p$) in [23], while it also holds in the general case and is proved in [48]. Note that these arguments relate to the identification of some *ideal* threshold $t^*$, which relates to the ideal cutoff for selection...
features when the parameters are fixed and fully known. However, the unfortunate nature
of the LRT requires the values of $r$ and $\beta$ to be fully known, which is unusual in practice.

A similar result holds true for the threshold associated with the original HCT functional.
We digress here to re-consider the construction of the original HCT in a supervised context
(which we will refer to as OHC), and argue that although we implement a variant of this
definition, our variant will theoretically contain the same properties as the original.

The original HC statistic, as defined in Chapter 1, is given by:

$$HC(j; \pi(j)) = \sqrt{p} \frac{j/p - \pi(j)}{\sqrt{j/p(1 - j/p)}}$$

(4.40)

The thresholding method here capitalizes on the principle that the p-values should
follow a uniform distribution under the null hypothesis of all features being equally im-
portant. Consequently, the ordered p-values are asymptotically normally distributed, i.e.
$\pi(j) \sim AN(i/p, \sqrt{i/p(1-i/p)})$. The HCT functional above is thus constructed directly
based on this distribution assumption. The threshold using this functional is defined as
the point $j^*$ such that:

$$HC^* = HC(j^*; \pi(j^*)) = \max_{1 \leq i \leq p/2} (HC(j; \pi(j)))$$

(4.41)

In relation to the hypothesis test for the $j$th feature, to conduct an $\alpha$-level test for the
global null, we must find a critical value $h(p, \alpha)$ as defined in [22, 47, 49] such that:

$$P_{H_0}\{HC^* > h(p, \alpha)\} = \alpha$$

(4.42)

Then, we reject $H_0$ if and only if:

$$HC^* \geq h(p, \alpha_p)$$

(4.43)

Note that in [75] it is shown that $h(p, \alpha) \sim \sqrt{2\log\log p(1 + o(1))}$. Authors in [23] prove
the following result:

**Theorem 6 ([23]).** Define \( \theta \in (0, 1) \) such that \( 0 < \beta < (1 - \theta) \) and \( n = p^\theta \). Given \( \rho^* (\beta) \) above, define

\[
\rho_0^* (\beta) = (1 - \theta) \rho^* \left( \frac{\beta}{1 - \theta} \right) \quad 0 < \beta < (1 - \theta)
\]

Then fixing \( (\beta, r, \theta) \in (0, 1)^3 \) such that \( 0 < \beta < (1 - \theta) \), if \( r > \rho_0^* (\beta) \) and if \( \alpha_p \) tends to 0 slowly enough (on the order of \( \sqrt{2 \log \log p} \)), the classification error of the trained HCT rule tends to 0 (and the power of the HCT rule tends to 1) as \( p \to \infty \). Similarly, if \( r < \rho_0^* (\beta) \), the classification error of any trained classification rule is no less that \( 1/2 + o(1) \) where \( o(1) \to 0 \) as \( p \to \infty \).

Authors in [27] argue that the consequence of the above theorem leads to subsequent “region of classifiable” and “region of unclassifiable” phase diagram in the context of OHC. Furthermore, it has been argued [27, 23] that the behavior of the OHC is similar to the ideal threshold in these regions, and that the phase boundaries for the success of OHC mimic the phase boundaries of the ideal threshold:

1. In the Classifiable Region \( (\beta, r) : 0 < \beta < 1 - \theta, r > \rho_0^* (\beta) \), the HC threshold \( t_p^{OHC} \) is such that \( t_p^{OHC} / t_p^* \to 1 \) in probability, with \( t_p^* \) representing the ideal threshold based on the scenario where parameter values of \( r, \beta, \) and \( \theta \) are known. Additionally, the classification error based on OHC thresholding tends to 0 as \( p \to \infty \).

2. In the Unclassifiable Region \( (\beta, r) : 0 < \beta < 1 - \theta, r < \rho_0^* (\beta) \), signals are too rare and too weak to be valuable for classification. The classification error based on any thresholding method, including the OHC method, is not substantially smaller than \( 1/2 \), and tends to \( 1/2 \) as \( p \to \infty \).

This theorem indicates that OHC is known to approximate the ideal class boundary threshold, and does so with low variance and fast computations. Authors in [23] explain in their 2009 follow up paper:
“The central surprise of our story is that ideal HCT behaves surprisingly well: the partition of phase space describing too regions where ideal thresholding fails and/or succeeds also describes the two regions where HCT fails and/or succeeds...here ‘succeeds’ means asymptotically zero misclassification rate, and ‘fails’ means asymptotically 50 per cent misclassification rate. In this sense of size of regions of success, HCT is just as good as the ideal threshold. Such statements cannot be made for some other popular thresholding schemes, such as false discovery threshold selection.” [23]

In other words, within the phase space, there is an optimal partition of performance, and the feature selection method based on higher criticism will give the same partition as the ideal. Consequently, in the idealized case where $r$ and $\beta$ (and consequently $\epsilon_p$ and $\tau_p$) are fixed known, the optimal test is the Neyman-Pearson LRT; when these parameter values are unknown, the HCT will mimic the ideal, and requires only the knowledge of p-values to do so. [47]

These results above were discovered in the case of supervised HC, where the p-value used in the HCT functional was one defined based on a z-score measuring the relationship between features and the known classes. However, authors Donoho and Jin argue in [21] that while their p-value works well in supervised cases, their construction of the p-value is not required for these thresholding properties to hold. In particular, they note the following:

“As we have defined it, the higher criticism statistic can obviously be used in a wide variety of situations and there is no need for the p-values to be derived from normally distributed Z-scores, for example. Consequently, numerous other settings for its deployment can be considered. We have found that, in a wide variety of settings where one has data which are sparsely nonnull, the HC statistic has an adaptive optimality.”

It is for this reason that we contend using HCT in the context of both our $F$ test and
nonparametric test is appropriate, for even though our p-values are defined in a different manner, they still act similarly to the supervised p-values generated in [22].

4.6 Connecting Phase Diagrams to Clustering Errors

The results in the previous section hold true for supervised datasets, in which class assignments were already known. However, this same phase diagram exists in the case of clustering problems as well, since signal recovery for clustering is closely tied to signal recovery for classification. We follow the arguments of [46] to discuss the connection between the phase space transition in the supervised and unsupervised contexts, based on the Hamming distance as a measure of true clustering error.

Recall the definition for the Hamming distance between cluster assignments $C$ and the true (typically hidden or unknown) classes $\tilde{C}$, dependent on $p$:

$$H_p(C, \tilde{C}) = \sum_{i=1}^{n} 1\{C(i) \neq \tilde{C}(i)\}$$

The Hamming distance can be considered as the number of misclassified instances. We consider the ideal Hamming distance to be the minimax of the expected value of this sum, i.e.:

$$H^*_p = \inf_C \{E(H_p(C, \tilde{C}))\}$$

(4.44)

We now define the ideal Hamming distance based on the parameters $r$ and $\beta$, as laid out in the following theorem (first presented in [46]):

**Theorem 7 ([46]).** Let $L_p > 0$ denote a multi-log($p$) term that satisfies $L_p p^\delta \to \infty$ and $L_p p^{-\delta} \to 0$ for any $\delta > 0$ as $p \to \infty$. Then fixing $\beta \in (0,1)$ and $r > 0$, consider a model $ARW(\beta, r)$. As $p \to \infty$:

$$H^*_p(\beta, r) \begin{cases} \geq L_p p^{1-(\beta+r)^2/(4r)} & r > \beta \\ \sim p^{1-\beta} & 0 < r < \beta \end{cases}$$
where \( H_p^* (\beta, r) \) represents the ideal Hamming distance.

Recall the standard phase function representing the boundary for full classification recovery in the case of uncorrelated noise, defined in the previous section as \( \rho^*(\beta) \):

\[
\rho_{\text{exact}}^*(\beta, \Sigma = I_p) = (1 + \sqrt{1 - \beta})^2 \quad 0 < \beta < 1
\]

Additionally, in the ARW model, we know that for the set of important features:

\[
S_r = \{1 \leq j \leq p : \mu(j) \neq 0\}
\]

where:

\[
|S_r| \sim p\epsilon_p = p^{1-\beta}
\]

Then the connection between the Hamming distance \( H_p^* \) and the phase space transition holds through basic algebra; namely, the following relations hold [46]:

- When \( r > \rho_{\text{exact}}^*(\beta, \Sigma = I_p) \), \( H_p^*(\beta, r; I_p) = o(1) \) and it is possible to fully recover \( S_r \) with overwhelming probability.
- When \( \beta < r < \rho_{\text{exact}}^*(\beta, \Sigma = I_p) \), \( 1 \ll H_p^*(\beta, r; I_p) \ll p^{(1-\beta)} \) and it is possible to recover most of the support \( S_r \), but not fully recover all of \( S_r \).
- When \( 0 < r < \beta \), \( H_p^*(\beta, r; I_p) \sim p^{(1-\beta)} \) and it is nearly impossible to delineate \( S_r \) from the entire feature space.

These characteristics hold when \( \Sigma = I_p \); a similar result holds in the case for all \( \Sigma \) where \( \rho_{\text{exact}}^*(\beta, \Sigma = I_p) \) is replaced by the more general function \( \rho_{\text{exact}}^*(\beta, \Sigma) \); authors in [47] present an argument for the minimax optimality of the Hamming distance in this case through a graphlet screening approach. In many cases, there is a constant \( c = c(\beta, r, \Sigma) \) depending on \((\beta, r)\) such that

\[
H_p^*(\beta, r; \Sigma) = L_p p^{1-c(\beta, r; \Sigma)}
\]
where examples include block-wise diagonal covariance and long-memory time series. In these cases, [47] provide precise versions of $\rho_{exact}^*(\beta, \Sigma)$ and show that these are equivalent to providing minimum error within the boundaries of the phase space.

As a result, the supervised regions for which HCT is optimal are equivalent to regions for which HCT will be optimal in unsupervised settings, under mild conditions. Consequently, even in the case of unsupervised data, HCT provides optimal thresholding performance.

4.7 From Ordinary HCT to Unsupervised HCT

In this section, we connect the standard, ordinary definition of HCT as presented in Chapter 1 and the beginning of this section (which we will refer to as OHC) to the definition of HCT that is implemented in Chapter 2 and Chapter 3 (which we will refer to as “unsupervised HC” or UHC). Our results reveal that these thresholds act similarly and thus will share the theoretical properties outlined in the previous sections.

We start with an argument presented in [23]. They prove that the OHC functional and the ideal functional are specifically designed to choose the threshold $t$ that maximizes the proxy separation relative to $\epsilon$ and $\tau$, specifically defined in [23] for the supervised case as:

$$\tilde{\text{Sep}}(t; \epsilon, \tau) = \frac{\epsilon \mu_\tau E(w(j) | \mu = \mu_\tau)}{\sqrt{\epsilon E(w^2(j) | \mu = \mu_\tau) + (1 - \epsilon) E(w^2(j) | \mu = 0)}}$$  \hspace{1cm} (4.45)

Where $w(j)$ represents the weight of feature $j$, as determined by hard thresholding (in our case). When $\epsilon$ and $\tau$ are fixed and known, this separation measure is maximized by the LRT; in a supervised context, when $\epsilon$ and $\tau$ are unknown but the classes for each sample are given, this separation measure is maximized using OHC.

In a similar vein, authors in [49, 48, 89, 90] explore the post-feature-selection signal to noise ratio (SNR), which can be seen as a functional analogous to the separation measure above in an unsupervised context. We follow the arguments in [90] to discuss the details of this functional, and how it leads to superior clustering quality.
Note that the thresholding that we consider here is known as \textit{hard thresholding}; that is, we keep the signal of important features whose values lie above the threshold, and set all other feature signals to 0. In this respect, we consider the \textit{empirical survival function} of any test statistic \( \Omega \) under the above restrictions, where:

\begin{equation}
G_p(t) = \frac{1}{p} \sum_{j=1}^{p} 1\{ \Omega(j) \geq t \} \tag{4.46}
\end{equation}

and define the expected value of this function given set of cluster centers \( \mu \) and cluster assignments \( y \) as:

\begin{equation}
G(t) = G(t, \mu, y) = E[G_p(t)|\mu, y] = \frac{1}{p} \sum_{j=1}^{p} P\{ \Omega \geq t|\mu, y \}
\end{equation}

. Finally, if we define:

\begin{equation}
W_p(t) = W_p(t, \mu, y) = \frac{1}{p} \sum_{j=1}^{p} \mu^2(j) P\{ \Omega \geq t|\mu, y \}
\end{equation}

We then define the post-selection signal to noise ratio as follows:

\begin{equation}
SNR(t) = \frac{pW_p(t)}{\sqrt{pG(t)/n + pW_p(t)}}
\end{equation}

While this scenario holds for the most generic setup, the most interesting scenario in which feature selection is truly advantageous is under “rare-weak” settings as characterized by \( ARW(r, \beta) \).

If we define the values \( g(t) = g(t, \epsilon, \tau, n) \) and \( w_p(t) = w_p(t, \epsilon, \tau, n) \) to be the counterparts of the functions \( G(t) \) and \( W_p(t) \) in the case of the rare-weak model, with some regularity as \( p \to \infty \), we can define a \textit{pseudo-SNR snr} \[ ? \]:

\begin{equation}
snr(t) = \frac{w_p(t)}{\sqrt{g(t)/n + w_p(t)}} = \frac{\sqrt{n}w_p(t)}{\sqrt{g(t) + nw_p(t)}} \tag{4.47}
\end{equation}
We argue that under moderate conditions, $SNR(t) \approx \sqrt{p} \cdot snr(t)$. The ideal threshold can be chosen as the threshold that maximizes this pseudo-SNR.

If we define the function $G_0(t, n)$ as the survival function of $\Omega(j)$ when $\mu(j) = 0$, and similarly we define the function $G_{\tau_p}(t, n)$ as the survival function of $\Omega(j)$ when $\mu(j) = \tau_p$. Consequently, $G_0(t, n)$ represents the survival function of $\Omega(j)$ for uninformative features, while $G_{\tau_p}(t, n)$ represents the survival function of $\Omega(j)$ for important features that contain pertinent clustering information. The overall survival function is thus given by $G = \epsilon G_\tau + (1 - \epsilon)G_0$. Here, we note that in the case of an ARW model:

$$g(t) = (1 - \epsilon_p)g_0(t, n_p) + \epsilon_p g_{\tau_p}(t, n_p)$$

and

$$w_p(t) = \frac{\epsilon_p \tau_p^2 g_{\tau_p}(t, n_p)}{\sqrt{n_p}}$$

Thus, the post-selection signal to noise ratio can be defined in rare-weak settings as:

$$snr(t) = \frac{\epsilon_p \tau_p^2 g_{\tau_p}(t, n_p)}{\sqrt{(1 - \epsilon_p)g_0(t, n_p) + \epsilon_p(1 + \sqrt{n\tau_p^2})g_{\tau_p}(t, n_p)}}$$

Comparing this function, defined in an unsupervised setting, to the function for the separation in the supervised setting given in equation 4.45, we can argue that these two criteria are relatively similar, and consequently the threshold that is chosen to maximize the post-selection $snr$ should act similarly to the threshold that maximizes the separation measure in a supervised setting. In particular we see that both provide a measures of separation between classes, whether known or unknown. We therefore choose a threshold that maximizes the $snr$ in an unsupervised setting, and argue that in doing so we are finding a threshold that will act similarly to an appropriate threshold in a supervised setting.

An additional argument for the usefulness of the $snr$ as a quality measure is given in [Jin/Wang]. They show that the $snr$ is a direct result of the singular value decomposition (SVD) of the post-selection covariance matrix $W(t) = X_t^T X(t)$ based on a threshold $t$. In
particular, if we define \( U(t) \) to be the left singular vector of \( W(t) \), then authors in [49, 48] argue:

\[
U(t) \propto \text{snr} \ast U + z + \text{rem}
\]  

(4.51)

where \( U \) is a left singular vector of \( W = X^T X \), \( z \) represents measurement noise and \( \text{rem} \) is a small-order remainder term. These authors then argue that maximizing the \( \text{snr} \) is equivalent to maximizing the left singular vectors, which they subsequently prove lead to optimal clustering in the case of spectral clustering. We extend their argument in our case by considering the fact that maximizing the \( \text{snr} \) provides the ideal separation of clusters in the case of spectral clustering, it should also naturally provide an ideal separation in the context of k-means clustering, which we use here.

As it turns out, the UHC functional results in a threshold that maximizes the \( \text{snr} \) as defined above. We mirror the arguments laid forth by [49, 89, 90] to draw the connection between the higher criticism threshold and the ideal threshold that maximizes the \( \text{snr} \).

The ideal UHC functional is defined for any survival function (right-tailed p-value) \( G = P(\Omega > t) \) as:

\[
UHC_{\text{ideal}}(t, G) = \frac{G - G_0}{\sqrt{G + \max\{\sqrt{n}(G - G_0), 0\}}}
\]  

(4.52)

In rare-weak scenarios, we assume \( G = \epsilon G_\tau + (1 - \epsilon)G_0 \), \( G_\tau = G_{\tau_\mu}(t) = P(\Omega > t|\mu = \mu_0) \), and \( G_0 = G_0(t, n) = P(\Omega > t|\mu = 0) \). Again, \( G_\tau \) represents the empirical survival function of informative features, and \( G_0 \) represents the expected value of the survival function of \( \Omega(j) \) for uninformative features. Therefore:

\[
UHC_{\text{ideal}}(t, G) = \frac{\epsilon(G_\tau - G_0)}{\sqrt{\epsilon G_\tau + (1 - \epsilon)G_0 + \sqrt{n}\epsilon(G_\tau - G_0)}}
\]  

(4.53)

In area of interest, \( G_\tau >> G_0 \) and so:

\[
UHC_{\text{ideal}}(t, G) = \frac{\epsilon G_\tau}{\sqrt{\epsilon(1 + \sqrt{n})G_\tau + (1 - \epsilon)G_0}}\left(1 + O\left(\frac{G_0}{G_\tau}\right)\right)
\]  

(4.54)
Recall the SNR definition:

\[ snr = \frac{\epsilon\tau^2 G_\tau}{\sqrt{\epsilon(1 + \sqrt{n} \tau^2)} G_\tau + (1 - \epsilon) G_0} \]  

(4.55)

Comparing two equations above, we have: \( \tau^2 UHC_{\text{ideal}}(t, G) \propto snr \) and so

\[ t^{snr} \approx t^{UHC_{\text{ideal}}} \]  

(4.56)

Now, we note that our version of UHC as presented in Chapters 2 and 3 mimics the ideal UHC functional. Recall our p-value definition of UHC, defined as:

\[ UHC(t, \pi(k)) = \frac{k/p - \pi(k)}{\sqrt{k/p + \max\{\sqrt{n(k/p - \pi(k))}, 0\}}} \]  

(4.57)

This is exactly the ideal UHC functional when \( G = k/p - \pi(k) \), and \( G_0 \approx 0 \). In fact, it is proven in [Wang]:

\[ |UHC_{\text{ideal}}(t, \pi(k)) - UHC(t, \pi(k))| \leq \frac{L_p}{\sqrt{p}} \]  

(4.58)

for some \( L_p \) representing a generic multi-log \( p \) term, with error on the order \( o(1 - 1/p^2) \). Therefore with overwhelming probability we have:

\[ t^{UHC} \approx t^{UHC_{\text{ideal}}} \]  

(4.59)

Combining results we are finally left with:

\[ t^{UHC} \approx t^{UHC_{\text{ideal}}} \approx t^{snr} \]  

(4.60)

And thus the threshold obtained through UHC mimics the ideal threshold. Consequently, theoretical properties surrounding the usefulness of HCT hold in our case, and using UHC in both our parametric and nonparametric methods becomes a useful thresholding choice for optimal feature selection in the case of rare-weak settings.
Chapter 5

Feature Selection Applications: Copy Number Data vs. Gene Expression Data

Feature selection plays a prominent role in cancer research. Scientists are interested in which genetic mutations and pathways are associated with different cancers and, more specifically, which mutations are linked to different subtypes of cancers. Knowing this information allows cancer researchers to develop new therapies that specifically target parts of the genome which are crucial to cancer subtype proliferation. Acquiring better knowledge of the genes, pathways, and characteristics of tumor growth can lead to more targeted and less-invasive therapies.

Currently, most feature selection methods used in bioinformatics determine specific genes or parts of the chromosome that are affected by cancer based on only one level of information. Generally, these methods deal with datasets that involve copy-number variation, SNP data, or gene-level expression data. Due to the subtleties that may occur in many diseases, however, establishing predictive information from one data level is not sufficient in determining cancer subtypes in some instances. Furthermore, using only one type of data restricts the type of feature selection methods available, making results between different data levels incomparable.

In this section, we investigate this problem in detail by generating and using a gene-level data platform that allows us to compare results between copy number data and gene expression data. We begin by discussing the advantages (and possibly disadvantages) of different data types. We then evaluate the performance of the novel bimodal feature
selection method on the gene-level integrated data. Results from the analysis, as well as implications of these results, are discussed in the final section of this chapter.

5.1 Explanation of Data Types and Integrative Analysis

5.1.1 Gene Expression Data

Gene expression (GE) data is the most common type of data used in bioinformatics. This data type represents a gene-level measurement that is associated with the transcription of DNA into mRNA. Typically, gene expression datasets provide information for whole transcriptome analysis, and can contain $1 - 2 \times 10^4$ measurements per sample. Gene expression data is used in the study of cancer because it is well-known that genetic mutation results in cancer proliferation. Therefore, studying the gene-level expression of both cancerous and normal cells can help to identify the specific genetic mutations that are linked to certain cancer types and their subsequent subtypes. [81]

There are some drawbacks to using gene expression data. For one, it is generated from mRNA versus being measured from the DNA itself. Consequently, alterations in the data that may not be representative of actual DNA changes can occur because it acts as a complemented copy of the DNA. Additionally, the purification of mRNA data is hard to verify, since contaminations are always possible in the transcription phase. [81] Therefore, measuring expressions at a gene level can sometimes be particularly noisy, since individual genes can be quite variable across different subjects.

Data extraction presents another difficulty associated with gene expression data. Different types of technology may provide measurements at different probe locations along the genome. Consequently, the resulting data does not always represent a one-to-one gene correspondence between probe and gene. Moreover, as newer chip versions are developed, the subsequent data collection process may change, leading to new chip readings that are inconsistent with older versions. Most labs which process this type of data are aware of these complications, however, and do their best to mitigate the measurement errors among
samples. The design and normalization methods carried out to obtain such data ensure that there is a gene-to-feature correspondence with high probability in the case of array measurements. [81]

### 5.1.2 Copy Number Variation Data

Copy number variation (CNV) data is the second-most common type of biological data. Copy number data has become a more recent focus in the area of cancer research, because it contains measurements that occur at the DNA or chromosomal level. Researchers have discovered that “in every known instance of cancer, individual genes may well contain mutations, but entire chromosomes, which carry thousands of genes, are also severely scrambled - duplicated, broken, structurally rearranged or missing entirely...growing evidence suggest that this chaos on the chromosomal level is not just a side effect of malignancy, but also the direct cause and driving force of cancer.” [24] In some cases, there are features of cancers that cannot be explained by gene-level mutation, yet are directly linked to chromosomal changes.

CNV data is composed of the copy number changes in a chromosomal segment. These changes can be representative of additions, mutations, or deletions in particular locations on the chromosome. Each segment contains many genes, and thus CNV measurements are usually generated as an average change over a set of probes that are located at regular intervals on the Chromosome. While raw CNV data typically contains measurements from approximately $2 \times 10^5$ probes, there are typically fewer ($2 \times 10^3$) averaged CNV measurements. Due to the size of the raw probe measurement data, CNV data also encounters the dimension glut issue, and requires dimension reduction methods in order to properly identify clusters, classes, and/or subtypes.[24]

The biggest drawback to using CNV data is its high-level versus detailed functionality. Because the data gathered at the DNA level represents an averaged signal from a region, which may contain a large number of genes, it is typically hard to identify exactly where the mutation originates. Contrastingly, individual probe measurements offer a much too
detailed approach to provide any significant or useful clustering information. Therefore, the use of CNV data alone makes it extremely challenging to locate a “driver gene” or set of driver genes for the cancer in question. This is particularly important when developing diagnosis as well as therapy treatments, which interact with the significant genes and disregard those that are not associated with different cancer types.

5.1.3 The Importance of Integrated Data

Methods relying solely on single data types have been developed in cancer classification or clustering problems. However, the construction of most methods is dependent on the data type, which limits their adaptability and usage in comparison among platforms. Additionally, methods based on a single data type do not fully capture the subtle abnormalities present in certain cancer types and subtypes. Therefore, integrated data should be considered for future analysis, and methods capable of handling multiple data types and data levels should be developed. The hope is that using both high-level and detailed information about patients will lead to more targeted and more reproducible diagnosis and gene therapy.

This concept has been difficult to deliver. To date, very little attention has been paid to generating methods to compare models that operate on different data platforms, and even fewer researchers have looked at ways to combine multi-level data platforms in an efficient manner. [58]

Here, we consider a multi-level analysis based on “gene level copy number” data. We investigate three case studies based upon which we obtain both GE level data and CNV data for each sample. We then match this data at the gene level in a manner described in [50], generating two uniquely related CNV and GE datasets. In particular we consider our typically gene expression dataset, and a mirrored “gene-level CNV dataset” where the copy number probe measurements are averaged and matched at the particular gene locations on the chromosome. Once the 2 datasets have been merged and produced, we apply both the parametric and nonparametric feature selection methods to the integrated
data, and compare the results. In each study, we used the KDE variance method for our feature ranking as described in Chapter 3, and found a threshold based on unsupervised higher criticism. Bimodal and multimodal KDE refer to this method. Clustering was done using k-means clustering. To our knowledge, this is one of the only attempts to establish a method of comparison for feature selection on the same dataset at multiple measurement levels, and one of the very few that investigate and try to profit from the relationship between copy number and gene expression data.

5.2 Case Study: Breast Cancer Dataset

We consider a dataset that contains both copy number and gene-expression data for a subset of \( n = 37 \) samples. The dataset is a preprocessed dataset presented in [64], originating from samples provided in The Cancer Genome Atlas (TCGA), which is a widely-used and easily-accessible data platform. The datasets contain 6,095 matched copy-number and gene-expression measurements for each sample. Sample information details that were used for clustering, including ER status (+/−) and tumor grade (1,2,3), were also reported in [64]. Both gene expression and copy number measurements were normalized before FS was implemented; error rates were measured based on the Hamming distance between recorded ER status/tumor grade, and the cluster assignments found via k-means clustering. Additionally, since these datasets were created so that CNV and GE were matched, we evaluate the consistency of the FS algorithms, to see whether similar features and locations were identified as significant in ER status and tumor grade identification.

5.2.1 Results

Results from both the two-class and three-class case are shown in the following figures. Overall, it looks like clustering errors are similar when clustering on the full set of data when either data type (GE vs CNV) is used. However, errors in the 2-class case are significantly lower than the 3-class case. We compared our FS method, both parametric
and nonparametric, to the performance of the IF-PCA method which clusters based on the KS statistic. In both the two-class and three-class scenarios, the new methods outlined in this thesis slightly outperform IF-PCA in both types of data.

According to our results, the nonparametric FS approach outperforms others in the case of GE data, and yet the bimodal F test works better in the case of CN data. This is an indication that the normally-distributed noise assumption might be more suitable in the case of CNV data.

Figure 5.1: Case Study: KS vs F vs KDE for GE/CNV data

<table>
<thead>
<tr>
<th>2 class case: ER status (+/-)</th>
<th>Full Error</th>
<th>KS statistic</th>
<th>Bimodal F test</th>
<th>Bimodal KDE test</th>
</tr>
</thead>
<tbody>
<tr>
<td>GE data</td>
<td>10/37</td>
<td>0.27027</td>
<td>14/37</td>
<td>0.378378</td>
</tr>
<tr>
<td>CN data</td>
<td>12/37</td>
<td>0.324824</td>
<td>8/37</td>
<td>0.216216</td>
</tr>
</tbody>
</table>

Table: Lowest Error and Features Retained

| GE data                     | 73/228     | 0.32017544   | 0.34%          | 24.74%          |
| CN data                     | 108/228    | 0.47355421   | 0.65%          | 6.09%           |

Figure 5.2: Case Study: KS vs F vs KDE for GE/CNV data

<table>
<thead>
<tr>
<th>3 class case: tumor grade (1,2,3)</th>
<th>Full Error</th>
<th>KS statistic</th>
<th>Multimodal F test</th>
<th>Multimodal KDE test</th>
</tr>
</thead>
<tbody>
<tr>
<td>GE data</td>
<td>23/36</td>
<td>0.039999</td>
<td>10/36</td>
<td>0.04444444</td>
</tr>
<tr>
<td>CN data</td>
<td>21/36</td>
<td>0.583333</td>
<td>18/36</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table: Lowest Error and Features Retained

| GE data                     | 16/36      | 0.44444444   | 0.46%            | 30.43%             |
| CN data                     | 16/36      | 0.44444444   | 0.05%            | 23.81%             |

Overall, the two feature selection methods shown here were able to reduce the error in the datasets by an average of 21%. More importantly, these error reductions were generated using a reduced dataset that contained less than 1% of the original features, in all cases. Overall, FS provided a greater improvement in cluster identification in GE vs CNV data. It is unclear from these results alone, however, whether there is a significant difference in the amount of features that are selected when using GE vs CNV data. Overall, based on
these results, it can be argued that FS is extremely useful in both types of data, although it might be more useful in the case of gene expression data.

5.3 Case Study: Glioma Dataset

The second dataset included in our investigation contained \( n = 229 \) samples of subjects who were diagnosed with a particular type of aggressive malignant glioma, known as glioblastoma. Our investigation in this case was based on both copy number and gene expression measurements, which had a total of 8,569 features once gene-location matching was performed, as first presented in [11]. The clusters that were assumed present in this case were the three previously-identified subtypes of this type of cancer, known as Proneural, Neural, and Mesenchymal. An additional study was done to compare the particular Mesenchymal subtype vs others, which reduced the problem to a two-class scenario.

5.3.1 Results

Feature selection resulted in approximately 24 \% improvement and 6\% improvement in subtype clustering using GE and CNV data, respectively - when compared to clustering the entire dataset. These results are similar to improvements in the breast cancer dataset in the three-class case using GE and CNV datasets. Again, < 1\% of the original features were retained in both datasets for use in clustering. In this particular study, it seems that KDE feature selection proved to be the most advantageous unsupervised method. However, in the case of copy number data, supervised classification (i.e. feature selection when subtypes are previously defined) clearly outperforms other unsupervised methods.
Other studies have suggested specific driver genes for the three identified subtypes of glioma present in this dataset. [13, 11] In particular, researchers have identified 27 significant genetic alterations that are consistent with glioblastoma subtype mutations. Due to data restrictions, only 24 of them were included in the full feature list - 3 were removed during the data cleaning and matching process. Comparing this list of driver genes to the list of selected features generated from both types of datasets. In general, we found that individually, FS using GE data identified 8 out of 24 of these known “driver genes” (approximately 33%). Similarly, FS using CNV data identified 11 of these same genes, closer to 46%.

One of the most interesting findings of this study is that there is little overlap between these target gene identifications via feature selection based on the two distinct data types. In particular, we find that even though each data type identifies less than half of the significant driver genes, the combined driver genes that are identified in either type jumps
to 60%, identifying 14 of the 24 driver genes. When considering only the list of driver genes that are known to be mutated in more than 10% of a typical set of samples, this identification percentage jumps to 73% (11 out of 15). Additionally, the genes that are identified in both datasets include the genes that seem to be significantly altered most often in all types (and particularly classical versions) of glioblastoma, namely $KLHK9$, $EGFR$, $CDK4$, and $MDM2/4$ [11]. Other driver genes that are less frequently mutated, but more often related with subtype identification, such as $PDGFRA$, $PIK3R1$, $PIK3CA$, and $CEBP\,D$ [13] are typically identified in one data type but not in the other. This result suggests that further study involving the integration and simultaneous use of both data level types might be more significant in identifying driver genes in the future. While GE and CNV are linked in some way, it seems that both levels contain different types of information that may be important in identifying cancer subtypes, and therefore using a merged version of these data types might further improve clustering quality and subsequent feature selection. Additionally, identifying pathway networks of genes that contain such driver genes might also provide insight in the proliferation of different cancer subtypes.

5.4 Large Multi-Class Case Study: METABRIC Dataset

The final, large-scale dataset included in our investigation comes from one of the largest compiled datasets for breast cancer, as first introduced in [15]. The dataset was compiled from multiple studies, and contained $n = 939$ samples (after cleaning) of subjects who were diagnosed with breast cancer. Our investigation was again based on both copy number and gene expression measurements, with 42,824 features once gene-location matching was performed, making this dataset by far the most comprehensive dataset that we considered. In addition, this dataset in particular was matched and cleaned based on probe location, making this the closest version of “integrated” data available. The clusters that were assumed present in this case were four previously-identified breast cancer subtypes, known as Basal, Her2, Luminal A, and Luminal B. Matched-normal and a smaller, lesser-known
subtype known as NC were also available in the dataset, but were removed before clustering was performed.

### 5.4.1 Results

Based on our results, it seems that clustering using the full dataset based on gene-level copy number results in approximately 20% improvement over clustering based on gene expression alone. However, it is clear that using all features leads to significant clustering errors independent of the data collection method. Feature selection seemed to result in approximately 42% improvement and 38% improvement in subtype clustering using GE and CNV data, respectively, using the bimodal F test. These results show a drastic improvement over clustering using the full dataset. In all cases, < 10% of the original features were retained in both datasets for use in clustering. Note that in this case, it seems that the gene-level copy number dataset provided marginally smaller clustering errors than the gene expression data. In particular, our bimodal F test seemed to outperform other methods in terms of clustering error, showing its use in the case of subtype discovery.

Figure 5.5: Case Study: KS vs F vs KDE for METABRIC data

<table>
<thead>
<tr>
<th></th>
<th>Full Error</th>
<th>KS statistic</th>
<th>Multimodal F test</th>
<th>Multimodal KDE test</th>
</tr>
</thead>
<tbody>
<tr>
<td>GE data</td>
<td>498/939</td>
<td>0.530351</td>
<td>310/939</td>
<td>0.330138</td>
</tr>
<tr>
<td># feat</td>
<td>42875</td>
<td>100%</td>
<td>1601</td>
<td>3.78%</td>
</tr>
<tr>
<td></td>
<td>289/939</td>
<td>0.3077742</td>
<td>313/939</td>
<td>0.333333333</td>
</tr>
<tr>
<td>CN data</td>
<td>406/939</td>
<td>0.432375</td>
<td>509/939</td>
<td>0.542066</td>
</tr>
<tr>
<td># feat</td>
<td>42875</td>
<td>100%</td>
<td>458</td>
<td>1.07%</td>
</tr>
<tr>
<td></td>
<td>250/939</td>
<td>0.2652407</td>
<td>3797</td>
<td>8.86%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Multimodal F test</th>
<th>% Features Retained</th>
<th>% Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>GE data</td>
<td>289/939</td>
<td>0.54%</td>
<td>41.57%</td>
</tr>
<tr>
<td>CN data</td>
<td>250/939</td>
<td>7.26%</td>
<td>38.42%</td>
</tr>
</tbody>
</table>

### 5.5 Conclusions

The results in all three dataset investigations demonstrate that feature selection is useful in both types of data settings. However, the advantage of using one data type vs the other for
clustering purposes is unclear. Our first and third case studies indicate that copy number data provides better clustering results; the second case study has better predictive quality when gene expression data is used. Additionally, it seems as if FS in the context of copy number data selects significantly fewer features than when using gene expression data in the first two case studies, and yet a significantly larger number of features are retained in the copy number dataset in the third study. Regardless of the data platform, it is clear that FS will provide an equal or better clustering than when all features are included.

It is important to note that even integrating two levels of biological data does not leave us with pristine results because, especially in the case of tumor analysis, even cancer cells within the same tumor may have different combinations and alterations at both the DNA and RNA levels. This inherent variation causes consistency issues in analysis, regardless of the data type or FS method that might be implemented, and is one reason why the problem of identifying, isolating, and correcting the damage that cancer causes continues to be an intractable problem.

These results imply that additional information can be obtained by simultaneously leveraging information from both data types. In fact, parallel analysis of DNA copy number changes and mRNA levels has been known to provide insight in subtleties that may not be apparent in one level of data. [44, 57, 58, 59] Previous studies have also discovered that DNA copy number changes are strongly correlated to changes in expression levels in cancerous cells, indicating that simultaneous analysis of both data levels would be useful in studying the progression of human cancers. Evaluating CNV data directly from the DNA allows us to reduce the level of noise in the data, while evaluating at the gene expression level allows for us to pinpoint driver genes that are significant in the proliferation of specific cancers.
Chapter 6

Further Improvements: Iterative Clustering

We consider a third method that iterates FS and clustering until convergence. Unlike the models described above, which perform FS as a preprocessing filter before clustering, this method combines feature selection and clustering to find a reduced set of important features. The intuition behind this iterative algorithm is that by assigning labels based on an initial clustering and then using these labels in feature selection, we can take advantage of many supervised methods for clustering in an unsupervised setting. Instead of filtering out all seemingly unimportant features in one step, the iterative method allows a measurement of clustering performance at multiple steps of feature screening. FS is performed based on a statistic similar to the one used in supervised higher criticism, with relatively fewer features eliminated at each step before clustering. Then the data is re-clustered using the reduced dataset, and new cluster assignments are found. Iterations are repeated until the resulting clustering is unchanged and clustering error is at a minimum.

An overview of the algorithm is as follows:

1. Given a dataset $X_{[n \times p]}$, perform k-means clustering to parse the dataset into the desired number of clusters. Assign each sample to a class $Y$ based on the resulting clustering.

2. Using both $X_{[n \times p]}$ and $Y_{[n \times 1]}$, establish a ranking of feature importance to the given classification. Remove $p'$ features that are least important to class identification (i.e. remove the $p'$ features with the lowest ranking).

3. Generate a new dataset $X_{[p' \times p]}$ with the newly-reduced set of features (i.e. the set of
features with $p'$ removed). Cluster this dataset and establish a new set of class labels $Y$.

4. Measure the cluster quality of the new reduced dataset compared with clustering from the previous step. 4. Repeat steps 1-4 until convergence; i.e., when minimum clustering error is reached.

This iterative process of FS takes advantage of supervised feature selection, and allows us to apply these methods in an unsupervised context. Additionally, this method provides a natural threshold for FS, due to its iterative nature; we select features that are important for clustering until cluster convergence is reached, and error is at a minimum. This iterative process performs feature selection surprisingly well, despite a longer run time. Performance of the method is tested on both real and simulated datasets in the subsequent sections. Results from the iterative method are then compared to competitive supervised models as well as the filter versions of the FS algorithms described earlier.

6.1 Description of Model

The method that we use here capitalizes on the ranking method outlined in [22] to rank features in a supervised context, but does so iteratively and in an unsupervised realm. Because this supervised method is developed for the two-class scenario, we assume a case where $k = 2$. Note that we could easily consider other supervised methods and a larger number of clusters, although they are not directly considered here. This method follows the following steps:

1. Given a data matrix $X_{[n \times p]}$, cluster the $n$ samples into $k = 2$ clusters using k-means clustering. Assign classes ($\pm 1$) based on the resulting clustering.

2. Construct a z-score based on the two-sided test $H_0 : \text{Cov}(X,Y) = 0$. That is, for each feature $j$, find $z(j) = n^{-1/2} \sum_i Y_i X_i(j)$. 

3. normalize the z-scores by the function $z(j) = \frac{z(j) - \bar{z}}{\hat{\sigma}(z)}$, where $\bar{z}$ and $\hat{\sigma}(z)$ are the mean and standard deviation of all $p$ z-scores.

4. Construct a p-value for the test based on the normalized z-scores, i.e., $p(j) = P(|N(0, 1)| > |z(j)|)$.

5. Rearrange the p-values into a list of increasing p-values $\pi$ such that $\pi(1) \leq \pi(2) \leq \ldots \pi(p)$; Use this list to discard $\delta \times 100\%$ of features with the largest p-values (therefore discarding those features that are least likely to be significantly related to the cluster assignments).

6. From the remaining $r$ features, construct a reduced data matrix $X_{[n \times r]}$; re-cluster the $n$ samples into $k = 2$ clusters using the reduced data matrix, and assign new classes $\pm 1$ based on the resulting clustering.

7. Compare the new clustering with the previous clustering based on a quality measure $Q(i)$, and determine the improvement in clustering quality.

8. Repeat steps 2-7 until convergence; i.e., until the quality of clustering is optimized; i.e., produces minimum clustering error on the smallest feature set.

The selection of $\delta$ is important in this process, as it will be directly related to the number of features that are removed in each step, and consequently has an effect on the computational run time and number of iterations needed for convergence to be reached. In the simulations, we adopt an $\delta = 0.1$ for selecting features, as we have found that this provides a good balance between run time and feature selection accuracy. However, it should be noted while $\delta$ has a drastic effect on run time and number of iterations, performance results for this method are actually fairly resistant to the choice of $\delta$. In fact, a choice of $\delta$ between 0.2 and 0.05 seemed to offer fairly consistent performance in terms of cluster quality, noting that a smaller alpha provides a more accurate choice of the reduced feature set.
The other criterion that is critical to the success of this method is the choice of clustering quality measurement, $Q(i)$. In fact, in most unsupervised problems, this choice is critical, and is often the hardest measurement to pinpoint. A desired quality of clustering is well-separated clusters, typically measured by comparing the some measure of the inter- and intra-cluster distance. The ideal clustering of samples will have small distances to their cluster mean; i.e., a small sum of squares within the same cluster ($SS_w$), and large distances between the mean locations themselves; i.e., a large sum of squares between clusters ($SS_b$). Thus, a natural clustering quality measure is to determine the ratio of these values $SS_b/SS_w$, and choose the feature set for which this measure is maximized.

One problem occurs in the case of high dimensionality, however. A natural dimensional bias exists in this calculation, similar to the dimension bias outlined for k-means in high dimensions in Chapter 4. When the dimensionality is low, this bias is negligible; however, as dimension increases, the bias increases, and thus high dimensions may significantly affect this particular clustering quality. Authors in [43] explain the problem with high dimensional data in the context of separability, as defined as the ratio above: “the separability criterion prefers higher dimensionality; i.e., the criterion value monotonically increases as features are added, assuming identical clustering assignments.” Considering the criterion, this is an intuitive result; the ratio will prefer higher dimensions since data tend to be more scattered as dimension increases. The resulting issue is that the criterion function will still increase, even if new features do not add deterministic value to identifying clusters.

Accordingly, we modify the separability criterion and quality measure accordingly. In particular, we consider the change in the ratio sum of square values, versus the values themselves. We also normalize the separability criterion by dimension. Both of these changes are a direct attempt to try and mitigate the natural bias present in the selection criterion, thus allowing us to compare clustering quality based on datasets of different dimensions.
For each iteration \( \{i : i > 1\} \), the criterion \( Q^{(i)} \) is calculated as:

\[
Q^{(i)} = r^{(i)} \times \Delta \left( \frac{SS_b^{(i)}}{SS_w^{(i)}} \right)
\]

(6.1)

Where \( \Delta \left( \frac{SS_b^{(i)}}{SS_w^{(i)}} \right) \) represents the change in the SS ratio between the \((i - 1)^{th}\) and \(i^{th}\) iteration, and \( r^{(i)} \) represents the number of features or dimension of the data at the current iteration \( i \). Note that if \( Q(i) > 0 \), this signifies an improvement in clustering quality between iteration \( i \) and \( i - 1 \); \( Q(i) < 0 \) represents a decrease in clustering quality. We choose the optimal iteration as the iteration that corresponds with the maximum quality measure, with a corresponding small feature set. In other words, the stopping iteration will be:

\[
i^* = \{i : r^{(i)} < \alpha \text{ and } Q^{(i)} = \max_i(Q^{(i)})\}
\]

This process allows us to choose the iteration that corresponds with the maximum change in clustering quality for a relatively small dataset. Note that this corresponds to the exact iteration that produces maximum clustering quality (i.e. minimal error) and, consequently, ideal feature selection. The reasoning behind this choice is intuitive: we expect clusterings early in the iterations to change dramatically as features are removed, particularly when features are ranked by importance and features with little clustering information are removed. It is for this reason that we only consider clusters that are a maximum for relatively small datasets whose size is depending on \( \alpha_A \). In practice, we chose a conservative value of \( \alpha_A = 0.75 \) in real datasets, meaning that this process will look for an iteration that provides maximum quality when at least 25% of the features are removed. However, we note that in simulated data settings, our choice of \( \alpha_A \) is much more stringent, as we are more certain that significantly fewer features contain clustering information. We ultimately want to choose the iteration for which \( Q^{(i)} \) is maximized while simultaneous retaining the minimum number of features.
This process, while slightly more time-consuming that the ones mentioned in previous chapters, successfully identifies significant features for clustering, and reduces the clustering error to a minimum. Additionally, while we have used a particular supervised feature selection technique, the beauty of this iterative method is that any supervised technique can be altered in this manner. This allows supervised feature selection algorithms to be directly applied to unsupervised problems, a factor which is particularly useful.

6.2 Simulation Study

A simulation study was conducted to test the performance of this method as a feature selection method for clustering. Datasets were constructed in the same manner as described in chapters 2 and 3, with \( p = 2 \times 10^4 \) and \( n \approx 100 \). For each combination of parameter values, the simulation was performed 30 times, and average errors based on Hamming distances were recorded. The results shown below include the average proportion of misclassified samples for both the full dataset and the iterative method. Also included in the results is the average proportion of features selected, and the percent accuracy of feature selection; i.e. how many true features were present in the resulting reduced feature matrix.

These results show the success of feature selection using the iterative method described above. We see that feature selection results in reduced clustering errors in all cases. Fur-
thermore, we see that in every case, over 65% of the true signal features were retained in the reduced dataset, while the size of the reduced dataset contained less than 2% of all features. Additionally, it is clear that errors were reduced by over 30% in all cases; some cases, particular those situations in which signals were rare but relatively strong, saw close to 90% improvement in clustering quality. These results are a clear indication that this iterative method provides superior clustering performance using a select number of features than when clustering on the entire dataset.

6.3 Case Study

We also test the effectiveness of this method on a real-world dataset. In particular, we have tested this method in the case of the Lung cancer dataset as first seen in Chapters 2 and 3. The choice is intentional, as we know that supervised feature selection works well on this dataset.

Results shown in the following figure display the number of features selected and corresponding clustering error (proportion of misclassified samples) at each iteration. The vertical line represents the iteration chosen as the ideal stopping point, as determined by the thresholding method above.
A visual example of the thresholding method based on the quality measure defined above is given in the following figure. Note that in this figure, the quality measurement is plotted as a function of iteration number. The horizontal line represents the value $Q = 0$, which delineates quality improvement from quality degeneration. Finally, the vertical line represents the iteration for which a small number of features are retained and the change in clustering is maximized, and therefore becomes our final choice of threshold.
Finally, we compare our results for clustering on this dataset with the supervised version of HC as defined by [22], as well as our unsupervised methods. Results indicate that this iterative method has superior performance in this case where the bimodal unsupervised methods fail, in the sense that there is minimal error. The drawback of this method compared to the supervised case is that while it provides optimal clustering quality, it does so on a significantly larger dataset.

6.4 Conclusions

Based on the results laid out in this chapter, it is clear that the iterative clustering method is a sufficient and powerful approach to feature selection in unsupervised settings. While this approach is computationally intensive, it allows for supervised methods to be implemented...
in situations that do not contain true class information. This result is extremely useful, as supervised feature selection is a widely studied area, for which many different methods have been constructed.

Results from this method also indicate that performing feature selection in an iterative manner leads to minimal clustering error in an unsupervised realm. In particular, this method performs with equivalent or better performance when compared to the supervised counterparts, or with other unsupervised methods outlined above. The main drawback to this approach is that it keeps a significantly larger number of features in the clustering compared to all other selection methods. However, even though the size of the resulting dataset is relatively large, it is still significantly smaller than the original dataset, allowing over 80% of the features to be removed, and resulting in clustering that is equivalent to the supervised results. It is for these reasons that the iterative feature selection can be considered a useful tool in connecting both supervised and unsupervised learning.
Bibliography


Curriculum Vitae

Contact
Jessica Curtis
Department of Mathematics and Statistics
Boston University
111 Cummington Street
Boston, MA 02215, USA

Education
Boston University PhD candidate, Statistics, September 2010 – present.
Thesis advisor: Mark Kon.

Publications
1. Jessica Curtis and Mark Kon, Class Discovery via Bimodal Feature Selection in Unsupervised Settings. IEEE, August 2015. (Submitted).