UNIVERSITY OF CALIFORNIA
RIVERSIDE

Bayes Neutral Zone Classification in Unsupervised and Semi-Supervised Settings

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Applied Statistics

by

Scott Robert Benecke

March 2012

Dissertation Committee:

Dr. Daniel R. Jeske, Chairperson
Dr. Jun Li
Dr. James Borneman

The Dissertation of Scott Robert Benecke is approved:

_____

_____

_____

Committee Chairperson

University of California, Riverside

ABSTRACT OF THE DISSERTATION

Bayes Neutral Zone Classification in
Unsupervised and Semi-Supervised Settings

by

Scott Robert Benecke

Doctor of Philosophy, Graduate Program in Applied Statistics
University of California, Riverside, March 2012
Dr. Daniel R. Jeske, Chairperson

Neutral zone classifiers allow for a region of neutrality when there is inadequate information to assign a predicted class label with suitable confidence. A neutral zone classifier is defined by classification regions that trade off the cost of an incorrect classification against the cost of remaining neutral. We derive a Bayes neutral zone classifier and demonstrate that it outperforms previous neutral zone classifiers with respect to the expected cost of misclassifications and also with respect to computational complexity. Additionally, we present the scenarios where the previous neutral zone classifiers and the proposed Bayes neutral zone classifier achieve equivalence in both the two-class and three-class setting.

Following the theoretical derivation of the Bayes neutral zone classifier we extend the methodology to both the unsupervised and semi-supervised setting via the EM algorithm for the purpose of developing neutral zone classifiers beyond the supervised setting. Previous versions of neutral zone classifiers have only dealt with the supervised

settings. The discussion of unsupervised and semi-supervised neutral zone classifiers covers both the parametric and nonparametric cases. Simulation studies in both the parametric and nonparametric cases show the improvements that can be obtained by adding labeled data for semi-supervised learning.

The Bayes neutral zone classifier is illustrated with a microbial community profiling application in which no training data is available. In this example we show the benefits obtained over previous neutral zone classifiers. Additionally, a simulation study is performed to investigate the benefits of using neutral zone classification to remove noise from microbial community profiling data sets.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1. Neutral Zone Classification

Classification is a procedure in which the attribute variables of an object are used to assign the object a class label. A classifier is an algorithm that maps the objects to the appropriate class labels. Often a classifier will learn to predict class labels based on a set of labeled training data and will therefore be able to subsequently operate on objects with unknown labels. Accuracy can be of extreme importance to the success and usefulness of classifiers. In many circumstances an incorrect classification can lead to a substantial cost. Misclassifications may arise out of the similarities between the attribute variables of two objects, or alternatively, large variability in the underlying class distributions. These situations can be managed by a classifier that, in addition to predicting class labels, utilizes "no classification" (*N*) as a prediction outcome. A classifier that allows for an *N* classification outcome is called a neutral zone classifier. The advantage of the *N* classification outcome is that it enables the user to minimize the cost of misclassifications by alerting them that more information would be needed in order to confidently assign a specific class label to an observation.

Jeske et al. (2007) developed a two-class neutral zone classifier and Yu et al. (2009) extended this concept to the three-class setting. Their applications were motivated

by a microbial community profiling application (see, for example, Valinsky 2002a and 2002b) where a classification rule was needed to predict whether individual nucleotide probes successfully binded to an rRNA gene based on an observed intensity measurement. For the binding outcomes where there was too much ambiguity for a confident prediction, an *N* (neutral) classification outcome was used because inaccurate predictions of binding outcomes would cause problems in a subsequent clustering analysis of the predicted binding outcomes. The *N* classification was a useable predicted class label, since the clustering analysis was performed on vectors of predicted class labels for each gene.

1.2. Bayes Classification

Here we quickly introduce the Bayes classification methodology, which is utilized extensively in this dissertation. The Bayes classifier is one of the most widely used classification tools and is defined in the following manner. Let $X = (X_1, X_2, ..., X_n)$ be the input vector consisting of n real numbers and let Y be a class label which takes the values $Y \in \{1, 2, ..., K\}$ where K is the number of classes. Then the probability that an observation x belongs to a class y is calculated using Bayes' theorem in the following manner:

$$P(y \mid x) = P(y)\frac{p(x \mid y)}{p(x)}$$

Typically, the $p(x \mid y)$ values are estimated using a training data set and using the equation for $P(y \mid x)$ it is possible to compute the probability that a new observation x is

in class y. Then taking the largest $P(y|x)$ value for the new observation we can assign a class label to x. The prior probabilities $P(y)$ are either assigned subjectively or perhaps estimated from the training data set.

1.3. Structure of the Dissertation

The rest of this dissertation is organized as follows. In Chapter 2 of this dissertation, we improve on the work of Jeske et al. (2007) and Yu et al. (2009) by developing an improved neutral zone classifier with respect to misclassification costs and computational complexity. The improvement in misclassification cost results from developing a neutral zone classifier within a minimum cost Bayes framework, while the improvement in computational complexity results from eliminating the need for a numerical search algorithm to find the classification boundaries. Neutral zone classification in an unsupervised setting is also addressed in this chapter. Unlike supervised classification, there is no training data available in unsupervised classification.

Semi-supervised contexts are where some of the training data has labels and some does not. In Chapter 3 we present simulation studies to illustrate the benefits of utilizing labeled data and unlabeled data together for semi-supervised learning. Chapter 4 addresses whether neutral zone classifiers are useful in reducing the noise in the microbial community profiling data set. This is done via a simulation study. Finally, Chapter 5 summarizes the work presented in the dissertation and discusses potential future work.

Chapter 2

Bayes Neutral Zone Classifiers with Applications to Nonparametric Settings

2.1. Introduction

In this chapter, we improve on the work of Jeske et al. (2007) and Yu et al. (2009) by developing a superior neutral zone classifier with respect to misclassification costs and computational complexity. The improvement in misclassification cost results from developing a neutral zone classifier within a minimum cost Bayes framework, while the improvement in computational complexity results from eliminating the need for a numerical search algorithm to find the classification boundaries.

Another aspect of classification addressed in this chapter is unsupervised classification. Unlike supervised classification, there is no training data available in unsupervised classification (Hastie et al., 2001). Therefore the prediction boundaries for classification must be determined entirely from unlabeled data. The microbial community profiling applications considered in Jeske et al. (2007) and Yu et al. (2009) were able to utilize labeled data. However, alternative data acquisition paradigms exist for that application where there is no labeled data (see Mitra et al., 1999 and Aach et al., 2004). These applications measure whether individual nucleotide probes successfully

bind to rRNA genes through observed intensity measurements. Instead of measuring probe binding events on indexed arrays, the application can utilize polony arrays. In polony arrays, the locations of specific genes are not indexed because it is not feasible to track their locations during the experimental process. As a result, many microbial community profiling applications do not have labeled training data. In this chapter, we also develop a methodology for unsupervised neutral zone classification in this context.

The rest of this chapter is organized as follows. In Section 2.2 and Section 2.3 we derive two-class and three-class Bayes neutral zone classifiers, respectively, and discuss why the proposed classifiers outperform the previously mentioned neutral zone classifiers developed by Jeske et al. (2007) and Yu et al. (2009). In Section 2.4 we develop the methodology for performing neutral zone classification in an unsupervised setting, which has not been previously addressed. Our proposed method for dealing with no training data can be used with any of the neutral zone classifiers we discuss in this paper. Also in Section 2.4, we use data from the microbial community profiling application to compare the results of a nonparametric implementation of the Bayes neutral zone classifier to previously published neutral zone classifiers. Comparisons are made in terms of both misclassification cost and computational efficiency. Finally, Section 2.5 summarizes the work presented in this paper.

2.2. Two-Class Neutral Zone Classifiers

In this section we look at neutral zone classifiers in the two-class setting. The two-class Bayes neutral zone classifier is developed and its relationship to previous neutral zone classifiers is derived.

2.2.1. Previous Work

For a two-class neutral zone classifier we are seeking to classify an observation as either 0, 1 or *N*. Let π_0 and π_1 be the prior probability that an observation belongs to class 0 ($C=0$) and class 1 ($C=1$), respectively, where $\pi_0+\pi_1=1$. Let *Y* denote the observed data, and *y* denote a realization of *Y*. Then the posterior probability of the event $C=1$ is given by $p_1(y)=f_1(y)\pi_1/\left(f_0(y)\pi_0+f_1(y)\pi_1\right)$ where f_i is the conditional density of *Y*, given it belongs to class i. We assume we have sufficient training data to estimate the densities via a suitable nonparametric density estimate (NDE). Yu et al. (2009) defined two-class neutral zone classifiers of the form

$$\hat{C}_{NZ}(y;L)\in\begin{cases}0 & \text{if} \quad p_0(y)-p_1(y)\geq L\\ 1 & \text{if} \quad p_1(y)-p_0(y)\geq L\\ N & \text{if} \quad \left|p_0(y)-p_1(y)\right|<L\end{cases} \tag{2.1}$$

where $L\in[0,1]$ and is a threshold that establishes the classification boundaries between the three outcomes 0, 1 and *N*. Using the relation $p_0(y)+p_1(y)=1$, it is easy to see that the neutral zone region in (2.1) is symmetric about 0.5. The optimal value of L for the

classifier in (2.1) leads to a neutral zone classifier and is determined by minimizing the expected cost with respect to Table 2.1, which is given by

$$\begin{aligned} EC_{NZ}(L) \quad \propto \quad & \pi_0\left[\rho_{10}P\left(\hat{C}=1|C=0\right)+P\left(\hat{C}=N|C=0\right)\right] \\ & +\ \pi_1\left[\rho_{01}P\left(\hat{C}=0|C=1\right)+P\left(\hat{C}=N|C=1\right)\right] \end{aligned} \tag{2.2}$$

where $\rho_{ij}=C_{ij}/C_N$. Since the optimal value of L cannot be solved for directly, a numerical search method is used to determine the minimizing value.

True Class Label	Predicted Class Label		
	0	1	N
0	0	C_{10}	C_N
1	C_{01}	0	C_N

Table 2.1. Asymmetric cost structure in two-class setting.

2.2.2. Two-Class Bayes Neutral Zone Classifier

We now derive a Bayes neutral zone classifier that has both a lower expected cost and less computational complexity than the neutral zone classifier in (2.1). Let EC_B be the expected cost of an arbitrary classifier that assigns labels 0, 1 and *N*, when *Y* falls in the regions R_0, R_1 and R_N, respectively. Consider again the cost structure in Table 2.1. Then,

$$\begin{aligned} EC_B \propto\ 1 & +\int_{R_1}\left[\rho_{10}\pi_0 f_0(y)-\pi_0 f_0(y)-\pi_1 f_1(y)\right]dy \\ & +\int_{R_0}\left[\rho_{01}\pi_1 f_1(y)-\pi_0 f_0(y)-\pi_1 f_1(y)\right]dy. \end{aligned}$$

Let $I_0(y)$ be the term in the integral over R_0 and $I_1(y)$ be the term in the integral over R_1. Using boundary-value conventions consistent with Johnson and Wichern (2007), it follows that the classifier that minimizes this cost, the so-called Bayes neutral zone classifier, has the form

$$\hat{C}_B(y) = \begin{cases} 0 & \text{if} \quad I_0(y) \le \min[0\,,\, I_1(y)] \\ 1 & \text{if} \quad I_1(y) < \min[0\,,\, I_0(y)] \\ N & \text{if} \quad \min[I_0(y)\,,\, I_1(y)] > 0 \end{cases} \tag{2.3}$$

Equivalently,

$$\hat{C}_B(y) \in \begin{cases} 0 & \text{if} \quad \rho_{01} p_1(y) \le 1 \le \rho_{10} p_0(y) \text{ or } \rho_{01} p_1(y) \le \rho_{10} p_0(y) \le 1 \\ 1 & \text{if} \quad \rho_{10} p_0(y) < 1 < \rho_{01} p_1(y) \text{ or } \rho_{10} p_0(y) < \rho_{01} p_1(y) < 1 \\ N & \text{if} \quad \rho_{01} p_1(y) > 1 \text{ and } \rho_{10} p_0(y) > 1, \end{cases}$$

which, noting that $p_0(y) + p_1(y) = 1$ can finally be rewritten as

$$\hat{C}_B(y) \in \begin{cases} 0 & \text{if} \quad p_1(y) \le \min\left(\dfrac{1}{\rho_{01}}, 1 - \dfrac{1}{\rho_{10}}\right) \text{ or } 1 - \dfrac{1}{\rho_{10}} \le p_1(y) \le \dfrac{\rho_{10}}{\rho_{01} + \rho_{10}} \\ 1 & \text{if} \quad p_1(y) > \max\left(\dfrac{1}{\rho_{01}}, 1 - \dfrac{1}{\rho_{10}}\right) \text{ or } \dfrac{\rho_{10}}{\rho_{01} + \rho_{10}} < p_1(y) < \dfrac{1}{\rho_{01}} \\ N & \text{if} \quad \dfrac{1}{\rho_{01}} < p_1(y) < 1 - \dfrac{1}{\rho_{10}}. \end{cases} \tag{2.4}$$

Unlike (2.1), it is clear that the neutral zone in (2.4) is not necessarily symmetric about 0.5, and thus the classifier in (2.1) is not generally optimal. Because (2.4) is the

Bayes neutral zone classifier, it is optimal, and additionally we are able to determine the classification outcome boundaries without the need to use a numerical search algorithm From (2.4) we can see that a neutral zone will exist if and only if $1/\rho_{01}+1/\rho_{10}<1$. When a neutral zone exists, then only the first part of each 'or' condition in (2.4) is pertinent and the classifier can be simplified further. Alternatively, when a neutral zone does not exist, then the two parts of each 'or' condition can be combined and (2.4) simplifies to the standard two-class Bayes classifier with asymmetric cost structure.

2.2.3. Equivalence of Bayes Neutral Zone Classifier

For the two-class case we can show that the neutral zone classifier, originally proposed by Jeske et al. (2007) and the Bayes classifier in (2.4) are equivalent. The form of the neutral zone classifier (Jeske et al., 2007) is:

$$\hat{C}\left(y;l_0,l_1\right)\in\begin{cases}0 & \text{if} \quad p_1\left(y\right)<l_0\\ 1 & \text{if} \quad p_1\left(y\right)>l_1\\ N & \text{if} \quad l_0<p_1\left(y\right)<l_1\end{cases} \tag{2.5}$$

where $\left\{0\le l_0\le l_1\le 1\right\}$ and $\left(l_0,l_1\right)$ are thresholds, determined by the user, that establish the classification boundaries of the three outcomes 0, 1 and N. It is equivalent to state the two-class neutral zone classifier as

$$\hat{C}\left(y;L_0,L_1\right)\in\begin{cases}0 & \text{if} \quad p_0\left(y\right)-p_1\left(y\right)>L_0\\ 1 & \text{if} \quad p_1\left(y\right)-p_0(y)>L_1\\ N & \text{if} \quad p_0\left(y\right)-p_1\left(y\right)<L_0 \text{ and } p_1\left(y\right)-p_0\left(y\right)<L_1\end{cases} \tag{2.6}$$

where $\{L_0, L_1 \in [0,1]\}$ The neutral zone classifier in (2.6) is the form of the classifier we will use for the purposes of this section. To see the equivalence of (2.5) and (2.6) we substitute $p_0(y) = 1 - p_1(y)$ into (2.6) to get

$$\hat{C}(y; L_0, L_1) \in \begin{cases} 0 & \text{if} \quad p_1(y) < \frac{1}{2} - \frac{L_0}{2} \\ 1 & \text{if} \quad p_1(y) > \frac{1}{2} + \frac{L_1}{2} \\ N & \text{if} \quad \frac{1}{2} - \frac{L_0}{2} < p_1(y) < \frac{1}{2} + \frac{L_1}{2} \end{cases}.$$

Letting $l_0 = \frac{1}{2} - \frac{L_0}{2}$ and $l_1 = \frac{1}{2} + \frac{L_1}{2}$ can be rewritten as

$$\hat{C}(y; L_0, L_1) \in \begin{cases} 0 & \text{if} \quad p_1(y) < l_0 \\ 1 & \text{if} \quad p_1(y) > l_1 \\ N & \text{if} \quad l_0 < p_1(y) < l_1 \end{cases}.$$

where we see $\{0 \le l_0 \le .5 \le l_1 \le 1\}$ since $\{L_0, L_1 \in [0,1]\}$ This is simply a reparameterization of (2.5) and yields a nice result where our region for (l_0, l_1) becomes $\{0 \le l_0 \le .5 \le l_1 \le 1\}$ instead of $\{0 \le l_0 \le l_1 \le 1\}$ which cuts the sample space in half.

Now, using the neutral zone classifier in (2.6), we can show equivalence to the Bayes classifier in (2.4). This leads to a direct method of calculating L_0 and L_1 in (2.6). To do this we will make use of **Lemma 1** and **Lemma 2** which allow us to write the two-class neutral zone classifier in an alternative way proposed by **Definition 1**.

Lemma 1

If $p_i(y) > p_j(y)\ for\{i, j = 0,1 \text{and } i \neq j\}$ then $I_j > 0$ if $\rho_j = C_j / C_N \geq 2$, where

$I_j = \pi_i C_j f_i - \pi_i C_N f_i - \pi_j C_N f_j$.

Proof

First note that $p_i(y) > p_j(y)$ implies that $p_i(y) > \frac{1}{2}$ and therefore $\frac{\pi_i f_i}{\pi_i f_i + \pi_j f_j} > \frac{1}{2}$.

Rearranging we have $2\pi_i f_i - (\pi_i f_i + \pi_j f_j) > 0$ and if $\rho_j \geq 2$ then

$$\begin{aligned} I_j &= \pi_i C_j f_i - \pi_i C_N f_i - \pi_j C_N f_j \\ &= C_N \left\{ \frac{C_j}{C_N} \pi_i f_i - (\pi_i f_i + \pi_j f_j) \right\} \\ &\geq C_N \{ 2\pi_i f_i - (\pi_i f_i + \pi_j f_j) \} \\ &> 0 \end{aligned}$$

Lemma 2

If $p_i(y) > p_j(y)\ for\{i, j = 0,1 \text{and } i \neq j\}$ then the neutral zone classifier in (2.6) reduces to

$$\hat{C}(y; L) = \begin{cases} i & if \quad p_i(y) - p_j(y) > L_i \\ N & if \quad p_i(y) - p_j(y) < L_i \end{cases}$$

Proof

Since $p_i(y) > p_j(y)$ this implies that $p_j(y) - p_i(y) < 0$. Since we know $L_j \geq 0$, the neutral zone classifier in (2.6) reduces to

$$\hat{C}(y;L)=\begin{cases} i & if \quad p_i(y)-p_j(y)>L_i \\ N & if \quad p_i(y)-p_j(y)<L_i \end{cases}$$

Lemma 2 shows us that we can write the two-class neutral zone classifier in the alternative form proposed in **Definition 1**.

Definition 1

For $p_i(y)>p_j(y)$ for $\{i, j=0,1 \text{and } i\neq j\}$, let $L_i>0$ be a threshold determined by the user, then

$$\hat{C}(y;L_i)=\begin{cases} i & \text{if} \quad p_i(y)-p_j(y)>L_i \\ N & \text{if} \quad p_i(y)-p_j(y)<L_i \end{cases}$$

Theorem 1

If $\rho_0>\rho_1/(\rho_1-1)$ then given the asymmetric cost structure in Table 2.1, the neutral zone classifier in **Definition 1** and the Bayes classifier in (2.3) are equivalent and the optimal choice of the optimal choice of (L_0,L_1) is $\left(L_0^*=1-\frac{2}{\rho_0}, L_1^*=1-\frac{2}{\rho_1}\right)$ if and only if $\{\rho_0\geq 2, \rho_1\geq 2\}$.

If $\rho_0\leq\rho_1/(\rho_1-1)$ then given the asymmetric cost structure in Table 2.1, the neutral zone classifier in **Definition 1** and the Bayes classifier in (2.3) are equivalent and the optimal choice of the optimal choice of (L_0,L_1) is $(L_0^*=0, L_1^*=0)$ if and only if $\rho_0=\rho_1$.

Proof: see Appendix A

Note that when we input L_0^* and L_1^* into the neutral zone classifier in **Definition 1** then our classifier can be expressed as

$$\hat{C}(y) \in \begin{cases} 0 & \text{if} \quad p_1(y) < \dfrac{1}{\rho_0} \\ 1 & \text{if} \quad p_1(y) > 1 - \dfrac{1}{\rho_1} \\ N & \text{if} \quad \dfrac{1}{\rho_0} \le p_1(y) \le 1 - \dfrac{1}{\rho_1} \end{cases} \tag{2.7}$$

2.2.4. Equivalence Examples

In this section we look at several examples showing when equivalence between the Bayes neutral zone classifier in (2.4) and the neutral zone classifier in (2.6) is achieved.

Example 1: Normal – Equal Costs

Our data is from one of two normal distributions with equal probability. The normal distributions are $f_0(y) \sim N(\mu = 1, \sigma = .5)$ and $f_1(y) \sim N(\mu = 2, \sigma = .6)$. Also our cost ratios are $\rho_0 = \rho_1 = 4$. Performing a grid search over $\{L_0, L_1 \in [0,1]\}$ to find the values of L_0 and L_1 that minimizes the expected cost in (2.2) we get $L_0 = L_1 = \frac{1}{2}$. This result is consistent with what we would expect from **Theorem 1** which tells us that $L_0^* = 1 - \frac{2}{4} = \frac{1}{2}$ and $L_1^* = 1 - \frac{2}{4} = \frac{1}{2}$. The neutral zone classifier can then be expressed in the form of (2.6) as

$$\hat{C}_{NZ}\left(y;L_0,L_1\right)\in\begin{cases}0 & \text{if} \quad p_0\left(y\right)-p_1\left(y\right)>\dfrac{1}{2}\\ 1 & \text{if} \quad p_1\left(y\right)-p_0(y)>\dfrac{1}{2}\\ N & \text{if} \quad p_0\left(y\right)-p_1\left(y\right)<\dfrac{1}{2}\text{and } p_1\left(y\right)-p_0\left(y\right)<\dfrac{1}{2}\end{cases}.$$

Furthermore, we simulated 50,000 observations our mixture of normal distributions and classified each value using both the neutral zone classifier with $L_0=L_1=\frac{1}{2}$ and the Bayes classifier in (2.4). The classification of each observation was identical for both classifiers. This illustrates our boundaries for neutral zone classification derived in **Theorem 1** are indeed optimal and eliminates the need to perform a computationally intensive grid search to the values of L_0 and L_1 hat minimizes the expected cost in (2.2). Additionally, for the one L neutral zone classifier in (2.1) our grid search for the choice of L hat minimizes the expected cost in (2.2) is $L=0.5$ which tells that in this equal cost situation there is no difference between the neutral zone classifier in (2.6) and the neutral zone classifier in (2.1). This equivalence under the equal cost setting is examined in Section 2.2.6.

Example 2: Normal – Unequal Costs

Our data is again from one of two normal distributions with equal probability. The normal distributions are $f_0\left(y\right)\sim N\left(\mu=1,\sigma=.5\right)$ and $f_1\left(y\right)\sim N\left(\mu=2,\sigma=.6\right)$. Our cost ratios are $\rho_0=5$ and $\rho_1=4$. Performing a grid search to find the values of L_0 and L_1 hat minimizes the expected cost in (2.2) we get $L_0=0.6$ and $L_1=0.5$. This result

is consistent with what we would expect from **Theorem 1** which tells us that

$L_0^* = 1 - \frac{2}{5} = \frac{3}{5}$ and $L_1^* = 1 - \frac{2}{4} = \frac{1}{2}$ This gives us a neutral zone classifier in the form of (2.6)that can be represented as

$$\hat{C}_{NZ}(y; L_0, L_1) \in \begin{cases} 0 & \text{if} \quad p_0(y) - p_1(y) > \frac{3}{5} \\ 1 & \text{if} \quad p_1(y) - p_0(y) > \frac{1}{2} \\ N & \text{if} \quad p_0(y) - p_1(y) < \frac{3}{5} \text{and } p_1(y) - p_0(y) < \frac{1}{2} \end{cases}.$$

Furthermore, we simulated 50,000 observations our mixture of normal distributions and classified each value using both the neutral zone classifier with $L_0 = 0.6$ and $L_1 = 0.5$ and the Bayes classifier in (2.4). The classification of each observation was identical for both classifiers. This illustrates our boundaries for neutral zone classification derived in **Theorem 1** are indeed optimal and eliminates the need to perform a computationally intensive grid search to the values of L_0 and L_1 hat minimizes the expected cost in (2.2). Additionally, for the one L neutral zone classifier in (2.1) our grid search for the choice of L hat minimizes the expected cost in (2.2) is $L = 0.558$. To compare the average cost of the neutral zone classifier in (2.6) and the neutral zone classifier in (2.1) we obtained a 95% confidence interval for the difference in the average cost of the two classifiers. The confidence interval is (-0.0538964, -0.0229836) for the average cost of the zone classifier in (2.6) minus the average cost of the neutral zone classifier in (2.1). This shows us that when we allow for two choices of L we do in fact obtain a classifier with a lower cost.

Example 3: Normal – Unequal Costs/Nonequivalence

Our data is again from one of two normal distributions with equal probability. The normal distributions are $f_0(y) \sim N(\mu = 1, \sigma = .5)$ and $f_1(y) \sim N(\mu = 2, \sigma = .6)$. Our cost ratios are $\rho_0 = 3$ and $\rho_1 = 1.8$. We can see from (2.4) that our boundaries for Bayes classification are as follows

$$\hat{C}_B(y) \in \begin{cases} 0 & \text{if} \quad p_1(y) > \dfrac{1}{3} \\ 1 & \text{if} \quad p_1(y) > \dfrac{4}{9} \\ N & \text{if} \quad \dfrac{1}{3} < p_1(y) < \dfrac{4}{9} \end{cases} .$$

However, when $y = 1.45$ then $p_0(y) = 0.549205$ and $p_1(y) = 0.450795$ which will lead us to a classification value of $\hat{C}_B(1.45) = 1$, which by definition is not possible in the neutral zone classifier. Therefore the two classifiers are not equivalent under this cost structure.

2.2.5. Other Cost Scenarios

Theorem 1 gives us conditions when the two-class neutral zone classifier and the two-class Bayes classifier are equivalent, which allows for direct calculation of L_0^* and L_1^*. In Figure 2.1 we see when the Bayes and neutral zone classifier are equivalent for all cost scenarios. As **Theorem 1** states when $\rho_0 > \rho_1 / (\rho_1 - 1)$ then equivalence between the Bayes and neutral zone classifier is achieved in the shaded area in Figure 2.1 or

$\{\rho_0 \geq 2, \rho_1 \geq 2\}$ Also, when $\rho_0 \leq \rho_1 / (\rho_1 - 1)$ then equivalence between the Bayes and neutral zone classifier can only be achieved along the line $\rho_0 = \rho_1$ as shown in Figure 2.1.

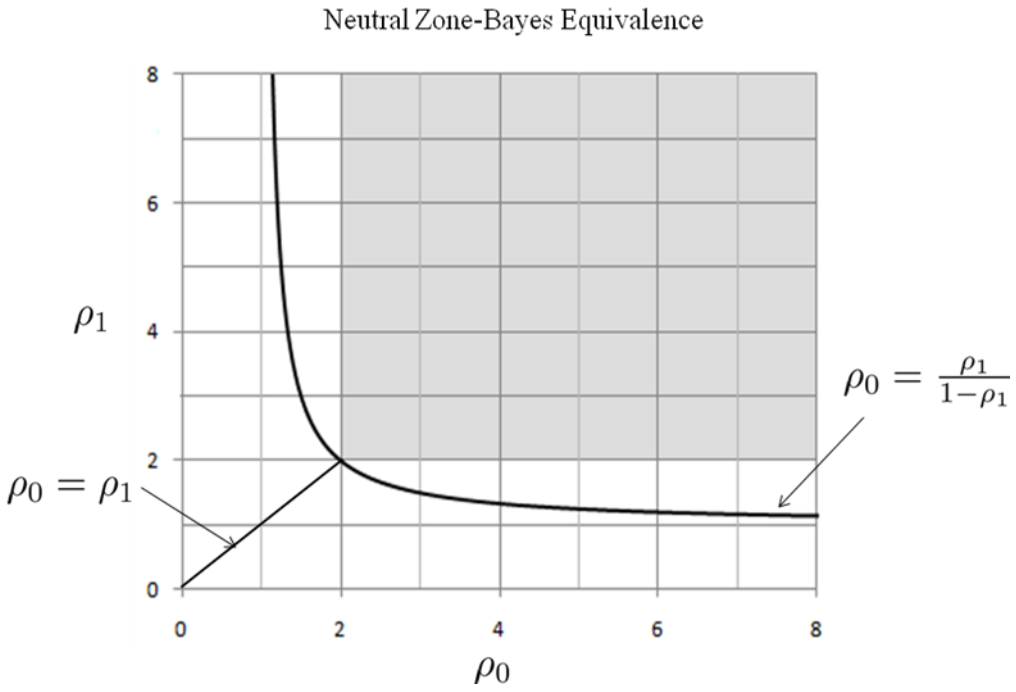

Figure 2.1. Shaded area shows neutral zone equivalence.

2.2.6. Symmetry Considerations

From (2.7), we see that for the neutral zone region to be symmetric around 0.5 then $0.5\left(\frac{1}{\rho_0} + 1 - \frac{1}{\rho_1}\right) = 0.5$. Equivalently, $\frac{1}{\rho_0} + 1 - \frac{1}{\rho_1} = 1$ or $\frac{1}{\rho_0} = \frac{1}{\rho_1}$. That is $\rho_0 = \rho_1$.

The classifier in this case becomes

$$\hat{C}(y) \in \begin{cases} 0 & \text{if} \quad p_1(y) < \dfrac{1}{\rho} \\ 1 & \text{if} \quad p_1(y) > 1 - \dfrac{1}{\rho} \\ N & \text{if} \quad \dfrac{1}{\rho} \le p_1(y) \le 1 - \dfrac{1}{\rho} \end{cases} .$$

and the necessary and sufficient conditions for which an *N* classification is possible becomes $\rho > 2$ This is the classifier mentioned in Yu et al. (2009).

2.3. Three-Class Neutral Zone Classifiers

2.3.1. Previous Work

For a three-class neutral zone classifier we are seeking to classify an observation as either 0, 1, 2 or *N*. Using similar notation as in Section 2.1, let π_0, π_1 and π_2 be the prior probability that an observation belongs to class 0 ($C = 0$), class 1 ($C = 1$) and class 2 ($C = 2$), respectively, where $\pi_0 + \pi_1 + \pi_2 = 1$. Then the posterior probability of the event $C = 1$ is given by $p_1(y) = f_1(y)\pi_1 / \left(f_0(y)\pi_0 + f_1(y)\pi_1 + f_2(y)\pi_2 \right)$ where f_i is the conditional density of class i. We assume we have sufficient training data to estimate the densities via a suitable NDE. Yu et al. (2009) defined a three-class neutral zone classifier as

$$\hat{C}_{NZ}(y; L) = \begin{cases} 0 & \text{if } p_0(y) - p_1(y) \ge L \text{ and } p_0(y) - p_2(y) \ge L \\ 1 & \text{if } p_1(y) - p_0(y) \ge L \text{ and } p_1(y) - p_2(y) \ge L \\ 2 & \text{if } p_2(y) - p_0(y) \ge L \text{ and } p_2(y) - p_1(y) \ge L \\ N & \text{otherwise ,} \end{cases} \tag{2.8}$$

where $L \in [0,1]$ and is a threshold that establishes the classification boundaries. The optimal value of L for the classifier in (2.8) is determined by minimizing the expected cost with respect to the general form in Table 2.2, which is given by

$$\begin{aligned} EC_{NZ}(L) \quad \propto \quad & \pi_0 \left[\rho_{10} P\left(\hat{C}=1|C=0\right) + \rho_{20} P\left(\hat{C}=2|C=0\right) + P\left(\hat{C}=N|C=0\right) \right] \\ + \quad & \pi_1 \left[\rho_{01} P\left(\hat{C}=0|C=1\right) + \rho_{21} P\left(\hat{C}=2|C=1\right) + P\left(\hat{C}=N|C=1\right) \right] \\ + \quad & \pi_2 \left[\rho_{02} P\left(\hat{C}=0|C=2\right) + \rho_{12} P\left(\hat{C}=1|C=2\right) + P\left(\hat{C}=N|C=2\right) \right] \end{aligned} \tag{2.9}$$

where $\rho_{ij} = C_{ij} / C_N$. As in the two-class case, a numerical search method must be used to determine the optimal L, making the implementation of the classifier demanding from a computational point of view.

True Class Label	Predicted Class Label			
	0	1	2	N
0	0	C_{10} (4)	C_{20} (4.5)	C_N (1)
1	C_{01} (4)	0	C_{21} (4)	C_N (1)
2	C_{02} (4.5)	C_{12} (4)	0	C_N (1)

Table 2.2. Asymmetric cost structure in three-class setting with exact costs used for polony example in parentheses.

2.3.2. Three-Class Bayes Neutral Zone Classifier

The three-class Bayes neutral zone classifier is developed as follows. First, EC_B is modified to be the expected cost of an arbitrary classifier that assigns labels 0, 1, 2 and

N, when *Y* falls in the regions R_0, R_1, R_2 and R_N, respectively. Assuming the cost structure shown in the general form in Table 2.2, we then have

$$\begin{aligned} EC_B \propto\ & 1+\int_{R_0}\left[\rho_{10}\pi_1 f_1(y)+\rho_{20}\pi_2 f_2(y)-\pi_0 f_0(y)-\pi_1 f_1(y)-\pi_2 f_2(y)\right]dy \\ & +\int_{R_1}\left[\rho_{01}\pi_0 f_0(y)+\rho_{21}\pi_2 f_2(y)-\pi_0 f_0(y)-\pi_1 f_1(y)-\pi_2 f_2(y)\right]dy \\ & +\int_{R_2}\left[\rho_{02}\pi_0 f_0(y)+\rho_{12}\pi_1 f_1(y)-\pi_0 f_0(y)-\pi_1 f_1(y)-\pi_2 f_2(y)\right]dy \end{aligned} \tag{2.10}$$

Let $I_0(y)$, $I_1(y)$ and $I_2(y)$ be the terms in the integral over R_0, R_1 and R_2, respectively. Again, using boundary-value conventions consistent with Johnson and Wichern (2007), it follows the Bayes neutral zone classifier in the three-class case has the form

$$\hat{C}_B(y)=\begin{cases} 0 & \text{if} \quad I_0(y)<\min[\,0\,,I_1(y)\,,I_2(y)\,] \\ 1 & \text{if} \quad I_1(y)<\min[\,0\,,I_0(y)\,,I_2(y)\,] \\ 2 & \text{if} \quad I_2(y)<\min[\,0\,,I_0(y)\,,I_1(y)\,] \\ N & \text{if} \quad \min[\,I_0(y)\,,I_1(y)\,,I_2(y)\,]>0 \end{cases} \tag{2.11}$$

If all the cost ratios satisfy $\rho_{ij}\geq 2$, then from **Theorem 2** it is possible to further simplify (2.11) by defining it individually on the three interior $(p_0(y)\,,\,p_1(y))$ regions shown in Figure 2.2.

Theorem 2

Suppose $\rho_{lr} \geq 2 \forall \{l, r = 1,2,3 \text{ and } l \neq r\}$. Let

$I_j(y) = \pi_i C_{ji} f(y)_i + \pi_k C_{jk} f(y)_k - \pi_i C_N f_i(y) - \pi_j C_N f_j(y) - \pi_k C_N f_k(y)$ and

$I_k(y) = \pi_i C_{ki} f(y)_i + \pi_j C_{kj} f_j(y) - \pi_i C_N f_i(y) - \pi_j C_N f_j(y) - \pi_k C_N f_k(y)$. Then, if

$p_i(y) > p_j(y) > p_k(y)$, for $\{i, j, k = 1,2,3 \text{ and } i \neq j \neq k\}$, we necessarily have $I_j(y) > 0$ and $I_k(y) > 0$. Consequently, the Bayes neutral zone classifier on the branch $p_i(y) > p_j(y) > p_k(y)$ reduces to

$$\hat{C}_B(y) \in \begin{cases} i & \text{if} \quad I_i(y) < 0 \\ N & \text{if} \quad I_i(y) > 0 \end{cases}$$

Proof: See Appendix B.

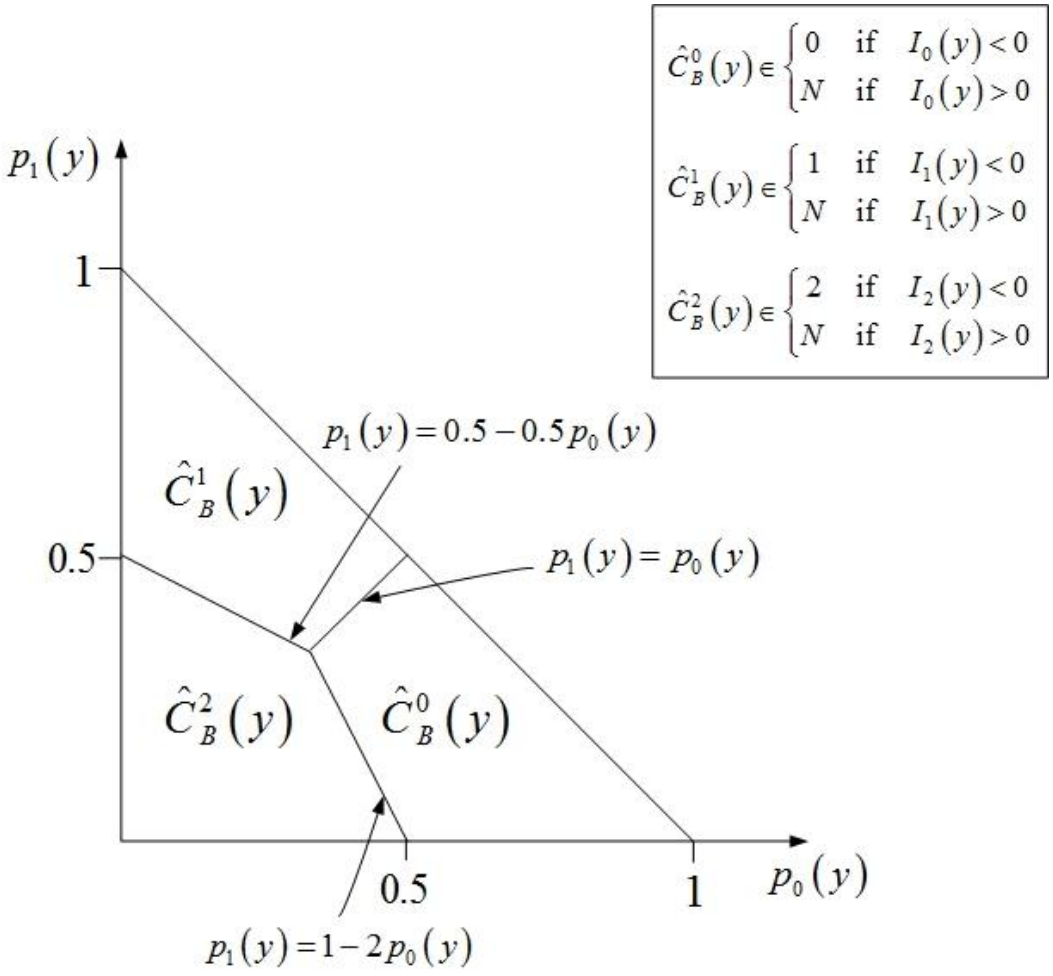

Figure 2.2. Classification regions for three-class Bayes neutral zone classifier under Theorem 2.

Evaluation of the classifier simply involves determining which regions the observed point $(p_0(y), p_1(y))$ lies, and then evaluating the predicted label according to the respective formulas $\hat{C}_B^0(y)$, $\hat{C}_B^1(y)$, and $\hat{C}_B^2(y)$ that are defined in Figure 2.2 and overlaid on each of the regions. Note that as in the two-class case, the Bayes neutral zone classifier is much easier to compute than the neutral zone classifier given by (2.8), and of course has a lower expected cost as well.

Examining (2.11), we can see that the Bayes neutral zone classifier will not have a neutral zone if $\min[\, I_0(y)\,, I_1(y)\,, I_2(y)\,] < 0$, for all *y*, and under this condition it will classify as $k = 0,1,2$ based on whichever of $I_0(y)\,, I_1(y)\,,$ or $I_2(y)$ is the smallest, respectively. After some algebraic manipulation, this can be restated as classifying $k = 0,1,2$ based on whichever of $\sum_{\substack{i=0 \\ i \neq k}}^{2} \pi_i f_i(y) C_{ki}$ is the smallest, and this characterization is equivalent to the standard three-class Bayes classifier with asymmetric cost structure which is detailed in **Theorem 3**.

Theorem 3

If $\min[\, I_0(y)\,, I_1(y)\,, I_2(y)\,] < 0$, for all *y*, then (2.11) can be restated as classifying $k = 0,1,2$ based on whichever of $\sum_{\substack{i=0 \\ i \neq k}}^{2} \pi_i f_i(y) C_{ki}$ is the smallest.

Proof: See Appendix C.

2.3.3. Equivalence of Bayes Neutral Zone Classifier

In this section we outline conditions when the neutral zone classifier in **Definition 2** and the Bayes classifier in (2.11) are equivalent. This leads to a direct method of calculating L_m in **Definition 2**. First let us state the neutral zone as follows:

Definition 2

For *Case m* where $p_i(y) > p_j(y) > p_k(y)$ for $\{i, j, k = 1,2,3 \text{and } i \neq j \neq k\}$ let $L_m > 0$ be a threshold determined by the user, then

$$\hat{C}(y; L_m) = \begin{cases} i & \text{if} \quad p_i(y) - p_j(y) > L_m \\ N & \text{if} \quad p_i(y) - p_j(y) < L_m \end{cases}$$

For our purposes we will let *Case 1* be the set $\{y: \ p_0(y) > p_1(y) > p_2(y)\}$ *Case 2* be the set $\{y: \ p_0(y) > p_2(y) > p_1(y)\}$ *Case 3* be the set $\{y: \ p_1(y) > p_0(y) > p_2(y)\}$ *Case 4* be the set $\{y: \ p_1(y) > p_2(y) > p_0(y)\}$ *Case 5* be the set $\{y: \ p_2(y) > p_0(y) > p_1(y)\}$ nd *Case 6* be the set $\{y: \ p_2(y) > p_1(y) > p_0(y)\}$ Note that not all cases are possible.

Notice that **Definition 2** proposes a three-class neutral zone classifier where the threshold, L_m, varies depending on the order of your posterior probabilities. This is a slightly different form than the neutral zone classifier proposed by Yu et al. (2009) in (2.8) and is necessary for equivalence of the Bayes neutral zone classifier. For instance, when $p_0(y) > p_1(y) > p_2(y)$ the choice of L_m is different than the case when $p_2(y) > p_1(y) > p_0(y)$. In this section define how to choose the optimal value of L_m. Also, **Definition 2** only allows for a classification of either the class with the highest posterior probability, $p_i(y)$, or *N*. This is consistent with previous definitions of neutral zone classification. Figure 2.3 shows the regions for the six cases in **Definition 2**.

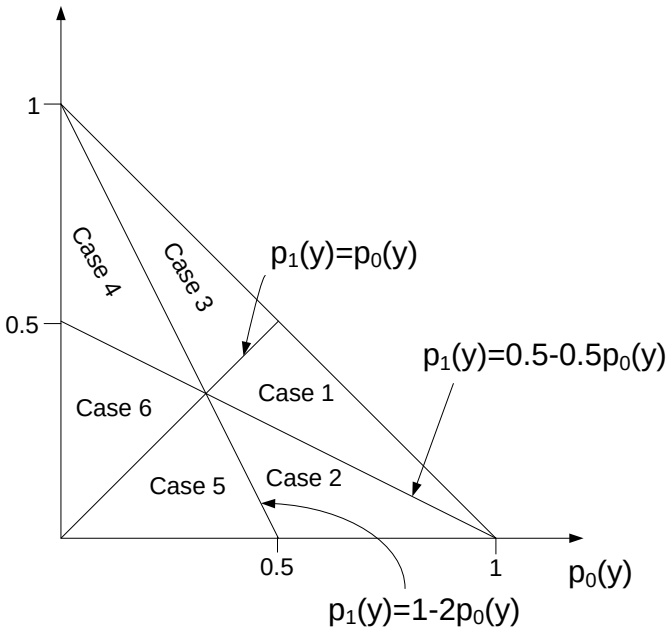

Figure 2.3. Six region of neutral zone in Definition 2.

Lemma 3

If $p_i(y) > p_j(y) > p_k(y) \text{for} \{i, j, k = 1,2,3 \text{and } i \neq j \neq k\}$ and if

$\rho_{lr} \geq 2 \forall \{l, r = 1,2,3 \, and \, l \neq r\}$ then $I_j > 0$ and $I_k > 0$ where

$I_j = \pi_i C_{ji} f_i + \pi_k C_{jk} f_k - \pi_i C_N f_i - \pi_j C_N f_j - \pi_k C_N f_k$ and

$I_k = \pi_i C_{ki} f_i + \pi_j C_{kj} f_j - \pi_i C_N f_i - \pi_j C_N f_j - \pi_k C_N f_k$

Proof

Because $p_i(y) > p_j(y) > p_k(y)$ we must have $p_i(y) + p_k(y) > \frac{1}{2}$ otherwise $p_j(y) > \frac{1}{2}$

and would be larger than $p_i(y)$ Now consider

$$
\begin{aligned}
I_j &= \pi_i C_{ji} f_i + \pi_k C_{jk} f_k - \pi_i C_N f_i - \pi_j C_N f_j - \pi_k C_N f_k \\
&= C_N \left\{ \pi_i \rho_{ji} f_i + \pi_k \rho_{jk} f_k - \left(\pi_i f_i + \pi_j f_j + \pi_k f_k \right) \right\} \\
&\geq C_N \left\{ 2\left(\pi_i f_i + \pi_k f_k \right) - \left(\pi_i f_i + \pi_j f_j + \pi_k f_k \right) \right\} \\
&= \frac{C_N}{\pi_i f_i + \pi_j f_j + \pi_k f_k} \left\{ 2\left(p_i(y) + p_k(y) \right) - 1 \right\} \\
&> 0
\end{aligned}
$$

Similarly, because $p_i(y) > p_j(y) > p_k(y)$ we must have $p_i(y) + p_j(y) > \frac{1}{2}$ otherwise $p_k(y) > \frac{1}{2}$ and would be larger than $p_i(y)$. Now consider

$$
\begin{aligned}
I_k &= \pi_i C_{ki} f_i + \pi_j C_{kj} f_j - \pi_i C_N f_i - \pi_j C_N f_j - \pi_k C_N f_k \\
&= C_N \left\{ \pi_i \rho_{ki} f_i + \pi_j \rho_{kj} f_j - \left(\pi_i f_i + \pi_j f_j + \pi_k f_k \right) \right\} \\
&\geq C_N \left\{ 2\left(\pi_i f_i + \pi_j f_j \right) - \left(\pi_i f_i + \pi_j f_j + \pi_k f_k \right) \right\} \\
&= \frac{C_N}{\pi_i f_i + \pi_j f_j + \pi_k f_k} \left\{ 2\left(p_i(y) + p_j(y) \right) - 1 \right\} \\
&> 0
\end{aligned}
$$

Theorem 4

For *Case m* where $y \in \left\{ y: \; y \in p_i(y) > p_j(y) > p_k(y) \right\}$ let $g(y) = p_i(y) - p_j(y)$, $h(y) = I_i$, $\varepsilon > 0$ and given the asymmetric cost structure in Table 2.2. If y^* is the unique root of $h(y)$ and one of the following conditions hold:

1. $h\left(y^* + \varepsilon \right) < 0$ and $g(y) < g\left(y^* \right) \forall\, y \in \left\{ y : y < y^* \right\}$ and

   $g(y) > g\left(y^* \right) \forall\, y \in \left\{ y : y > y^* \right\}$

2. $h\left(y^{*}-\varepsilon\right)<0$ and $g(y)>g\left(y^{*}\right)\forall\, y\in\left\{y: y<y^{*}\right\}$ and

$g(y)<g\left(y^{*}\right)\forall\, y\in\left\{y: y>y^{*}\right\}$

then the three-class neutral zone classifier proposed in **Definition 2** and the Bayes classifier in (2.11) are equivalent and the optimal choice (i.e. the choice that minimizes the expected cost) of L_m is $L_m^{*}=g\left(y^{*}\right)$ if and only if $I_j>0$ and $I_k>0$

Proof: See Appendix C.

Note: If there is no unique root of $h(y)$ then $L_m^{*}=0$ if $h(y)<0$ and $L_m^{*}=1$ if $h(y)>0$

Note that **Lemma 3** gives us a sufficient condition on ρ_{lr} where $\left\{l,r=1,2,3 \text{ and } l\neq r\right\}$ for **Theorem 4** to hold, that is $\rho_{lr}\geq 2\,\forall\left\{l,r=1,2,3 \text{ and } l\neq r\right\}$. We can, however, find a cost structure illustrated by the Example 3 in Section 2.3.4 where $\rho_{lr}<2$ for some $\left\{l,r=1,2,3 \text{ and } l\neq r\right\}$ and **Theorem 4** will still hold.

The neutral zone classifier in **Definition 2** under cases 1-6 is represented graphically in Figure 2.4 where the coordinates of the points are as follows:

$$a_1 = \left(\frac{1+L_1^*}{2}, \frac{1-L_1^*}{2} \right) \quad b_1 = \left(\frac{1+2L_1^*}{3}, \frac{1-L_1^*}{3} \right)$$

$$a_2 = \left(\frac{1+L_2^*}{2}, 0 \right) \quad b_2 = \left(\frac{1+2L_2^*}{3}, \frac{1-L_2^*}{3} \right)$$

$$a_3 = \left(\frac{1-L_3^*}{2}, \frac{1+L_3^*}{2} \right) \quad b_3 = \left(\frac{1-L_3^*}{3}, \frac{1+2L_3^*}{3} \right)$$

$$a_4 = \left(0, \frac{1+L_4^*}{2} \right) \quad b_4 = \left(\frac{1-L_4^*}{3}, \frac{1+2L_4^*}{3} \right)$$

$$a_5 = \left(\frac{1-L_5^*}{2}, 0 \right) \quad b_5 = \left(\frac{1-L_5^*}{3}, \frac{1-L_5^*}{3} \right)$$

$$a_6 = \left(0, \frac{1-L_6^*}{2} \right) \quad b_6 = \left(\frac{1-L_6^*}{3}, \frac{1-L_6^*}{3} \right)$$

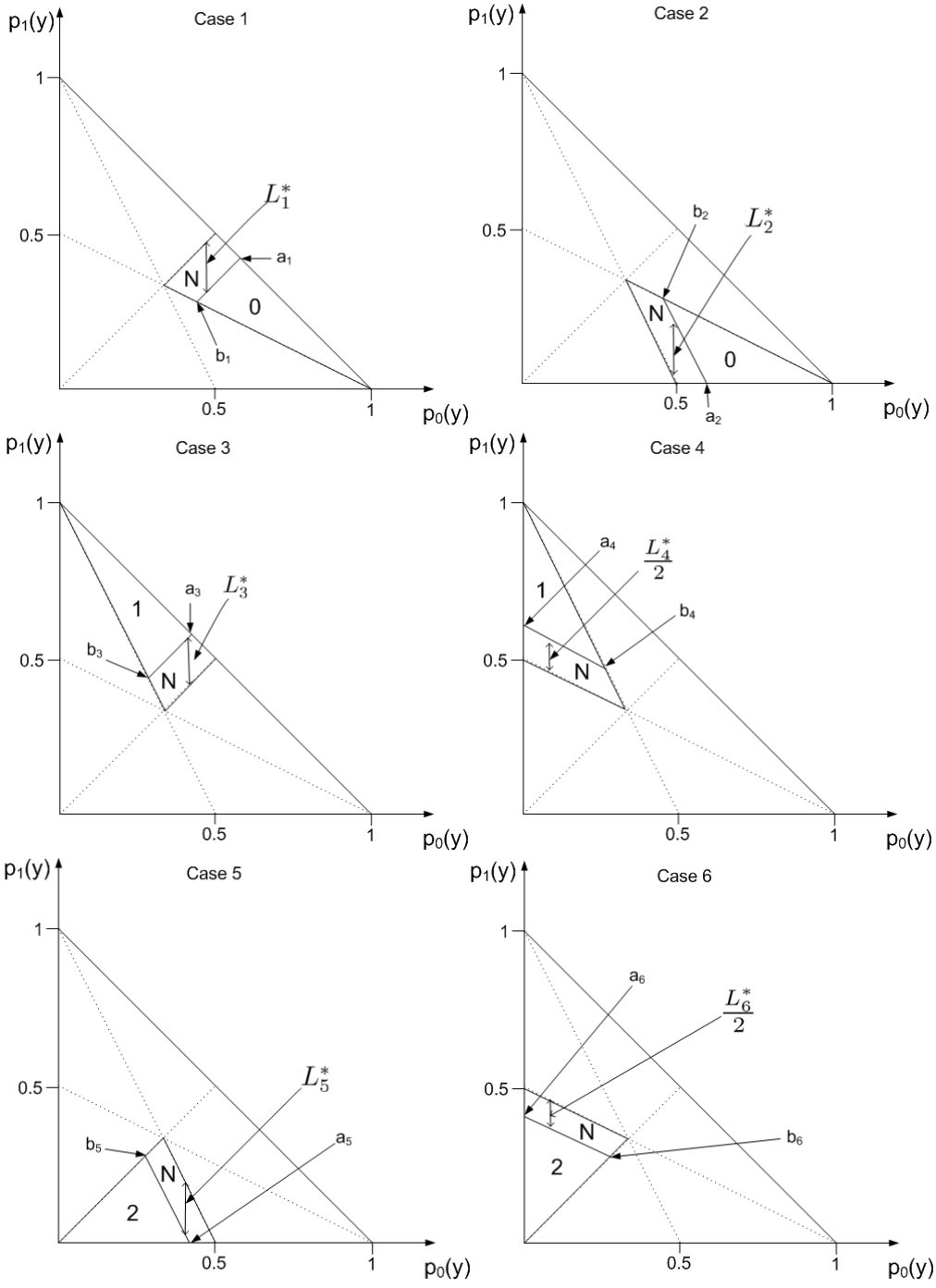

Figure 2.4. Neutral zone classifier for six cases.

Combining the regions in cases 1-6 into one plot we have the following regions of classification shown in Figure 2.5.

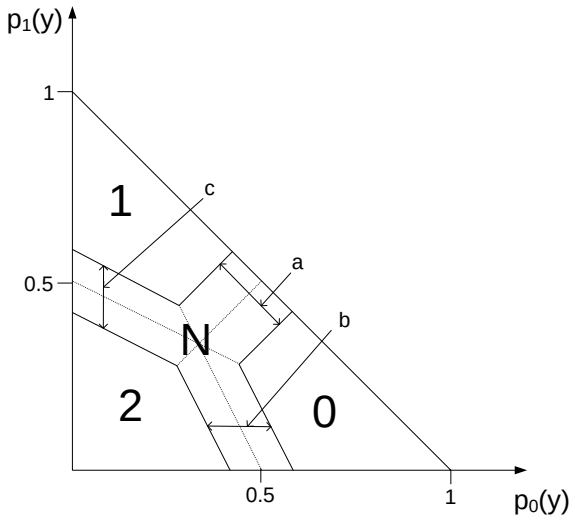

Figure 2.5. Combined classification plot for neutral zone.

where in Figure 2.5 $a = \frac{1}{\sqrt{2}}\left(L_2^* + L_5^*\right)$, $b = \frac{1}{2}\left(L_2^* + L_5^*\right)$ and $c = \frac{1}{2}\left(L_4^* + L_6^*\right)$. Figure 2.5 and the length of a, b and c are consistent with the derivation in Yu et al. (2009).

Theorem 4 shows a direct way of calculating the optimal choice of L_m for the neutral zone classifier proposed in **Definition 2**. This greatly reduces the computational burden of finding an optimal six L neutral zone classifier. Additionally **Theorem 4** shows that in order for the neutral zone classifier to obtain optimal boundaries then six different thresholds are needed, one for each of the six cases.

2.3.4. Equivalence Examples

In this section we look at several examples showing when equivalence between the Bayes neutral zone classifier in (2.11) and the neutral zone classifier in **Definition 2** is achieved.

Example 1: Normal – Equal Variances, Equal Cost

Consider the following example where we have misclassification costs equal to $\rho_{lr} = 3 \forall \{l, r = 1,2,3 \, and \, l \neq r\}$. Our data is from one of three normal distributions with equal prior probabilities $f_0(y) \sim N(\mu = 1, \sigma = 1)$, $f_1(y) \sim N(\mu = 3, \sigma = 1)$ and $f_2(y) \sim N(\mu = 5, \sigma = 1)$. The results are summarized in Table 2.3.

For example, under case 1 we see that the possible y values are $y \in \{y : y < 2\}$. In addition we see in Figure 2.6 that $h(y)$ has a unique root, $y^* = 1.648909$ and $h(y^* - \varepsilon) < 0$, and also that $g(y) > g(y^*) \forall \, y \in \{y : y < y^*\}$ and $g(y) < g(y^*) \forall \, y \in \{y : y > y^*\}$. Therefore $L_1^* = g(y^*) = 0.3363313$ and our optimal classification region when $y \in \{y : \, p_0(y) > p_1(y) > p_2(y)\}$ is

$$\hat{C}(y; L) = \begin{cases} 0 & \text{if } y < 1.648909 \\ N & \text{if } y > 1.648909 \end{cases}$$

Case	Posterior Probability Order	y range	y^*	$g(y^*)$	N Region
1	$p_0(y) > p_1(y) > p_2(y)$	$y < 2$	1.648909	0.3363313	$y > y^*$
2	$p_0(y) > p_2(y) > p_1(y)$	$y \in \varnothing$	-	-	-
3	$p_1(y) > p_0(y) > p_2(y)$	$2 < y < 3$	2.388046	0.3598662	$y < y^*$
4	$p_1(y) > p_2(y) > p_0(y)$	$3 < y < 4$	3.611954	0.3598662	$y > y^*$
5	$p_2(y) > p_0(y) > p_1(y)$	$y \in \varnothing$	-	-	-
6	$p_2(y) > p_1(y) > p_0(y)$	$y > 4$	4.351091	0.3363313	$y < y^*$

Table 2.3. Example 1 simulation results.

Furthermore, we simulated 10,000 observations from our mixture of three normal distributions and classified each simulated value using both the neutral zone classifier in **Definition 2** with $L_1^* = 0.3363313$, $L_3^* = 0.3598662$, $L_4^* = 0.3598662$ and $L_6^* = 0.3363313$ and the Bayes classifier in (2.11). The classification of each observation was identical for both classifiers. This illustrates the boundaries for neutral zone classification derived in **Theorem 4** are indeed optimal since the neutral zone and Bayes classifications are identical. Additionally, we compared the accuracy and runtime for the neutral zone classifier in **Definition 2** with $L_1^* = 0.3363313$, $L_3^* = 0.3598662$, $L_4^* = 0.3598662$ and $L_6^* = 0.3363313$ and the neutral zone classifier in (2.8). The neutral zone classifier in (2.8), with $L = 0.346$ takes 817.86 seconds to run as opposed to 5.43 seconds for the neutral zone classifier in **Definition 2**. Furthermore, a 95% confidence interval for the

mean cost of the neutral zone classifier in **Definition 2** minus the mean cost of the neutral zone classifier in (2.8) is -0.005 meaning we would need a sample size of 290,705 in order for the mean cost of the neutral zone classifier in **Definition 2** to be significantly less than the mean cost for the neutral zone classifier in (2.8). Additionally, **Figure 2.7** is the spider plot for this dataset where the bold line through the region is the possible points that can be attained. Note that this bold line does not pass through the region for *Case 2* or *Case 5*.

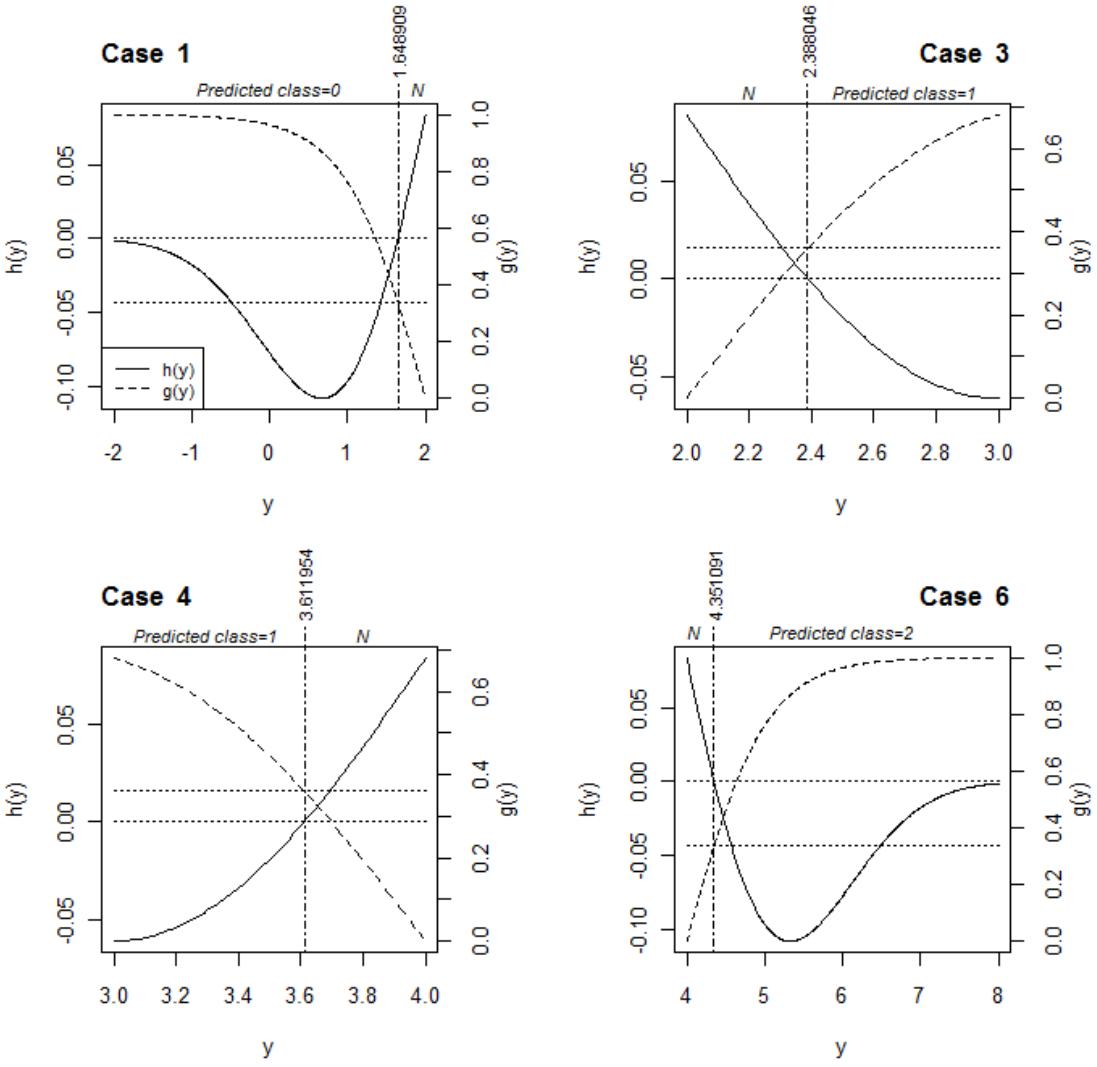

Figure 2.6. Example 1 simulation results.

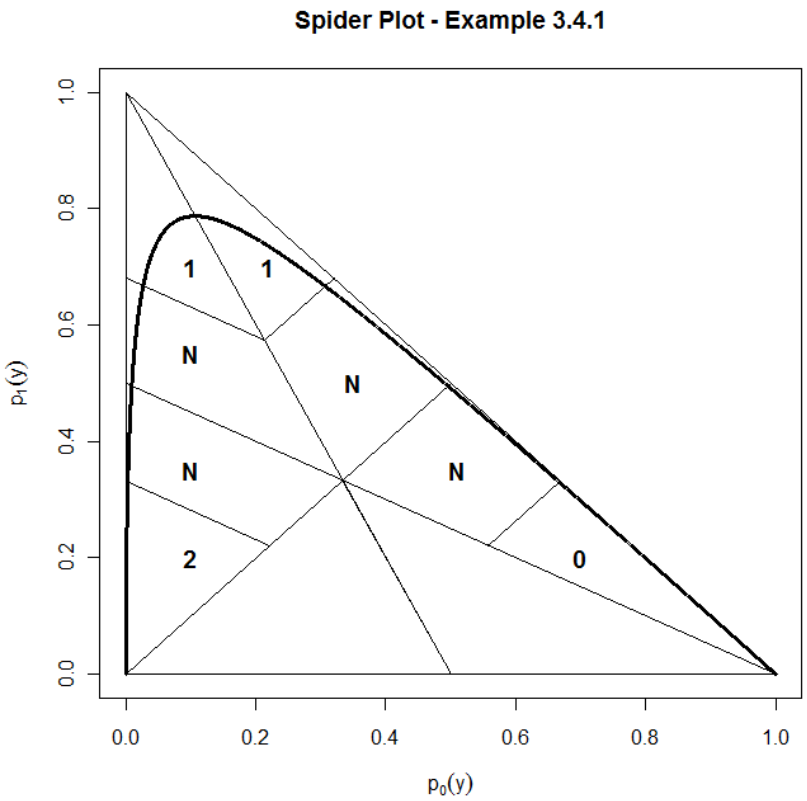

Figure 2.7. Example 1 spider plot with possible values plotted.

Example 2: Normal – Unequal Variances, Unequal Costs

Consider the following cost structure in Table 2.4

True Class Label	Predicted Class Label			
	0	1	2	N
0	0	2.2	2.5	1
1	2.3	0	2.4	1
2	3.1	2.8	0	1

Table 2.4. Example 2 cost structure.

therefore $\rho_{01} = 2.2$, $\rho_{02} = 2.5$, $\rho_{10} = 2.3$, $\rho_{12} = 2.4$, $\rho_{20} = 3.1$ and $\rho_{21} = 2.8$.

Our data is from one of three normal distributions with prior probabilities $\pi_0 = .3$, $\pi_1 = .3$ and $\pi_2 = .4$ and $f_0(y) \sim N(\mu = 1, \sigma = .5)$, $f_1(y) \sim N(\mu = 2.6, \sigma = .7)$ and $f_2(y) \sim N(\mu = 4.7, \sigma = 2.2)$. The results are summarized in Table 2.5.

For example, under case 1 we see that the possible y values are $y \in \{y : 1.02 < y < 1.67\}$. In addition we see in Figure 2.8 that $h(y)$ has a unique root, $y^* = 1.5581601$ and $h(y^* - \varepsilon) < 0$ and also that $g(y) > g(y^*) \forall\, y \in \{y : y < y^*\}$ and $g(y) < g(y^*) \forall\, y \in \{y : y > y^*\}$. Therefore $L_1^* = g(y^*) = 0.3406686$ and our optimal classification region when $y \in \{y :\ p_0(y) > p_1(y) > p_2(y)\}$ is

$$\hat{C}(y; L) = \begin{cases} 0 & \text{if } y < 1.5581601 \\ N & \text{if } y > 1.5581601 \end{cases}$$

Case	**Posterior Probability Order**	***y* range**	y^*	$g(y^*)$	***N* region**
1	$p_0(y) > p_1(y) > p_2(y)$	$1.02 < y < 1.67$	1.5581	0.3406	$y > y^*$
2	$p_0(y) > p_2(y) > p_1(y)$	$-0.39 < y < 1.02$	-0.2122	0.3562	$y < y^*$
3	$p_1(y) > p_0(y) > p_2(y)$	$1.67 < y < 1.98$	1.9362	0.3684	$y < y^*$
4	$p_1(y) > p_2(y) > p_0(y)$	$1.98 < y < 3.70$	3.2835	0.2857	$y > y^*$
5	$p_2(y) > p_0(y) > p_1(y)$	$-3.01 < y < -0.39$	-0.4787	0.2013	$y > y^*$
6	$p_2(y) > p_1(y) > p_0(y)$	$y < -3.01$ or $y > 3.70$	3.7252	0.1666	$y < y^*$

Table 2.5. Example 2 simulation results.

Furthermore, we simulated 1,000,000 observations from our mixture of three normal distributions and classified each simulated value using both the neutral zone classifier in **Definition 2** with $L_1^* = 0.340668$, $L_2^* = 0.3562209$, $L_3^* = 0.3684428$, $L_4^* = 0.2857388$, $L_5^* = 0.2013093$ and $L_6^* = 0.1666675$ and the Bayes classifier in (2.11). The classification of each observation was identical for both classifiers. This illustrates the boundaries for neutral zone classification derived in **Theorem 4** are indeed optimal since the neutral zone and Bayes classifications are identical. Additionally, we compared the accuracy and runtime for the neutral zone classifier in **Definition 2** with $L_1^* = 0.340668$, $L_2^* = 0.3562209$, $L_3^* = 0.3684428$, $L_4^* = 0.2857388$, $L_5^* = 0.2013093$ and $L_6^* = 0.1666675$ and the neutral zone classifier in (2.8). The neutral zone classifier in (2.8), with $L = 0.264$ takes 1191.82 seconds to run as opposed to 4.81 seconds for the neutral zone classifier in **Definition 2**. Furthermore, a 95% confidence interval for the mean cost of the neutral zone classifier in **Definition 2** minus the mean cost of the neutral zone classifier in (2.8) is (-0.004296347, 0.0009263471) indicating there is no significant different in the average cost of the two classifiers. In order for there to have been a significant difference we would have had to increase our sample size to 2,401,759 given our point estimate of -0.001685. Additionally, **Figure 2.9** is the spider plot for this dataset where the bold line through the region is the possible points that can be attained. Notice how in this example all cases are possible.

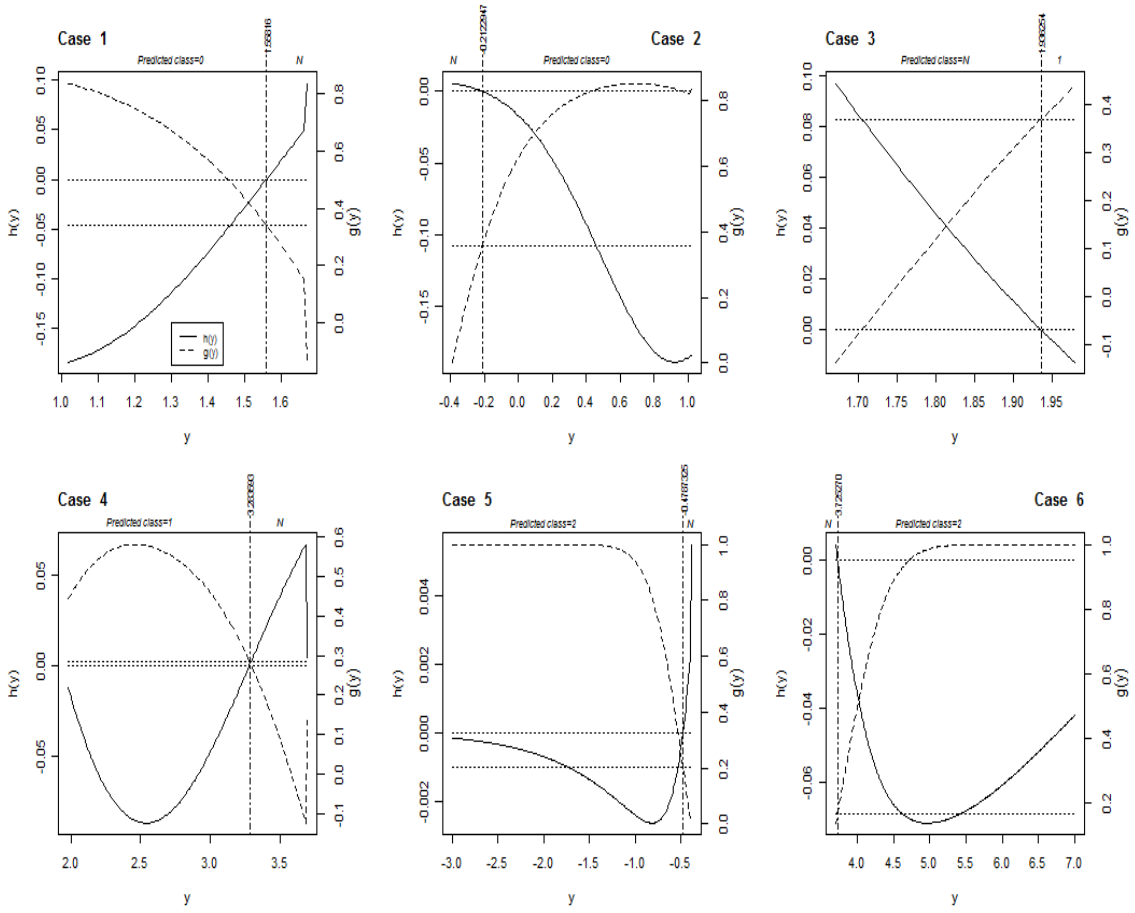

Figure 2.8. Example 2 simulation results.

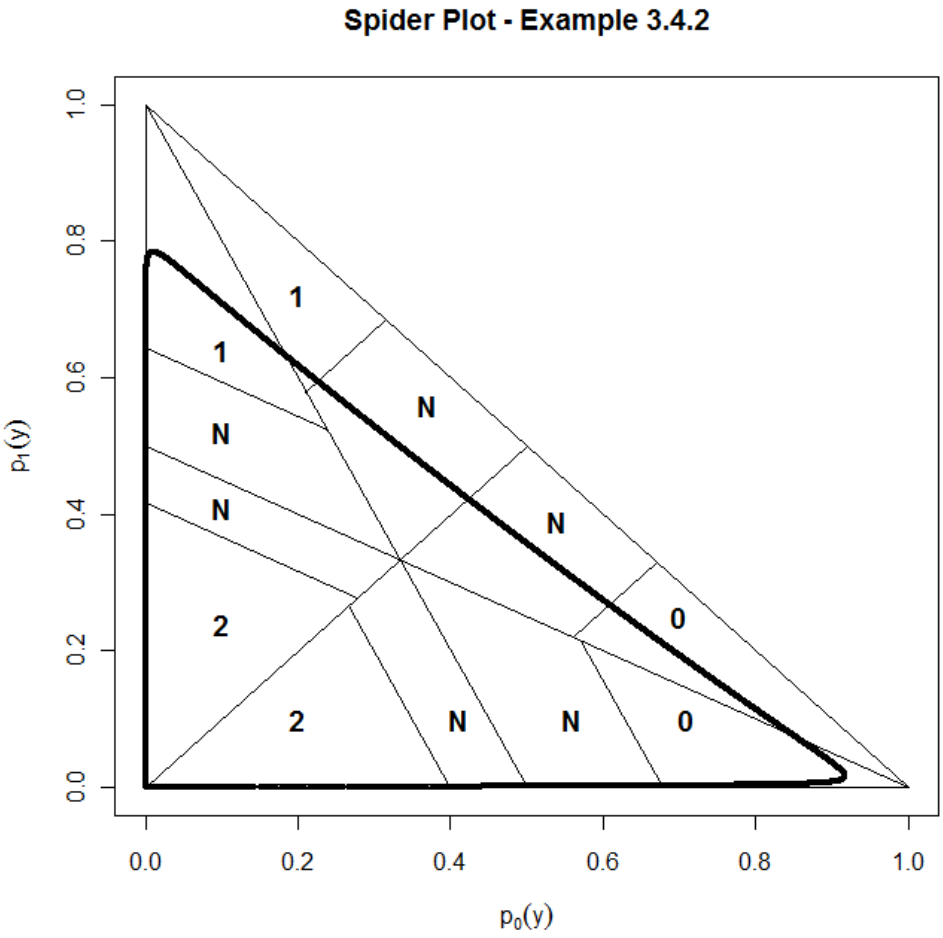

Figure 2.9. Example 2 spider plot.

Example 3: Normal – One $\rho_{ij} < 2$

Consider the following cost structure in **Table 2.6**

True Class Label	Predicted Class Label			
	0	1	2	N
0	0	1.9	2.5	1
1	2.3	0	2.4	1
2	3.1	7.0	0	1

Table 2.6. Example 3 cost structure.

therefore $\rho_{01} = 1.9$, $\rho_{02} = 2.5$, $\rho_{10} = 2.3$, $\rho_{12} = 2.4$, $\rho_{20} = 3.1$ and $\rho_{21} = 7.0$.

Our data is from one of three normal distributions with prior probabilities $\pi_0 = .3$, $\pi_1 = .3$, $\pi_2 = .4$ and $f_0(y) \sim N(\mu = 1, \sigma = 1)$, $f_1(y) \sim N(\mu = 3, \sigma = 1)$ and $f_2(y) \sim N(\mu = 5, \sigma = 1)$. The results are summarized in Table 2.7.

For example, under case 1 we see that the possible y values are $y \in \{y : 1.02 < y < 1.67\}$. In addition we see in Figure 2.10 that $h(y)$ has a unique root, $y^* = 1.5581601$ and $h(y^* - \varepsilon) < 0$ and also that $g(y) > g(y^*) \forall\, y \in \{y : y < y^*\}$ and $g(y) < g(y^*) \forall\, y \in \{y : y > y^*\}$. Therefore $L_1^* = g(y^*) = 0.3406686$ and our optimal classification region when $y \in \{y :\ p_0(y) > p_1(y) > p_2(y)\}$ is

$$\hat{C}(y; L) = \begin{cases} 0 & \text{if } y < 1.5581601 \\ N & \text{if } y > 1.5581601 \end{cases}$$

Case	Posterior Probability Order	y range	y^*	$g(y^*)$	N region
1	$p_0(y) > p_1(y) > p_2(y)$	$y < 2$	1.85780	0.140410	$y > y^*$
2	$p_0(y) > p_2(y) > p_1(y)$	$y \in \varnothing$	-	-	-
3	$p_1(y) > p_0(y) > p_2(y)$	$2 < y < 3$	2.00631	0.006253	$y < y^*$
4	$p_1(y) > p_2(y) > p_0(y)$	$3 < y < 4$	3.04512	0.669837	$y > y^*$
5	$p_2(y) > p_0(y) > p_1(y)$	$y \in \varnothing$	-	-	-
6	$p_2(y) > p_1(y) > p_0(y)$	$y > 4$	4.17510	0.172414	$y < y^*$

Table 2.7. Example 3 simulation results.

Furthermore, we simulated 10,000 observations from our mixture of three normal distributions and classified each simulated value using both the neutral zone classifier in **Definition 2** with $L_1^* = 0.1404102$, $L_3^* = 0.0062533$, $L_4^* = 0.6698374$ and $L_6^* = 0.1724142$ and the Bayes classifier in(2.11). The classification of each observation was identical for both classifiers. This illustrates the boundaries for neutral zone classification derived in **Theorem 4** are indeed optimal since the neutral zone and Bayes classifications are identical. This example illustrates that one of the cost ratios can be smaller than two and still have equivalence between the neutral zone classifier in **Definition 2** and the Bayes classifier in (2.11).

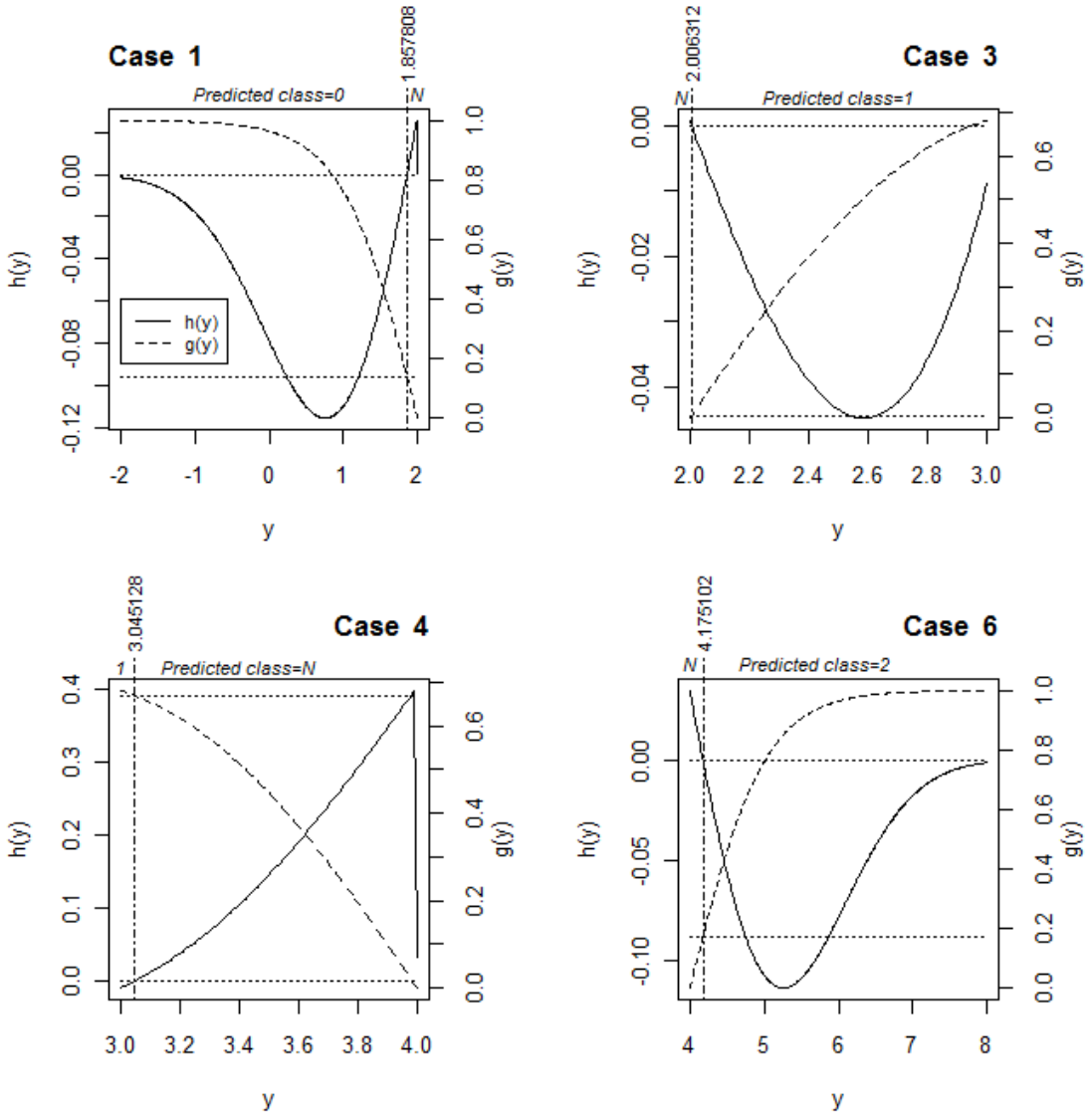

Figure 2.10. Example 3 simulation results.

2.3.5. Comments On One L Neutral Zone Classifier

In this section we look at some concerns with the version of the three-class neutral zone classifier as proposed by Yu et al. (2009). Note again that all inequalities hold with probability one. Given posterior probabilities,

$p_k(y) = P(C = k \mid Y = y) = f_k(y)\pi_k / \sum_{i=0}^{n} f_i(y)\pi_i$ where f_i is the conditional density of the i^{th} class and is estimated based on the training data from that class, let L be a threshold determined by the user, then the neutral zone classifier assigns a class value of N if $| p_{(n)}(y) - p_{(n-1)}(y) | < L$. Where $p_{(n)}(y)$ denotes the largest value among $\{p_i(y)\}_{i=0}^{n}$. For our purposes $n = 3$. Another way of expressing N if $| p_{(n)}(y) - p_{(n-1)}(y) | < L$ is as follows:

$$\hat{C}(y;L) = \begin{cases} 0 & \text{if } p_0(y) - p_1(y) > L \text{ and } p_0(y) - p_2(y) > L \\ 1 & \text{if } p_1(y) - p_0(y) > L \text{ and } p_1(y) - p_2(y) > L \\ 2 & \text{if } p_2(y) - p_0(y) > L \text{ and } p_2(y) - p_1(y) > L \\ N & \text{otherwise} \end{cases} \tag{2.12}$$

For a three-class example, Yu et al. (2009) showed that (2.12) can be expressed as:

$$\hat{C}(y;L) = \begin{cases} 0 & \text{if } p_1(y) > 1 - 2p_0(y) + L \text{ and } p_1(y) < p_0(y) - L \\ 1 & \text{if } p_1(y) > p_0(y) + L \text{ and } p_1(y) > -.5p_0(y) + .5 + .5L \\ 2 & \text{if } p_1(y) < 1 - 2p_0(y) - L \text{ and } p_1(y) < -.5p_0(y) + .5 - .5L \\ N & \text{otherwise} \end{cases} \tag{2.13}$$

Figure 2.11 gives us a visual representation of the form of a three-class neutral zone classifier when expressed in the form in (2.13). Then in the neutral zone classifier we find the optimal value of L by minimizing the expected cost function given by:

$$
\begin{aligned}
EC(L) \quad &\propto \quad \pi_0\left[\rho_2 P\left(\hat{C}=1|C=0\right)+\rho_1 P\left(\hat{C}=2|C=0\right)+P\left(\hat{C}=N|C=0\right)\right] \\
&+ \quad \pi_1\left[\rho_2 P\left(\hat{C}=0|C=1\right)+\rho_2 P\left(\hat{C}=2|C=1\right)+P\left(\hat{C}=N|C=1\right)\right] \\
&+ \quad \pi_2\left[\rho_1 P\left(\hat{C}=0|C=2\right)+\rho_2 P\left(\hat{C}=1|C=2\right)+P\left(\hat{C}=N|C=2\right)\right]
\end{aligned}
$$

where π_k is the prior probability of class k and ρ_i are the cost ratios associated with the misclassifications, $\rho_1 = C_1 / C_N$ and $\rho_2 = C_2 / C_N$. The costs of a misclassification are given in Table 2.8.

True Class Label	Predicted Class Label			
	0	1	2	N
0	0	C_2	C_1	C_N
1	C_2	0	C_2	C_N
2	C_1	C_2	0	C_N

Table 2.8. Three class symmetric cost structure.

Furthermore, $P\left(\hat{C}=j \mid \hat{C}=k\right)$ is defined as $P(\hat{C}=j \mid C=k)=\int_{A_k} f_j\left(y\right)dy$ where $A_k=\left\{y:\hat{C}\left(y;L\right)=k\right\}$. Then to find the optimal value of L, we search over L from 0 to 1 in order to find the value which minimizes the expected cost (Yu et al., 2009).

There are some concerns, however, when using the neutral zone classifier defined in (2.12). Note that if we use the definition for neutral zone classification as *N* if $|\, p_{(n)}(y)-p_{(n-1)}(y)\,| < L$ or the two-class neutral zone classifier, the classifier will take on the following form:

$$\hat{C}(y;L)=\begin{cases}0 & \text{if } p_0(y)-p_1(y)>L \\ 1 & \text{if } p_1(y)-p_0(y)>L \\ N & \text{if } p_0(y)-p_1(y)<L \text{ and } p_1(y)-p_0(y)<L\end{cases} \tag{2.14}$$

Which substituting $p_1(y)=1-p_0(y)$ into (2.14) yields the following:

$$\hat{C}(y;L)=\begin{cases}0 & \text{if } p_1(y)<\frac{1}{2}-\frac{L}{2} \\ 1 & \text{if } p_1(y)>\frac{1}{2}+\frac{L}{2} \\ N & \frac{1}{2}-\frac{L}{2}<p_1(y)<\frac{1}{2}+\frac{L}{2}\end{cases}.$$

Since the one L neutral zone classifier in (2.14) does not achieve optimal boundaries for the *N* region, except in the case of symmetric cost, we suspect that it is also insufficient in the three-class case.

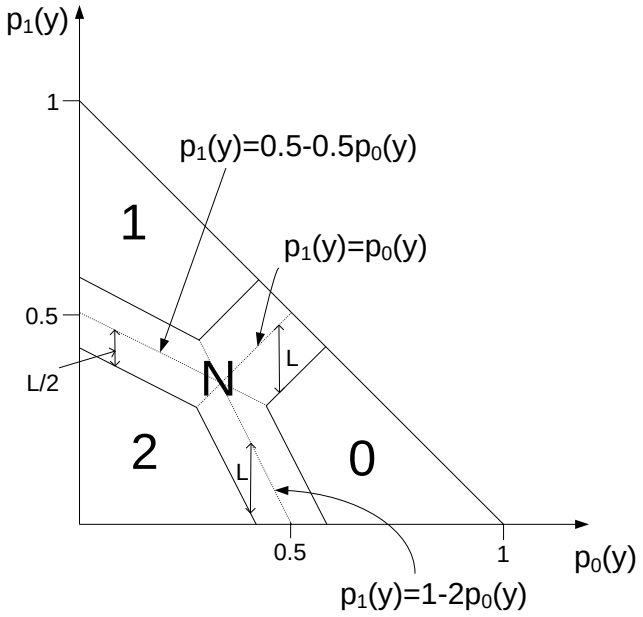

Figure 2.11. Spider plot one L.

In Figure 2.11 we see that we are making three separate sets of comparisons, a comparison between class 0 and class 1, class 0 and class 2 as well as class 1 and class 2, with each comparison having its own neutral zone region. Now if each one of these comparisons behaves in the same way as separate two-class cases we would need two distinct choices of L for each of our three comparisons, or six total choices of L. Updating (2.12) to reflect a three-class neutral zone classifier allowing for six choices of L would be:

$$\hat{C}(y;L)=\begin{cases}0 & \text{if } p_0(y)-p_1(y)>L_{01} \text{ and } p_0(y)-p_2(y)>L_{02}\\ 1 & \text{if } p_1(y)-p_0(y)>L_{10} \text{ and } p_1(y)-p_2(y)>L_{12}\\ 2 & \text{if } p_2(y)-p_0(y)>L_{20} \text{ and } p_2(y)-p_1(y)>L_{21}\\ N & \text{otherwise}\end{cases} \tag{2.15}$$

Another logical choice might be to allow for one L for each comparison, or three total choices of L. This would have the form:

$$\hat{C}(y;L)=\begin{cases}0 & \text{if } p_0(y)-p_1(y)>L_0 \text{ and } p_0(y)-p_2(y)>L_0\\ 1 & \text{if } p_1(y)-p_0(y)>L_1 \text{ and } p_1(y)-p_2(y)>L_1\\ 2 & \text{if } p_2(y)-p_0(y)>L_2 \text{ and } p_2(y)-p_1(y)>L_2\\ N & \text{otherwise}\end{cases} \tag{2.16}$$

In order to determine which of our three-class neutral zone classifiers, (2.12), (2.15) or (2.16) gives us the minimum expected cost we run a simulation for three densities and find our optimal L values. We use the following densities

$f_0(y)\sim N(\mu=1.0,\sigma=0.9)$, $f_1(y)\sim N(\mu=2.2,\sigma=0.8)$, $f_2(y)\sim N(\mu=4.2,\sigma=1.7)$

and the cost structure in Table 2.8 with $C_1=3$nd $C_2=2$ The results of the simulation are summarized in the following table:

	Minimizing *L* Values	Minimum Expected Cost	Run Time
1-*L*	0.1	0.5904834	16.3 seconds
3-*L*	0.2, 0.1, 0.1	0.5902864	192.1 seconds
6-*L*	0.2, 0.3, 0.1, 0.0, 0.3, 0.1	0.5902346	210314.2 seconds

We see that the six L neutral zone classifier does in fact give us the minimum expected cost but it takes 13,144 times longer to run than the one L case. At such a small gain in expected cost the additional run time makes the six L case impractical to implement in a real world setting. If, however, we use the neutral zone classifier proposed in **Definition 2** we have shown that we are able to obtain a classifier that is equivalent to the Bayes classifier and therefore obtains the minimum with a much quicker computation time. We are also able to obtain a nice visual representation of the classifier in **Definition 2** as we saw in Figure 2.3.

2.4. Neutral Zone Classification in Unsupervised Setting

2.4.1. Motivation

Yu et al. (2009) developed a method to handle a microbial community profiling application when labeled data exists. The outcome of the classifier in this application was a classification of either no binding (0), partial binding (1), complete binding (2), or a neutral outcome (*N*) when binding experiments between probes and genes were performed. As discussed in the introduction, recent changes in the methodologies in this

field have made it impractical to collect training data motivating the need for developing a neutral zone classification methodology within an unsupervised setting. In this section, we extend our neutral zone classifier methodology to handle this paradigm by combining a-priori knowledge about the correlation between *Y* and the binding status and an NDE algorithm developed by Benaglia et al. (2009).

2.4.2. Nonparametric Density Estimation

In order to implement the Bayes neutral zone classifier in (2.11) we need a method for determining the underlying class distributions. However, without training data we are unable to estimate these distributions directly. For our application, which we will discuss in more detail in the next section, our data is probe-to-polony binding intensity measurements. For each polony we measure the binding intensity (*Y*) to each probe and seek to classify those intensities as a 0, 1, 2 or *N*. The intensity observation can be organized into a matrix where the rows correspond to polonies and the columns correspond to the probes.

We classify based on the underlying mixture of class distributions (corresponding to the 0, 1 and 2 outcomes of the binding experiment) of each probe. Since we do not know the form of the component distributions, we will estimate them using the EM algorithm for NDE described in Benaglia et al. (2009a, 2009b) as follows. A separate EM algorithm for NDE is executed each probe in our data. Starting with *n* observations corresponding to one of the probes and, in our case, $m = 3$ classes, we will initialize an *n x m* matrix $P^0 = (\gamma_{ij}^0)$ where γ_{ij}^0 is the probability of the i^{th} observation belonging to the

j^{th} class at the 0^{th} iteration. To determine the initial P^0 matrix, we use a 3-class clustering algorithm (e.g., k-means algorithm) and arbitrarily assign class labels 0, 1 and 2 to the outputted clusters. This will leave P^0 to contain ones and zeros, which works well in practice (Benaglia et al., 2009a). Once we have our initial P^0 matrix each iteration of the EM algorithm for NDE consists of the following three steps:

1. For $t = 0$ initialize the $P^0 = (\gamma_{ij}^0)$ matrix using k-means where γ_{ij}^0 is the probability of the i^{th} observation belonging to the j^{th} class at the 0^{th} iteration.
2. The E-step: $\hat{\gamma}_{ij}^t = \dfrac{\lambda_j^t f_j^t(x_i)}{\sum_{k=1}^m \lambda_k^t f_k^t(x_i)}$. When $t = 0$ we skip this step.
3. The M-step:
    a. $\lambda_j^{t+1} = \sum_{i=1}^n \hat{\gamma}_{ij}^t / n$ where λ_j are the mixing proportions and t is our iteration number.
    b. $f_j^{t+1}(u) = \dfrac{1}{nh\lambda_j^{t+1}} \sum_{i=1}^n \hat{\gamma}_{ij}^t K\left(\dfrac{u - x_i}{h}\right)$ where h is a bandwidth chosen by the user and $K(\cdot)$ is a kernel density function. The kernel density function typically chosen is the standard normal density function and a suitable choice for $h = 0.9n^{-1/5} \min\left\{\text{SD}, \dfrac{\text{IQR}}{1.34}\right\}$ which is Silverman's rule of thumb.

In this section, we assume that n is large enough that the resulting NDEs from the EM algorithm are accurate. Once we have our NDEs for each class from EM algorithm, we can use them as inputs to the Bayes neutral zone classifier the same way a parametric density estimate would be used. A problem, however, lies in what labels to assign to each density estimate (Castelli and Cover, 1995). When executing the EM algorithm above, the initial step assigned class labels arbitrarily. Having prior knowledge about the ordering of the densities allows us to confidently assign these labels. In particular, Y is positively correlated with the class label since stronger intensity measurements correspond to more binding. Therefore, considering the three NDEs, we can label the distribution with the smallest mean to be class 0, the distribution with the second largest mean to be class 1 and the distribution with the largest mean to be class 2.

2.4.3. Neutral Zone Classifier

As an example, in Figure 2.12, we see an overlay of a histogram and an NDE of the intensity measurements for corresponding to probe 17. The mixing proportions for the three classes are .751, .121 and .127 from left to right in Figure 2.12.

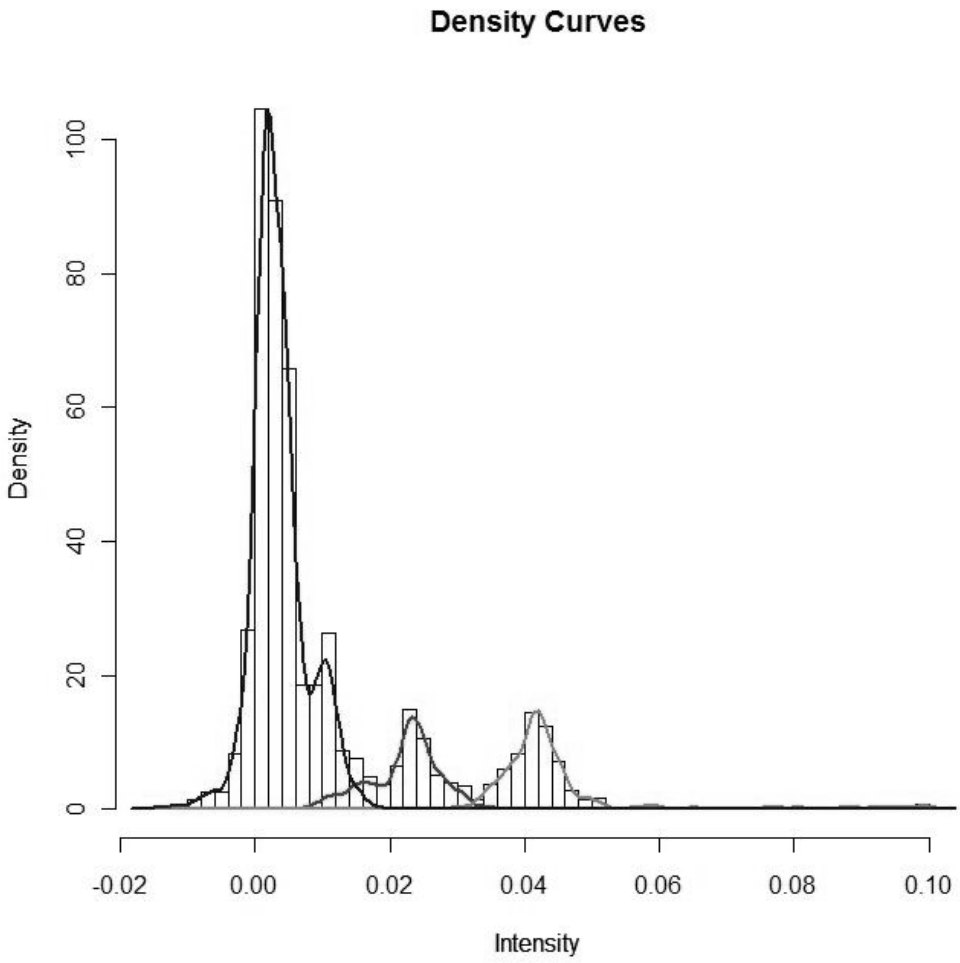

Figure 2.12. Fitted nonparametric density estimates
for the three-class mixture for Probe 17.

Using our prior knowledge about the ordering of the groups, we can use output from the last NDE-step in the EM algorithm for NDE to obtain estimates of $f_0(y)$, $f_1(y)$ and $f_2(y)$ for constructing the Bayes neutral zone classifier in (2.11). Figure 2.13 is the graphical representation of the resulting classifier using the cost structure shown by the

numerical values in Table 2.2. Evaluating $(p_0(y), p_1(y))$ for each polony, Figure 2.13 can be used classify each polony's binding status to probe 17 as a 0, 1, 2 or N.

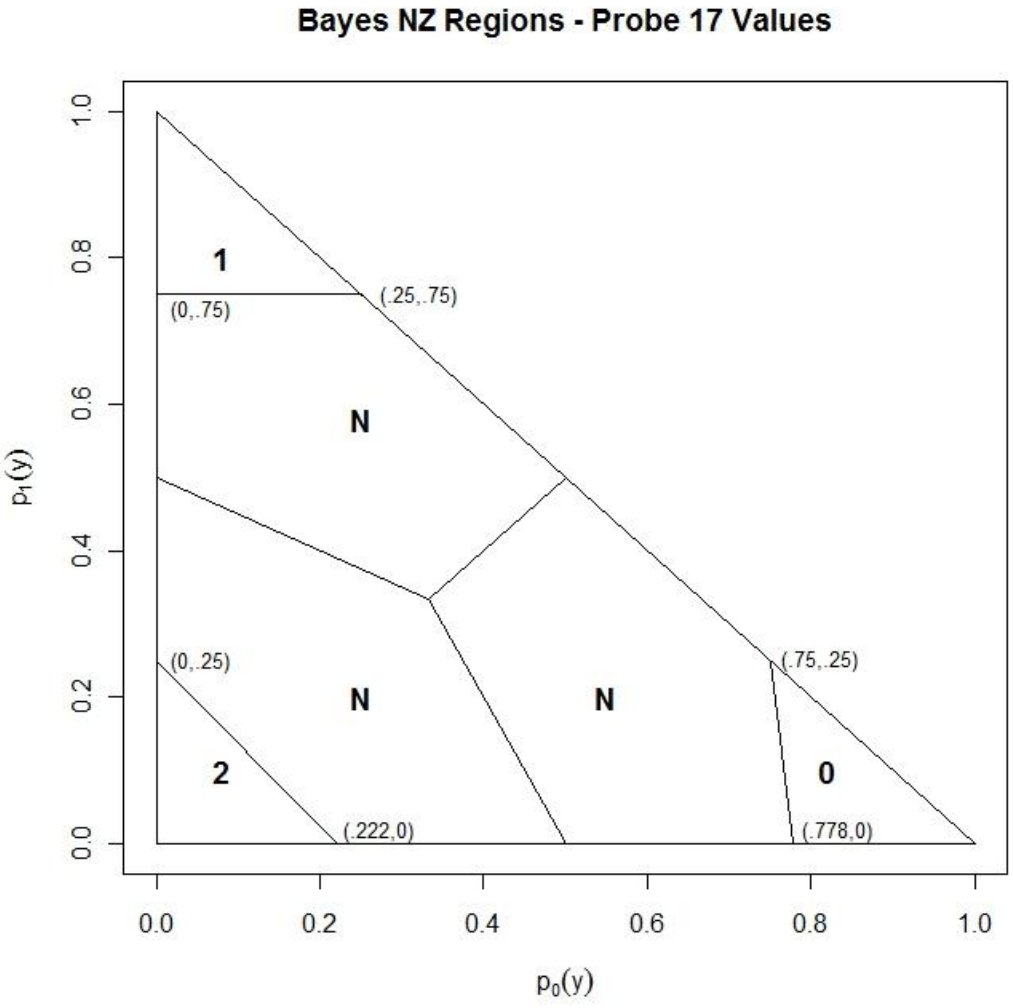

Figure 2.13. Bayes neutral zone classifier regions for probe 17 and the numerical cost structure in Table 2.2.

2.4.4. Cost and Computational Comparisons

Using the Bayes neutral zone classifier in (2.11) as opposed to the neutral zone classifier in (2.8) has two substantial benefits, improved accuracy and reduced runtime. We can use our example to compare the two classifiers by their estimated expected costs. The expected cost for the Bayes neutral zone classifier is obtained by evaluating (2.10) after inserting the NDEs, and the expected cost for the neutral zone classifier in (2.8) is obtained by evaluating (2.9) where, $P(\hat{C} = i \mid C = j) = \int_{A_i} f_j(y)\,dy$ where

$A_i = \{y : \hat{C}_{NZ}(y; L) = i\}$ is evaluated using Simpson's Rule where again the NDEs are used for $f_j(y)$.

For probe 17 the Bayes neutral zone classifier has an expected cost of 0.0791 while the neutral zone classifier in (2.8) has an expected cost of 0.1136. Also, the runtime for the Bayes neutral zone classifier was 5.85 seconds while the runtime for the neutral zone classifier (executed on the same computer, Dell Studio 1537, 2.40 GHz, 4.00 GB) was 330.14 seconds. Both runtimes include the times spent by the EM algorithm for the NDE. The results for all 40 probes are summarized in Table 2.9. As we can see the results vary by probe, however, the Bayes neutral zone classifier always performs at least as good as the neutral zone classifier in terms of expected cost and in some cases performs as much as three times better. Also, computation time of the Bayes neutral zone classifier is on the order of 1.7-23.3 minutes smaller than the time needed to compute the neutral zone classifier.

	Computation Time (sec)		Expected Cost			Computation Time (sec)		Expected Cost	
Probe	**Bayes**	**NZ**	**Bayes**	**NZ**	**Probe**	**Bayes**	**NZ**	**Bayes**	**NZ**
1	3.21	566.71	0.127	0.129	22	3.02	554.25	0.038	0.061
2	10.12	273.05	0.986	0.986	23	4.06	471.06	0.032	0.032
3	5.19	107.60	0.555	0.577	24	2.78	292.25	0.033	0.033
4	2.89	428.16	0.085	0.085	25	2.57	600.09	0.033	0.070
5	8.85	250.37	0.817	0.851	26	2.46	277.12	0.034	0.036
6	4.97	1403.90	0.086	0.999	27	18.29	118.62	0.991	0.991
7	2.08	205.71	0.019	0.022	28	2.71	359.10	0.007	0.007
8	1.98	212.82	0.012	0.015	29	3.18	512.40	0.079	0.079
9	3.12	695.68	0.017	0.019	30	2.67	569.59	0.004	0.008
10	2.73	443.69	0.044	0.047	31	1.71	224.35	0.008	0.014
11	3.71	418.41	0.062	0.065	32	2.47	638.22	0.118	0.146
12	18.99	281.36	0.992	1.156	33	3.33	560.04	0.005	0.006
13	1.93	193.76	0.067	0.069	34	3.76	781.50	0.059	0.061
14	3.51	659.95	0.005	0.005	35	10.05	317.82	0.767	0.907
15	22.62	220.57	0.591	0.695	36	2.96	375.44	0.444	0.448
16	4.87	313.31	0.097	0.110	37	2.65	170.97	0.094	0.094
17	5.85	330.14	0.079	0.113	38	3.09	509.56	0.009	0.009
18	10.61	653.49	0.308	0.999	39	2.25	354.82	0.007	0.007
19	14.85	208.27	0.849	0.849	40	12.90	686.78	0.260	0.488
20	2.49	405.53	0.003	0.006	41	2.98	397.58	0.008	0.009
21	3.00	392.03	0.005	0.005					

Table 2.9. Computation time and expected cost for each probe in polony example.

2.5. Summary

Neutral zone classifiers allow for a region of neutrality when the data is too ambiguous to confidently assign a predicted class. Previous versions of neutral zone classifiers have involved computationally complex methods for finding the boundaries for classification that minimize expected cost. We developed a neutral zone classifier from a Bayes point of view that significantly reduces the computation time for

classification while at the same time reducing the expected cost of the classifier. We extended the original application of neutral zone classifiers to cover the paradigm of unsupervised classification. To do this we, we incorporated use of an EM algorithm for NDE into the development of the Bayes neutral zone classifier. We demonstrated superior performance of the Bayes neutral zone classifier with respect to both estimated expected cost and computational complexity.

Chapter 3

Semi-Supervised Neutral Zone Classification

3.1. Introduction

Supervised learning methods require a large amount of labeled data to obtain adequate classification results. For instance, consider a two-class model, if there is only a small number of labeled data available then estimating class densities can often be difficult. Often the labels can be very time consuming or expensive to obtain and there are some instances where gathering labeled data is not possible or impractical, such as some microbial community profiling applications, which we discussed in the previous chapter. Because of the problems that arise in using solely supervised learning methods, it is an intriguing idea to supplement labeled data with unlabeled data. Unlabeled data alone are generally insufficient to yield a classification outcome that is better than a random choice because there is no information about the class label (Castelli and Cover, 1995). However, when unlabeled data is supplemented with labeled data the classification performance can often be improved dramatically.

When we are dealing with unlabeled data, the problem is not separating the data in classification groups. The problem lies in declaring which class those groups represent. In the case of a two-class unsupervised classification problem, we can often effectively separate our data into two groups but are unable to determine which of the groups to label

0 and which of the groups to label 1. Because the best we can do is make a random assignment, the probability of an error is equal to 0.5. However, if we can augment our unlabeled data with labeled data, even if we can only obtain one labeled observation, we can reduce the probability of an error (Castelli and Cover, 1995, Nigam et al., 1999).

This chapter deals with component density estimation in the semi-supervised setting and not directly with neutral zone classification. However, it is straightforward to apply the neutral zone classification methods introduced in the previous chapter once we have our component density estimates.

The rest of this chapter is set up as follows. Section 3.2. details the previous work by Castelli and Cover (1995) on the value of a single labeled observation. In Section 3.3. we outline the methodology for implementing semi-supervised classification for the case of a mixture of normal distributions and perform a simulation study. In Section 3.4., we consider how semi-supervised classification can be done in a nonparametric way. Section 3.5 compares the results of the parametric and nonparametric simulations.

3.2. Importance of Labeled Data

Determining the relationship between the number of labeled observations l and the number of unlabeled observations u in relation to the resulting probably of a classification error $R(l,u)$ was addressed by Castelli and Cover (1995). They showed that labeled data is exponentially more valuable than unlabeled data in reducing $R(l,u)$. The value of labeled data can be shown using the following framework. Suppose a label

$Y \in (0,1)$ has a distribution $P(Y=0)=\pi$ and $P(Y=1)=1-\pi$, and let the corresponding X be conditionally distributed with density $f_{X|Y}(x)=f_Y(x)$. Following this setup the labeled observations are distributed according to the joint density function

$$f_{X,Y}(x,y)=\left[\pi f_0(x)\right]^{1-y}\left[(1-\pi) f_1(x)\right]^{y}$$

The marginal distribution of X is the two-class mixture distribution

$$f(x)=\pi f_0(x)+(1-\pi) f_1(x).$$

Let a new observation (X_0,Y_0) be distributed from $f_{X,Y}(x,y)$ but for which only X_0 is observable. We want a prediction of Y_0, denoted by $\hat{Y}_0$. If $f_0(x)$, $f_1(x)$ and π are known, then an optimal classifier is given by the Bayes decision rule as follows

$$\hat{Y}_0=\begin{cases} 0 & \text{if } \dfrac{f_0(X_0)}{f_1(X_0)}>\dfrac{1-\pi}{\pi} \\ 1 & \text{if } \dfrac{f_0(X_0)}{f_1(X_0)}<\dfrac{1-\pi}{\pi} \end{cases}$$

and the Bayes risk R^*, or the probability of an error, is given by

$R^* \triangleq P\left(\hat{Y}_0 \neq Y_0\right)=\int_{R_1}\pi f_0(x)dx+\int_{R_0}(1-\pi) f_1(x)dx$ where R_i is the region where $\hat{Y}_0=i$.

When $f_0(x)$, $f_1(x)$ and π are unknown, however, the classification problem is not as simple as using the previously stated Bayes classifier. In order to gain an understanding of the value of one labeled observation we will assume that $f_0(x)$, $f_1(x)$ and π are unknown, but a training data set contains an infinite number of unlabeled samples, $\{X_1', X_2', \ldots\}$.

Since we have an infinite amount of unlabeled data it is known that we can estimate the mixture distribution $f_X(x)$ accurately (McLachlan and Krishnan, 1997). Despite being able to estimate the mixture distribution we are unable to construct a classification since we do not know which component of the mixture corresponds to which label. Rewrite the mixture distribution as $f_X(x) = \lambda g_0(x) + (1-\lambda) g_1(x)$ where $g_0(x)$ and $g_1(x)$ arbitrarily refer to the identified component densities. That is $g_0(x)$ refers to either $f_0(x)$ or $f_1(x)$. Likewise, λ is either π or $1-\pi$, but we do not know which. When we have an infinite number of unlabeled samples we can recover $g_0(x)$, $g_1(x)$ and λ and therefore recover $f_X(x)$. However, we are unable to determine if $(g_0, g_1, \lambda) = (f_0, f_1, \pi)$ or if $(g_0, g_1, \lambda) = (f_1, f_0, 1-\pi)$. To denote the two different cases we define the random variable Z to take the following values

$$Z = \begin{cases} 0 & \text{if } (g_0, g_1, \lambda) = (f_0, f_1, \pi) \\ 1 & \text{if } (g_0, g_1, \lambda) = (f_1, f_0, 1-\pi) \end{cases}$$

With no other information available, one would have to guess at Z, implying that $P(Z=0) = P(Z=1) = 0.5$ (Castelli and Cover, 1995), which corresponds to randomly assigning labels to the two components. In order to make a classification decision on X_0 when no labeled data is available we follow the following two-step procedure:

1. Randomly choose $Z \in \{0,1\}$
2. $$\hat{Y}_0 = (1-Z)\, I\left[\lambda g_0(x_0) < (1-\lambda) g_1(x_0)\right] + Z\, I\left[\lambda g_0(x_0) > (1-\lambda) g_1(x_0)\right]$$

The probability of an error $P\left(\hat{Y}_0 \neq Y_0\right)$, using the two-step procedure, can then be expressed as:

$$\begin{aligned} P(\text{error}) &= P(\text{error in step 1 AND no error in step 2}) \\ &\quad + P(\text{no error in step 1 AND error in step 2}) \\ &= P(\text{error in step 1})P(\text{no error in step 2}) + P(\text{no error in step 1})P(\text{error in step 2}) \\ &= 0.5\left[P(\text{no error in step 2}) + P(\text{error in step 2})\right] \\ &= 0.5 \end{aligned}$$

where we note that since we are guessing at Z in step 1 then $P(\text{error in step 1}) = 0.5$ and is independent of step 2.

When one labeled observation is introduced to the data, however, the misclassification rate can be reduced. The method of classification when we have one labeled observation (X_1, Y_1) is a two-step procedure developed by Castelli and Cover (1995) as follows:

1. $$Z = \begin{cases} 0 & \text{if } \lambda^{1-Y_1}(1-\lambda)^{Y_1} g_{Y_1}(X_1) > \lambda^{Y_1}(1-\lambda)^{1-Y_1} g_{1-Y_1}(X_1) \\ 1 & \text{if } \lambda^{1-Y_1}(1-\lambda)^{Y_1} g_{Y_1}(X_1) < \lambda^{Y_1}(1-\lambda)^{1-Y_1} g_{1-Y_1}(X_1) \end{cases}$$

2. $$\begin{aligned} \hat{Y}_0 &= (1-Z)\, I\left[\lambda g_0(x_0) < (1-\lambda) g_1(x_0)\right] \\ &\quad + Z\, I\left[\lambda g_0(x_0) > (1-\lambda) g_1(x_0)\right] \end{aligned}$$

Step 2 can be obtained by looking at the case when $Z = 0$ and $Z = 1$ separately. First, looking at the case when $Z = 0$ we have:

$$\hat{Y}_0 = \begin{cases} 0 & \text{if } \lambda g_0(X_0) > (1-\lambda) g_1(X_0) \\ 1 & \text{if } \lambda g_0(X_0) < (1-\lambda) g_1(X_0) \end{cases}$$

Then looking at the case when $Z = 1$ we have

$$\hat{Y}_0 = \begin{cases} 0 & \text{if } (1-\lambda) g_1(X_0) > \lambda g_0(X_0) \\ 1 & \text{if } (1-\lambda) g_1(X_0) < \lambda g_0(X_0) \end{cases}$$

which can be represented by the expression in step 2.

A classification error then occurs when either step 1 or step 2 gives an incorrect answer. For instance, consider an error in step 1 where $Y_1 = 0$ and we assign $Z = 1$ when the truth is $Z = 0$. In this scenario $(g_0, g_1, \lambda) = (f_0, f_1, \pi)$, however we have assigned $(g_0, g_1, \lambda) = (f_1, f_0, 1-\pi)$. Then in step 2 we are using the incorrect indicator function and will classify X_0 as the opposite of what is true. If, however, both steps are incorrect then the mistakes cancel each other out. Using the two-step procedure, Castelli and Cover (1995) show the probability of an error reduces from 0.5 to $2R^*(1-R^*)$ which is less than twice the Bayes risk of R^*.

3.3. Parametric Semi-Supervised Learning

In this section we examine the value of labeled data in the parametric setting for a two-class problem. In Section 3.3.1. the general form of the EM algorithm when both labeled and unlabeled data are present is developed. In Section 3.3.2. the EM algorithm is derived for a mixture of two normal distributions. Section 3.3.3. provides an illustrative example in the form of a simulation study and Section 3.3.4. compares the EM algorithm to the Castelli and Cover procedure. Finally, Section 3.3.5. looks at the EM algorithm for a mixture of exponential distributions.

3.3.1. General Form

In order to estimate the underlying class distribution in the data we will use the EM algorithm. For the case of semi-supervised learning, the log-likelihood function is modified to include both labeled and unlabeled data. Therefore, the general representation of the log-likelihood function in the semi-supervised setting is

$$\log l\left(\boldsymbol{\theta} \mid \left(X_i, Y_i\right)_{i=1}^{l}, \left(X_i\right)_{i=l+1}^{l+u}\right) = \sum_{i=1}^{l} \log\left(\left(1-y_i\right) f\left(x_i \mid \theta_0\right) + y_i\, f\left(x_i \mid \theta_1\right)\right) + \sum_{i=l+1}^{l+u} \log\left(\left(1-\pi\right) f\left(x_i \mid \theta_0\right) + \pi\, f\left(x_i \mid \theta_1\right)\right)$$

where $\left(X_i, Y_i\right)_{i=1}^{l}$ represents the labeled data, $\left(X_i\right)_{i=l+1}^{l+u}$ represents the unlabeled data and $\boldsymbol{\theta} = \left(\pi, \theta_0, \theta_1\right)$. Our initial data can be organized as follows:

i	X_i	Y_i
1	x_1	y_1
2	x_2	y_2
…	…	…
l	x_l	y_l
$l+1$	x_{l+1}	
…	…	
$l+u$	x_{l+u}	

Table 3.1. General form EM algorithm data format.

The EM algorithm for labeled and unlabeled data will proceed in the following way.

1. Set $t = 0$, where t represents the current iteration of the EM algorithm. Calculate the initial parameter estimates, in our case $\hat{\boldsymbol{\theta}}^0 = \left(\hat{\pi}^0, \hat{\theta}_0^0, \hat{\theta}_1^0\right)$, from the labeled $\left(X_l, Y_l\right)$ pairs using the maximum likelihood estimates. Therefore,

$$\hat{\pi}^0 = \frac{\sum_{i=1}^{l} y_i}{l}$$

$$\hat{\theta}_0^0 = \arg\max_{\theta_0} \sum_{i=1}^{l} \log\left(\left(1 - y_i\right) f\left(x_i \mid \theta_0\right)\right)$$

$$\hat{\theta}_1^0 = \arg\max_{\theta_1} \sum_{i=1}^{l} \log\left(y_i \; f\left(x_i \mid \theta_1\right)\right)$$

2. E-step: Compute the predicted label probabilities

$$\hat{\gamma}_i^t = P\left(y_i = 1 \mid x_i, \hat{\boldsymbol{\theta}}^t\right) = \frac{\hat{\pi}^t f\left(x_i \mid \hat{\theta}_1^t\right)}{\left(1 - \hat{\pi}^t\right) f\left(x_i \mid \hat{\theta}_0^t\right) + \hat{\pi}^t f\left(x_i \mid \hat{\theta}_1^t\right)}$$

for all $x \in X_u$. Note that $p\left(y_i = 0 \mid x_i, \hat{\boldsymbol{\theta}}^t\right) = 1 - \hat{\gamma}_i^t$.

3. M-step: Update the parameter estimates using the maximum likelihood estimates. Therefore,

$$\hat{\theta}^{t+1} = \arg\max_{\theta} \log l\left(\boldsymbol{\theta} \mid \left(X_i, Y_i\right)_{i=1}^{l}, \left(X_i, \hat{\gamma}_i^t\right)_{i=l+1}^{l+u}\right)$$

4. Iterate steps 2 and 3, setting $t = t + 1$ before each iteration, until $\log l\left(\hat{\boldsymbol{\theta}}^{t+1} \mid \left(X_i, Y_i\right)_{i=1}^{l}, \left(X_i, \hat{\gamma}_i^{t+1}\right)_{i=l+1}^{l+u}\right) - \log l\left(\hat{\boldsymbol{\theta}}^t \mid \left(X_i, Y_i\right)_{i=1}^{l}, \left(X_i, \hat{\gamma}_i^t\right)_{i=l+1}^{l+u}\right) < \delta$ where δ is an acceptable level of convergence (e.g. $\delta = 10\mathrm{e}^{-5}$).

After each iteration of the EM algorithm our data will look as follows:

i	X_i	Y_i	$P(Y_i=0)$	$P(Y_i=1)$
1	x_1	y_1	$1-y_1$	y_1
2	x_2	y_2	$1-y_2$	y_2
…	…	…	…	…
l	x_l	y_l	$1-y_l$	y_l
$l+1$	x_{l+1}		$1-\hat{\gamma}_{l+1}^t$	$\hat{\gamma}_{l+1}^t$
…	…		…	…
$l+u$	x_{l+u}		$1-\hat{\gamma}_{l+u}^t$	$\hat{\gamma}_{l+u}^t$

Table 3.2. General form EM algorithm data iteration format.

3.3.2. Normal Mixture

For our illustration we are interested in two versions of the EM algorithm when both labeled and unlabeled data are present, one for a known Normal mixture and one for a nonparametric mixture. In this section we will look at the first version of interest, the two-class Normal mixture. In the previous section the general form for the EM algorithm was detailed. Applying that general form to the two-class Normal mixture situation yields a log-likelihood function of

$$\log l\left(\boldsymbol{\theta} \mid \left(X_i, Y_i\right)_{i=1}^{l}, \left(X_i\right)_{i=l+1}^{l+u}\right) = \sum_{i=1}^{l} \log\left(\left(1-y_i\right) f\left(x_i \mid \mu_0, \sigma_0\right) + y_i\, f\left(x_i \mid \mu_1, \sigma_1\right)\right) + \sum_{i=l+1}^{l+u} \log\left(\left(1-\pi\right) f\left(x_i \mid \mu_0, \sigma_0\right) + \pi f\left(x_i \mid \mu_1, \sigma_1\right)\right)$$

where $\left(X_i, Y_i\right)_{i=1}^{l}$ represents the labeled data, $\left(X_i\right)_{i=l+1}^{l+u}$ represents the unlabeled data and $\boldsymbol{\theta} = \left(\pi, \theta_0, \theta_1\right) = \left(\pi, \mu_0, \sigma_0, \mu_1, \sigma_1\right)$. As in the general case, our data can be organized as follows:

i	X_i	Y_i
1	x_1	y_1
2	x_2	y_2
…	…	…
l	x_l	y_l
$l+1$	x_{l+1}	
…	…	
$l+u$	x_{l+u}	

Table 3.3. Normal mixture EM algorithm data format.

The EM algorithm when labeled and unlabeled data are present for the two-class Normal mixture will proceed in the following way.

1. Set $t=0$, where t represents the current iteration of the EM algorithm. Calculate the initial parameter estimates, in our case $\hat{\boldsymbol{\theta}}^0=\left(\hat{\pi}^0,\hat{\theta}_0^0,\hat{\theta}_1^0\right)=\left(\hat{\pi}^0,\hat{\mu}_0^0,\hat{\sigma}_0^0,\hat{\mu}_1^0,\hat{\sigma}_1^0\right)$, from the labeled $\left(X_l,Y_l\right)$ pairs using the maximum likelihood estimates.

   Therefore,

$$\hat{\pi}^0=\frac{\sum_{i=1}^{l}y_i}{l}$$

$$\hat{\mu}_0^0=\frac{\sum_{i=1}^{l}\left(1-y_i\right)x_i}{\sum_{i=1}^{l}\left(1-y_i\right)}$$

$$\hat{\sigma}_0^0=\sqrt{\frac{\sum_{i=1}^{l}\left(1-y_i\right)\left(x_i-\hat{\mu}_0^0\right)^2}{\sum_{i=1}^{l}\left(1-y_i\right)}}$$

$$\hat{\mu}_1^0 = \frac{\sum_{i=1}^{l} y_i \; x_i}{\sum_{i=1}^{l} y_i}$$

$$\hat{\sigma}_1^0 = \sqrt{\frac{\sum_{i=1}^{l} y_i \left(x_i - \hat{\mu}_1^0 \right)^2}{\sum_{i=1}^{l} y_i}}$$

2. E-step: Compute the predicted label probabilities

$$\hat{\gamma}_i^t = p\left(y_i = 1 \mid x_i, \hat{\boldsymbol{\theta}}^t \right) = \frac{\hat{\pi}^t f\left(x_i \mid \hat{\mu}_1^t, \hat{\sigma}_1^t \right)}{\left(1 - \hat{\pi}^t \right) f\left(x_i \mid \hat{\mu}_0^t, \hat{\sigma}_0^t \right) + \hat{\pi}^t f\left(x_i \mid \hat{\mu}_1^t, \hat{\sigma}_1^t \right)}$$

for all $x \in \left(X_i \right)_{i=l+1}^{l+u}$. Note that $p\left(y_i = 0 \mid x_i, \hat{\boldsymbol{\theta}}^t \right) = 1 - \hat{\gamma}_i^t$.

3. M-step: Update the parameter estimates using the weighted means and variances. Therefore,

$$\hat{\mu}_0^{t+1} = \frac{\sum_{i=1}^{l} \left(1 - y_i \right) x_i + \sum_{i=l+1}^{u} \left(1 - \hat{\gamma}_i^t \right) x_i}{\sum_{i=1}^{l} \left(1 - y_i \right) + \sum_{i=l+1}^{u} \left(1 - \hat{\gamma}_i^t \right)}$$

$$\hat{\sigma}_0^{t+1} = \sqrt{\frac{\sum_{i=1}^{l} \left(1 - y_i \right) \left(x_i - \hat{\mu}_0^{t+1} \right)^2 + \sum_{i=l+1}^{u} \left(1 - \hat{\gamma}_i^t \right) \left(x_i - \hat{\mu}_0^{t+1} \right)^2}{\sum_{i=1}^{l} \left(1 - y_i \right) + \sum_{i=l+1}^{u} \left(1 - \hat{\gamma}_i^t \right)}}$$

$$\hat{\mu}_1^{t+1} = \frac{\sum_{i=1}^{l} y_i \; x_i + \sum_{i=l+1}^{u} \hat{\gamma}_i^t x_i}{\sum_{i=1}^{l} y_i + \sum_{i=l+1}^{u} \hat{\gamma}_i^t}$$

$$\hat{\sigma}_1^{t+1} = \sqrt{\frac{\sum_{i=1}^{l} y_i \left(x_i - \hat{\mu}_1^{t+1} \right)^2 + \sum_{i=l+1}^{u} \hat{\gamma}_i^t \left(x_i - \hat{\mu}_1^{t+1} \right)^2}{\sum_{i=1}^{l} y_i + \sum_{i=l+1}^{u} \hat{\gamma}_i^t}}$$

$$\hat{\pi}^{t+1} = \frac{\sum_{i=1}^{l} y_i + \sum_{i=l+1}^{l+u} \hat{\gamma}_i^t}{l + u}$$

4. Iterate steps 2 and 3, setting $t = t+1$ before each iteration, until

$$\log l\left(\hat{\boldsymbol{\theta}}^{t+1} \mid \left(X_i, Y_i\right)_{i=1}^{l}, \left(X_i, \hat{\gamma}_i^{t+1}\right)_{i=l+1}^{l+u}\right) - \log l\left(\hat{\boldsymbol{\theta}}^{t} \mid \left(X_i, Y_i\right)_{i=1}^{l}, \left(X_i, \hat{\gamma}_i^{t}\right)_{i=l+1}^{l+u}\right) < \delta \text{ where}$$

δ is an acceptable level of convergence (e.g. $\delta = 10\text{e}^{-5}$).

As in the previous section, after each iteration of the EM algorithm our data will look as follows:

i	X_i	Y_i	$P(Y_i = 0)$	$P(Y_i = 1)$
1	x_1	y_1	$1 - y_1$	y_1
2	x_2	y_2	$1 - y_2$	y_2
…	…	…	…	…
l	x_l	y_l	$1 - y_l$	y_l
$l+1$	x_{l+1}		$1 - \hat{\gamma}_{l+1}^t$	$\hat{\gamma}_{l+1}^t$
…	…		…	…
$l+u$	x_{l+u}		$1 - \hat{\gamma}_{l+u}^t$	$\hat{\gamma}_{l+u}^t$

Table 3.4. Normal mixture EM algorithm data iteration format.

3.3.3. Simulation Study

Previous literature has established that supplementing unlabeled data with labeled data will improve classification results. We provide numerical insight, when dealing with a two-class Normal mixture with common variance, as to the situations when performance is improved most in relation to the Mahalanobis distance between the two classes. Mahalanobis distance is defined as $d = \left|\mu_1 - \mu_2\right| / \sigma$. For each Mahalanobis distance that we test $(0.5, 1, 2, 4)$, various unlabeled and labeled data sample size combinations are examined in order to find the smallest labeled data sample size that provides a significant improvement over classification that uses solely labeled data. It is

important to note that typically the results of unsupervised methods provide no better than a random outcome (Castelli and Cover, 1995), however, as explained in the previous chapter regarding the microbial community profiling application we will assume that unsupervised methods are of value since we have prior knowledge of group labels based on the ordering of their respective means.

The simulation study is setup in the following manner. First, choose parameter values from which the data will be simulated. For a normal mixture in the two-class case the density function with common variance can be expressed as $f\left(x;\pi_0,\pi_1,\mu_0,\mu_1,\sigma\right)=\pi_0 N\left(x;\mu_0,\sigma\right)+\pi_1 N\left(x;\mu_1,\sigma\right)$ therefore the parameter values we must choose are $\theta=\left(\pi_0,\pi_1,\mu_0,\mu_1,\sigma\right)$. Then, after the parameter values are determined, we must simulate three sets of data: labeled data, unlabeled data and a test dataset which will be used to evaluate classification accuracy. The sample sizes of both the labeled data and unlabeled data will be varied to investigate how the accuracy varies. The sample size of the test set is chosen to be 10,000 so that the estimated misclassification rates are trustworthy. For each combination of labeled and unlabeled data we average the error rates on the test set over 25 simulation replicates.

After the data sets are simulated the EM algorithm is executed as described in Section 3.3.2. Once there is convergence in the EM algorithm and our respective class densities are estimated we classify the test set using a standard Bayes classifier. In order to show that augmenting unlabeled data with labeled data improves the classification error rates we calculated the average error rates of the classifier over all the simulation replications and categorized the situations based on the Mahalanobis distance,

$d = |\mu_1 - \mu_2| / \sigma$, between the two class densities. Based on our simulations there was not a significance difference in the error rate when different parameter values were used but the Mahalanobis distance was equal.

The tables below show the results for four different Mahalanobis distances, 0.5, 1, 2 and 4. For instance, when the Mahalanobis distance between the two groups is 0.5 then we see a significant improvement when up to 200 labeled observations are added to 100 and 500 unlabeled observations and when up to 100 labeled observations are added to 1,000 and 10,000 unlabeled observations. This guideline tells us that when we are dealing with either 100 or 500 unlabeled observations it is best to augment with up to 200 labeled observations and when we are dealing with either 1,000 or 10,000 unlabeled observations it is best to augment with up to 100 labeled observations when our Mahalanobis distance is 0.5. After these points there is a diminishing return on the value of additional labeled observations. The plot below shows this relationship graphically when the Mahalanobis distance is 0.5.

For the case when the Mahalanobis distance is 1 then we see a leveling off of the benefit of additional labeled data points for all unlabeled sample sizes at 100 labeled observations. Therefore, if our data set has a Mahalanobis distance of 1 then we would see the best cost benefit in improving our classification error rate if we were able to obtain up to 100 labeled observations. This relationship is illustrated in the figure below labeled for a Mahalanobis distance of 1. When the Mahalanobis distance between the two groups is 2, however, we see from the table below that we only need 5 labeled observations before we see the first significant improvement in the classification error

rate. In addition, when the Mahalanobis distance between the two groups is 4, we only need 1 labeled observation to realize a significant improvement in classification error rate. This relationship is illustrated in the figures below for a Mahalanobis distance of 2 and 4, respectively.

The results illustrated in the tables below indicate the value of semi-supervised classification and show, in relationship to Mahalanobis distance, the amount of labeled data necessary before we have seen a leveling off of the classification error rates, meaning that by adding more labeled data we will no longer see any significant improvement in the classification results. We can see as per the earlier discussion that adding even 1 labeled observation will yield significantly better results than the case when there are no labeled observations. For example, the Mahalanobis distance of 0.5 (e.g. $\theta = \left(\pi_0 = 0.5, \pi_1 = 0.5, \mu_0 = 1, \mu_1 = 3, \sigma = 4\right)$) has a true error rate of $R^* = 0.401295 = 0.5\int_2^{\infty} N\left(x;1,4\right)dx + 0.5\int_{-\infty}^{2} N\left(x;3,4\right)dx$, and therefore $2R^*\left(1-R^*\right) = 0.4805$ which is close to the simulated values of the error rate when 1 labeled observation is present in the data.

Error Rates (Mahalanobis Distance = 0.5)				
$\theta = (\pi_0 = 0.5, \pi_1 = 0.5, \mu_0 = 1, \mu_1 = 3, \sigma = 4)$				
	Unlabeled			
Labeled	**100**	**500**	**1000**	**10000**
0	0.4976	0.4983	0.5055	0.5019
1	0.4791	0.4601	0.4698	0.4677
2	0.4745	0.4569	0.4673	0.4626
5	0.4721	0.4546	0.4649	0.4603
10	0.4697	0.4524	0.4626	0.4580
20	0.4614	0.4417	0.4532	0.4487
50	0.4360	0.4343	0.4408	0.4363
100	0.4321	0.4252	0.4132	0.4091
200	0.4106	0.4042	0.4112	0.4071
500	0.4017	0.4032	0.4017	0.3977
1000	0.4035	0.4008	0.4037	0.3997
	margin of error = .01			

Table 3.5. Parametric semi-supervised classification results Mahalanobis distance = 0.5.

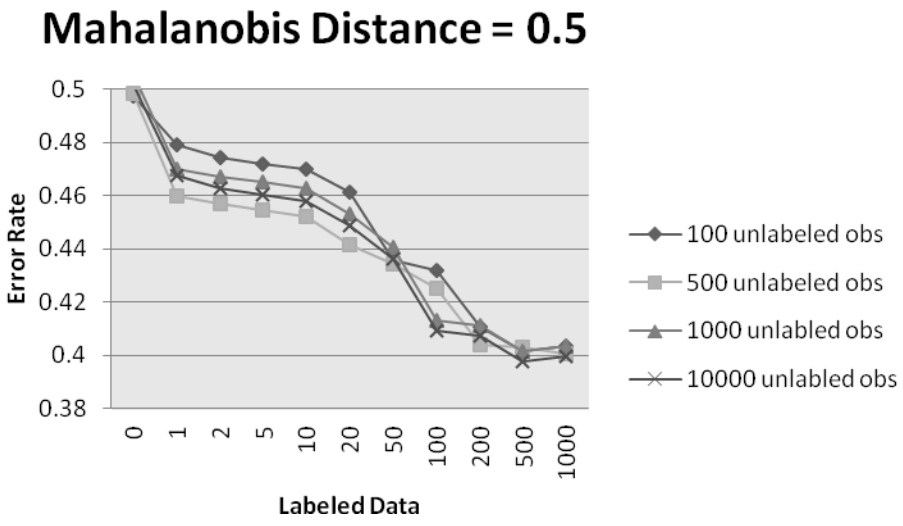

Figure 3.1. Parametric semi-supervised classification results Mahalanobis distance = 0.5.

Error Rates (Mahalanobis Distance = 1)				
$\theta = (\pi_0 = 0.5, \pi_1 = 0.5, \mu_0 = 1, \mu_1 = 3, \sigma = 2)$				
	Unlabeled			
Labeled	**100**	**500**	**1000**	**10000**
0	0.4935	0.5091	0.5012	0.4987
1	0.4409	0.4113	0.4129	0.4145
2	0.4384	0.4060	0.4040	0.4036
5	0.3715	0.3562	0.3672	0.3636
10	0.3696	0.3544	0.3654	0.3618
20	0.3496	0.3548	0.3373	0.3339
50	0.3394	0.3378	0.3338	0.3325
100	0.3113	0.3130	0.3155	0.3124
200	0.3102	0.3097	0.3099	0.3068
500	0.3090	0.3093	0.3079	0.3049
1000	0.3092	0.3084	0.3097	0.3066
	margin of error = .01			

Table 3.6. Parametric semi-supervised classification results Mahalanobis distance = 1.

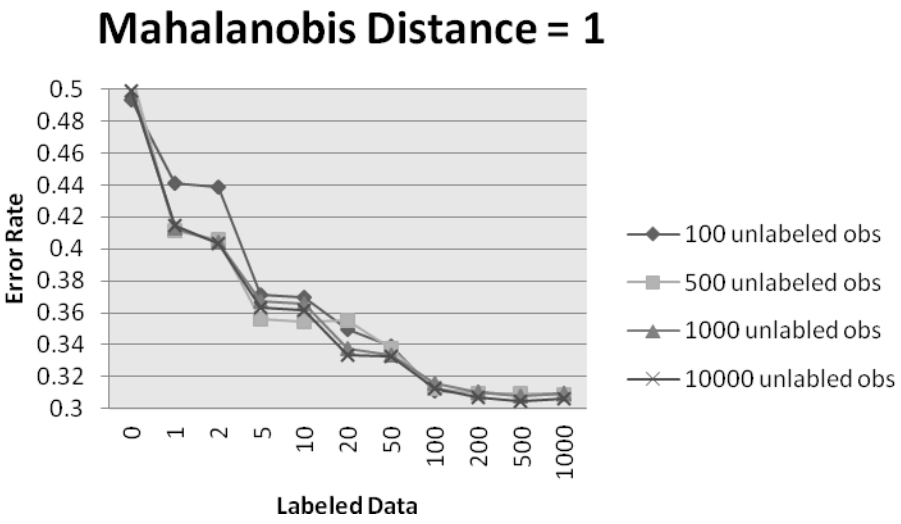

Figure 3.2. Parametric semi-supervised classification results Mahalanobis distance = 1.

Error Rates (Mahalanobis Distance = 2)				
$\theta = (\pi_0 = 0.5, \pi_1 = 0.5, \mu_0 = 1, \mu_1 = 3, \sigma = 1)$				
	Unlabeled			
Labeled	**100**	**500**	**1000**	**10000**
0	0.5042	0.4991	0.5034	0.4991
1	0.2445	0.2367	0.2198	0.2031
2	0.2433	0.2273	0.2171	0.1978
5	0.2028	0.1748	0.1737	0.1720
10	0.2017	0.1740	0.1728	0.1711
20	0.1715	0.1633	0.1667	0.1650
50	0.1627	0.1632	0.1655	0.1639
100	0.1605	0.1631	0.1628	0.1612
200	0.1598	0.1580	0.1609	0.1593
500	0.1594	0.1593	0.1580	0.1565
1000	0.1584	0.1587	0.1580	0.1564
	margin of error = .01			

Table 3.7. Parametric semi-supervised classification results Mahalanobis distance = 2.

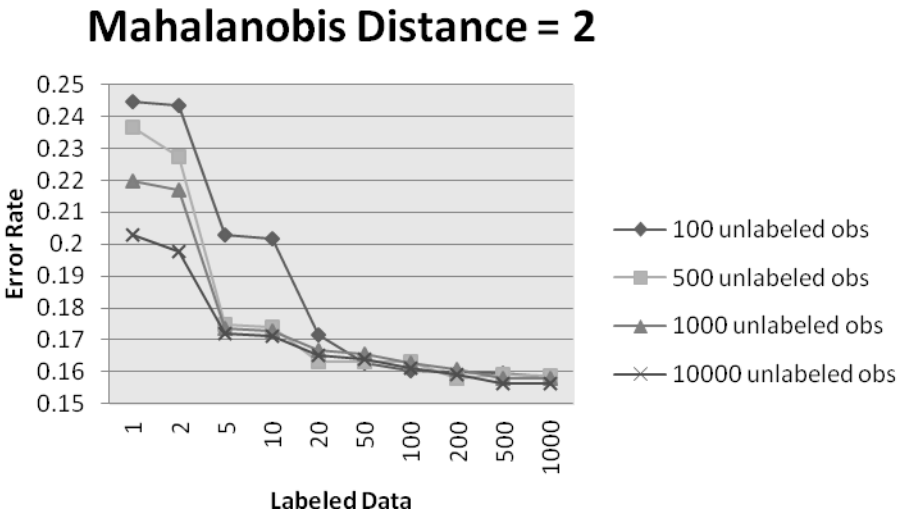

Figure 3.3. Parametric semi-supervised classification results Mahalanobis distance = 2.

Error Rates (Mahalanobis Distance = 4)				
$\theta = (\pi_0 = 0.5, \pi_1 = 0.5, \mu_0 = 1, \mu_1 = 3, \sigma = 0.5)$				
	Unlabeled			
Labeled	**100**	**500**	**1000**	**10000**
0	0.5010	0.4975	0.4963	0.5034
1	0.0559	0.0367	0.0349	0.0328
2	0.0541	0.0312	0.0308	0.0304
5	0.0451	0.0240	0.0237	0.0234
10	0.0449	0.0239	0.0235	0.0232
20	0.0267	0.0240	0.0233	0.0231
50	0.0246	0.0231	0.0234	0.0232
100	0.0244	0.0228	0.0230	0.0228
200	0.0235	0.0234	0.0226	0.0224
500	0.0233	0.0233	0.0225	0.0223
1000	0.0231	0.0223	0.0225	0.0223
	margin of error = .01			

Table 3.8. Parametric semi-supervised classification results Mahalanobis distance = 4.

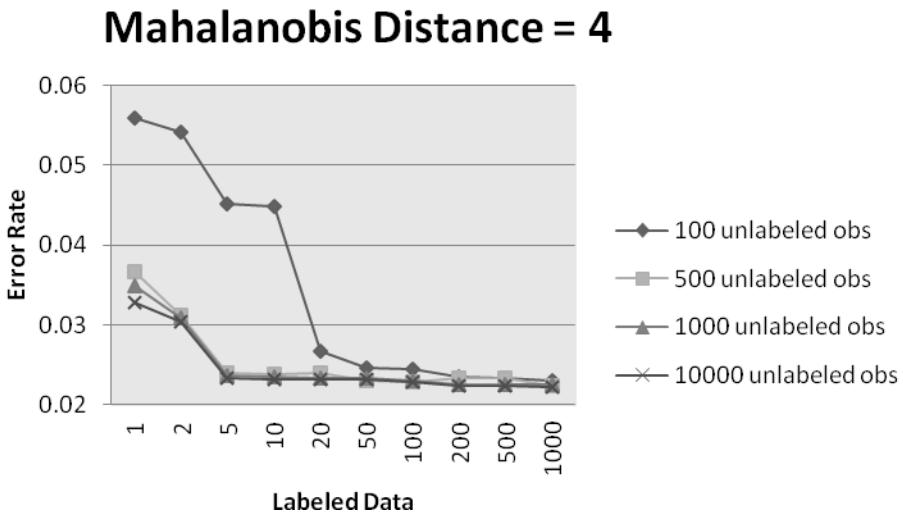

Figure 3.4. Parametric semi-supervised classification results Mahalanobis distance = 4.

3.3.4. One Labeled Observation Comparison

While the EM algorithm outlined in Section 3.3.3. would require at least 2 labeled observations from each component in order to calculate initial parameter estimates we can get around this issue by performing the EM algorithm for a grid of starting values and choosing the result with the maximum likelihood as our final component estimates (Nigam et al., 1999). To illustrate this consider a situation where the Mahalanobis distance is 4, for instance $(\pi_0 = \pi_1 = 0.5, \mu_0 = 5, \mu_1 = 9, \sigma_0 = \sigma_1 = 1)$. Simulating unlabeled data of size 100,000 and one labeled observation from group 0 we get unique maximum likelihood estimates for our parameters as $(\hat{\pi}_0 = 0.499, \hat{\pi}_1 = 0.501, \hat{\mu}_0 = 5.002, \hat{\mu}_1 = 9.004, \hat{\sigma}_0 = 0.993, \hat{\sigma}_1 = 1.001)$. Once we have our estimates we can then proceed to the classification step.

Next we will compare the EM algorithm to the procedure outlined by Castelli and Cover (1995). We use the same population parameters and unlabeled data sample size of 100,000 to ensure accurate parameter estimates which is necessary for the Castelli and Cover procedure. After estimating the parameters and performing the classification step the EM algorithm has a misclassification rate of 0.0227 and the Castelli and Cover procedure has a misclassification rate of 0.0227. This suggests the EM algorithm, together with the subsequent Bayes classifier, is equivalent to the Castelli and Cover procedure.

3.3.5. Semi-Supervised Exponential Simulation

In Section 3.3.3. we looked at the simulation results using a mixture of two Normal distributions to determine if classification results could be improved by adding labeled data to unlabeled data. The resulting semi-supervised classification showed that for various Mahalanobis distances adding labeled data up to a certain amount would improve the classification error rates. In order to determine if other distributions behave similarly we looked at a mixture of two exponential distributions. To compare the results of the normal distribution and exponential distribution we will use the Kullback-Leibler divergence which is defined as

$$E\left[\log\frac{f_1(x)}{f_0(x)}\right]$$

The Kullback-Leibler divergence for a Normal distribution is $\frac{1}{2\sigma^2}(\mu_1-\mu_0)^2$ and for an exponential distribution is $\frac{\mu_1}{\mu_0}-\log\left(\frac{\mu_1}{\mu_0}\right)-1$. Therefore, for our example with a Mahalanobis distance of 1 our parameters for the Normal distribution are $\theta=(\pi_0=0.5,\pi_1=0.5,\mu_0=1,\mu_1=3,\sigma=2)$ which gives a Kullback-Leibler divergence of 0.5. Using a mixture of exponential distributions with mean equal to 1 and 2.35 and mixing proportion equal to 0.5 we also have Kullback-Leibler divergence of 0.5. Using these parameters and repeating the simulation experiment for the exponential mixture we see the error rate improve by adding up to 100 labeled observations. The results of this simulation are similar to that of the Normal setting with the same Kullback-Leibler

divergence where we also saw significant improvement for up to 100 labeled observations. This provides evidence that for different distributions the results of adding more labeled observations will be similar.

Error Rates (Exponential Mean = (1,2.35))				
	Unlabeled			
Labeled	**100**	**500**	**1000**	**10000**
2	0.4719	0.4589	0.4290	0.4272
5	0.4765	0.4165	0.4197	0.4093
10	0.4237	0.4195	0.4093	0.4116
20	0.3783	0.3746	0.3713	0.3749
50	0.3694	0.3639	0.3633	0.3634
100	0.3507	0.3478	0.3534	0.3528
200	0.3503	0.3524	0.3477	0.3496
500	0.3486	0.3484	0.3477	0.3506
1000	0.3480	0.3483	0.3467	0.3495
	margin of error = .01			

Table 3.9. Parametric semi-supervised classification results for an exponential mixture.

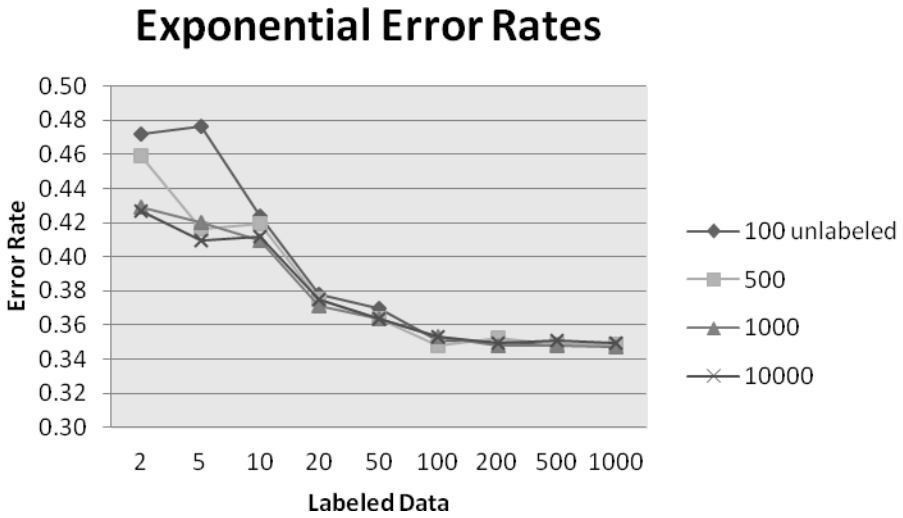

Figure 3.5. Parametric semi-supervised classification results for an exponential mixture.

3.4. Nonparametric Semi-Supervised Learning

In this section we examine the value of labeled data in the two-class nonparametric setting. In Section 3.4.1. a nonparametric EM algorithm for semi-supervised learning is presented. Section 3.4.2. provides an illustrative example in the form of a simulation study.

3.4.1. Nonparametric EM Algorithm

In this section we introduce a nonparametric EM algorithm when both labeled and unlabeled data are present for a two-class mixture model. In Chapter 2, we introduced the nonparametric EM algorithm developed by Benaglia et al. (2009a, 2009b). The nonparametric EM algorithm is an unsupervised learning method, however when labeled data is present we can utilize that information in combination with the unlabeled data and improve on the classification results of the unsupervised method.

The nonparametric EM algorithm for semi-supervised learning is setup in the following way for the case when we have $m = 2$ classes. Starting with l labeled observations and u unlabeled observations, we will initialize an $n \times 2$ matrix $P^0 = (p_{ij}^0)$ where p_{ij}^0 is the probability of the i^{th} observation belonging to the j^{th} class at the 0^{th} iteration and $n = l + u$. As in the unsupervised setting, an initial P^0 matrix must also be determined in the semi-supervised setting. In the unsupervised setting, a clustering algorithm such as the k-means algorithm is used to arbitrarily assign class labels which leaves P^0 to contain ones and zeroes (Benaglia et al., 2009a). In the semi-supervised

setting we again will use a clustering algorithm to assign class labels, however, the labels will not be arbitrary. Instead of arbitrary labels, the labeled data is utilized in determining the initial P^0 matrix. To do this we use seeded k-means, which is a semi-supervised k-means clustering algorithm (Basu et al., 2002). As in the parametric case our data can be organized as follows:

i	X_i	Y_i
1	x_1	y_1
2	x_2	y_2
…	…	…
l	x_l	y_l
$l+1$	x_{l+1}	
…	…	
$l+u$	x_{l+u}	

Table 3.10. Nonparametric EM algorithm data format.

Utilizing labeled data to perform seeding in the k-means algorithm works in the following way. Taking a dataset X, the k-means algorithm generates $m=2$ clusters $\{X_j\}_{j=1}^{2}$ of X so that the k-means objective function is locally minimized where the k-means objective function is given by

$$\sum_{j=1}^{m}\sum_{x_i\in X_j}\left\|x_i-\mu_j\right\|^2$$

where μ_j is the mean of the observations in X_j. Now let X_u be the set of unlabeled data and X_l be the set of labeled data, or the seed set, where the labels for X_l are denoted by Y_l. We assume that corresponding to each desired partition X_j of X there is at least one labeled observation. Each labeled observation is then used to guide the

seeded k-means algorithm as follows. Rather than initializing the k-means algorithm from $m = 2$ random means, the initial means are determined as the averages of the labeled observations (Basu et al., 2002).

Once we have completed the semi-supervised or seeded k-means initialization of the P^0 matrix each iteration of the nonparametric EM algorithm for NDE consists of the following three steps:

1. For $t = 0$ initialize the $P^0 = (\gamma_{ij}^0)$ matrix using seeded k-means where γ_{ij}^0 is the probability of the i^{th} observation belonging to the j^{th} class at the 0^{th} iteration.
2. The E-step: $\hat{\gamma}_{ij}^t = \dfrac{\lambda_j^t f_j^t(x_i)}{\sum_{k=1}^m \lambda_k^t f_k^t(x_i)}$. When $t = 0$ we skip this step.
3. The M-step:
   a. $\lambda_j^{t+1} = \sum_{i=1}^n \hat{\gamma}_{ij}^t / n$ where λ_j are the mixing proportions and t is our iteration number.
   b. $f_j^{t+1}(u) = \dfrac{1}{nh\lambda_j^{t+1}} \sum_{i=1}^n \hat{\gamma}_{ij}^t K\left(\dfrac{u - x_i}{h}\right)$ where h is a bandwidth chosen by the user and $K(\cdot)$ is a kernel density function. The kernel density function typically chosen is the standard normal density function and a suitable choice for $h = 0.9n^{-1/5} \min\left\{\text{SD}, \dfrac{\text{IQR}}{1.34}\right\}$ which is Silverman's rule of thumb.

Note that these three steps are the same as the nonparametric EM algorithm introduced in Chapter 2 and that the semi-supervised learning comes from the utilization of the seeded k-means algorithm. After each iteration of the nonparametric EM algorithm our data will look as follows:

i	X_i	Y_i	$P(Y_i=0)$	$P(Y_i=1)$
1	x_1	y_1	$1-y_1$	y_1
2	x_2	y_2	$1-y_2$	y_2
…	…	…	…	…
l	x_l	y_l	$1-y_l$	y_l
$l+1$	x_{l+1}		$1-\hat{\gamma}_{l+1}^t$	$\hat{\gamma}_{l+1}^t$
…	…		…	…
$l+u$	x_{l+u}		$1-\hat{\gamma}_{l+u}^t$	$\hat{\gamma}_{l+u}^t$

Table 3.11. Nonparametric EM algorithm data iteration form.

3.4.2. Simulation Study

In the previous section we outlined a nonparametric EM algorithm for semi-supervised learning. In this section we will perform a simulation study to provide numerical insight to the benefits of labeled data in the nonparametric setting. The simulation study is implemented in the following manner. First, simulate three sets of data, (X_l, Y_l) which denotes the labeled data set, X_u which denotes the unlabeled data set and a data set that we will use to test the accuracy of the classifier which we will call the test set. The sample sizes we chose for each of these three data sets are (2, 5, 10, 20, 50, 100, 200, 500, 1000) for the labeled sets, (100, 500, 1000, 10000) for the unlabeled

sets and 10,000 for the test set. For this example, as in the parametric setting, we simulated data from two Normal distributions of varying Mahalanobis distances. The Mahalanobis distances that we examined were 0.5, 1, 2 and 4. For each Mahalanobis distance, we will perform a simulation for each combination of labeled and unlabeled sample size combinations and average the classification error rates over 25 simulation replicates of each combination.

Once we have simulated the three data sets for the sample sizes of the current simulation, we execute the nonparametric EM algorithm for semi-supervised learning on our simulated data sets. First it is necessary to use the labeled data set to estimate the starting cluster means for the k-means algorithm. Therefore, in this two-class simulation study the starting cluster means can be estimated by

$$\hat{\mu}_0 = \frac{\sum_{i=1}^{l}(1-y_i)\,x_i}{\sum_{i=1}^{l}(1-y_i)}$$

and

$$\hat{\mu}_1 = \frac{\sum_{i=1}^{l} y_i\ x_i}{\sum_{i=1}^{l} y_i}\,.$$

After the cluster means are calculated, we use seeded k-means to estimate the initial P^0 matrix and then use the nonparametric EM algorithm to estimate each class density function. The next step after estimating each class density is to perform classification. To do this we use a standard Bayes classifier on the test set to determine the predicted class of each observation in the test set. The setup for the Bayes classifier in the semi-supervised nonparametric setting is similar to the standard Bayes classifier for the

parametric case. First, from the nonparametric EM algorithm we have the estimated NDEs for each class, $\hat{f}_0(x)$ and $\hat{f}_1(x)$. Also, the EM algorithm yields the estimated mixing proportions, $\hat{\lambda}_0$ and $\hat{\lambda}_1$. Therefore, we will classify an observation according to the following

$$\hat{C}(x) = \begin{cases} 0 & \text{if} \quad \dfrac{\hat{\lambda}_0 \hat{f}_0(x)}{\hat{\lambda}_0 \hat{f}_0(x) + \hat{\lambda}_1 \hat{f}_1(x)} > 0.5 \\ 1 & \text{if} \quad \dfrac{\hat{\lambda}_0 \hat{f}_0(x)}{\hat{\lambda}_0 \hat{f}_0(x) + \hat{\lambda}_1 \hat{f}_1(x)} < 0.5 \end{cases}$$

Then, using the Bayes classifier with our estimated densities, the average error rate is calculated for the 10,000 simulation outcomes.

As we can see from the tables below adding more labeled data does significantly improve the classification results in most cases. Let us first look at the case when the Mahalanobis distance is 0.5, or when the two groups are closest together. We see that for each unlabeled sample size that we can improve the classification results of as we add more labeled data. However, once we reach 50 labeled observations there is a leveling off of the benefit of adding more labeled data. Therefore if the data you are working with has a Mahalanobis distance of 0.5 then you would get the most benefit by adding 50 labeled observations to your unlabeled data. This is also illustrated in the plot for the error rates when the Mahalanobis distance is 0.5. Next, for the case when the Mahalanobis distance is 1, we again see improvement in the classification results for each unlabeled data sample size as we add more labeled observations to the semi-supervised learning procedure. In this case, however, we appear to no longer see a significant

improvement in the results once we reach 20 labeled observations. Therefore when the data has a Mahalanobis distance of 1 the greatest benefit can be reached by adding 20 labeled observations to the semi-supervised learning process. This result can also be seen in the plot of the error rates when the Mahalanobis distance is 1. For the case when the Mahalanobis distance between the two groups is 2, there is once again a benefit of adding labeled data. In this case, however, the benefit appears to stop having a significant effect on the error rate after adding 5 labeled observations. Therefore is your data has a Mahalanobis distance of 2 then the most cost effective approach for improving the classification error rate is to add 5 labeled observations. This is illustrated as well in the plot below for the case of a Mahalanobis distance of 2. In our last case, when the Mahalanobis distance is 4 and therefore has the greatest separation of the two groups, we do not see a significant improvement after adding 2 labeled observations. This means that as long as we have one labeled observation from each group we will gain that same benefit as if we had 1,000 labeled observations. This is because the groups are so well separated the probability of having an error in the labeled observations is negligible. As in the other cases, the plot below for when the Mahalanobis distance is 4 clearly illustrates these findings.

In summary, for all the Mahalanobis distances, we can improve the error rates of classification by adding labeled data. Depending on the degree of separation of the two groups, or the size of the Mahalanobis distance, the most cost effective labeled sample size varies. For instance, the labeled sample size where we stop seeing a significant improvement in the classification results, assuming there is at least one labeled

observation in each group, is 50 when the Mahalanobis distance is .5, 20 when the Mahalanobis distance is 1, 5 when the Mahalanobis distance is 2 and 2 when the Mahalanobis distance is 4. Therefore as the groups are better separated the need probability of a incorrect labeled observation decreases which reduces the number of labeled observations that are necessary for improving the classification results in the semi-supervised nonparametric learning setting.

Error Rates (Mahalanobis Distance = 0.5)				
	Unlabeled			
Labeled	**100**	**500**	**1000**	**10000**
2	0.5853	0.5687	0.5075	0.4823
5	0.5614	0.5460	0.5059	0.4642
10	0.5590	0.5278	0.4768	0.4470
20	0.5315	0.5018	0.4638	0.4335
50	0.5105	0.4881	0.4456	0.4142
100	0.5023	0.4766	0.4414	0.4103
200	0.4891	0.4629	0.4396	0.4043
500	0.4749	0.4383	0.4294	0.4042
1000	0.4344	0.4259	0.4196	0.4042
	margin of error = .01			

Table 3.12. Nonparametric semi-supervised classification results Mahalanobis distance = 0.5.

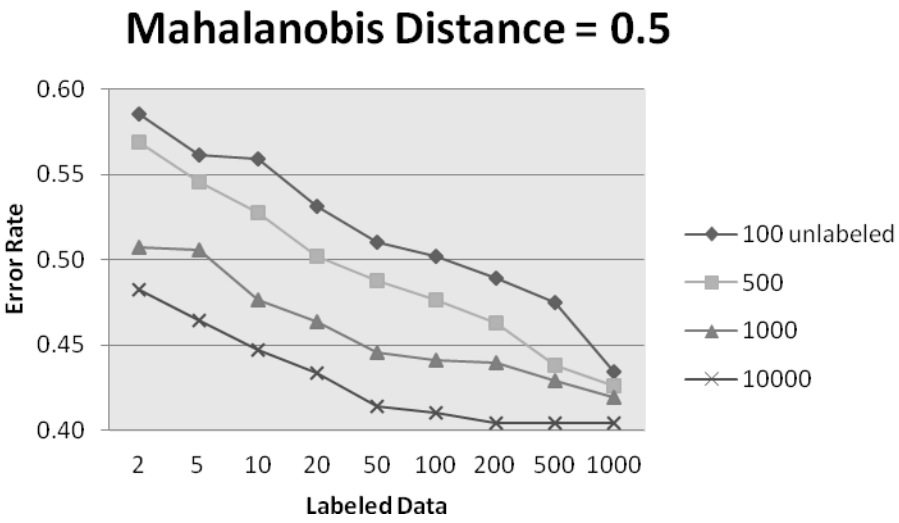

Figure 3.6. Nonparametric semi-supervised classification results Mahalanobis distance = 0.5.

Error Rates (Mahalanobis Distance = 1)				
	Unlabeled			
Labeled	**100**	**500**	**1000**	**10000**
2	0.4972	0.4430	0.4345	0.4029
5	0.4722	0.4300	0.3938	0.3893
10	0.4586	0.3848	0.3759	0.3404
20	0.4261	0.3723	0.3287	0.3101
50	0.4135	0.3530	0.3302	0.3104
100	0.3939	0.3504	0.3290	0.3103
200	0.3643	0.3422	0.3261	0.3098
500	0.3482	0.3277	0.3226	0.3104
1000	0.3291	0.3232	0.3195	0.3099
	margin of error = .01			

Table 3.13. Nonparametric semi-supervised classification results Mahalanobis distance = 1.

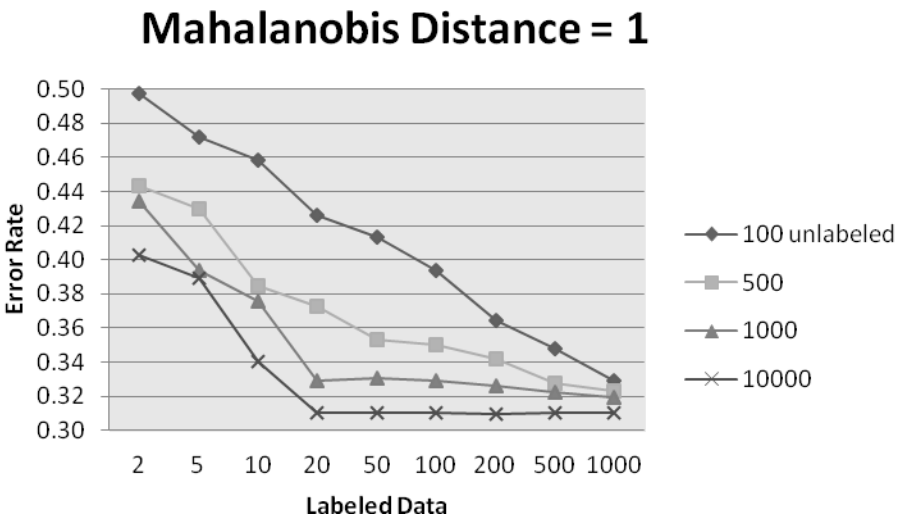

Figure 3.7. Nonparametric semi-supervised classification results Mahalanobis distance = 1.

Error Rates (Mahalanobis Distance = 2)				
	Unlabeled			
Labeled	**100**	**500**	**1000**	**10000**
2	0.2751	0.2513	0.2158	0.1873
5	0.2565	0.2021	0.1892	0.1735
10	0.2476	0.1815	0.1703	0.1665
20	0.2387	0.1846	0.1709	0.1593
50	0.2255	0.1794	0.1698	0.1596
100	0.2115	0.1804	0.1695	0.1594
200	0.2096	0.1770	0.1666	0.1597
500	0.1790	0.1699	0.1658	0.1590
1000	0.1671	0.1651	0.1645	0.1591
	margin of error = .01			

Table 3.14. Nonparametric semi-supervised classification results Mahalanobis distance = 2.

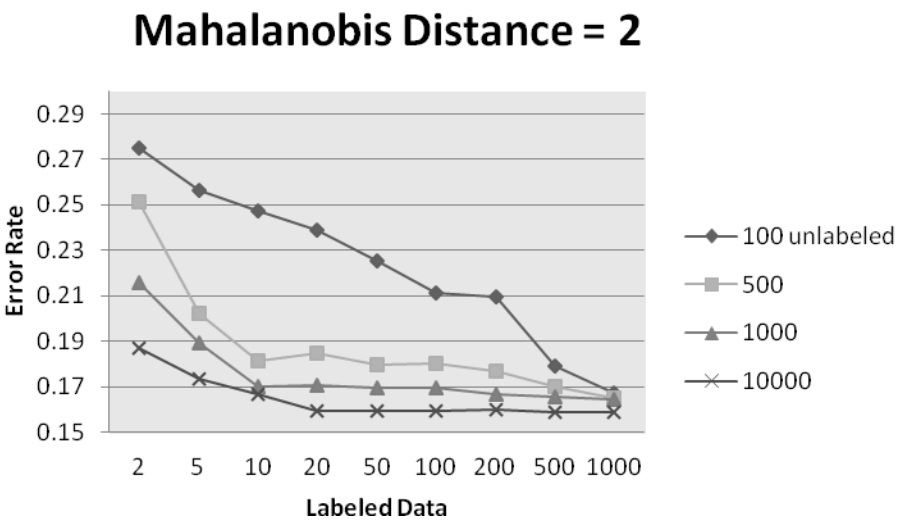

Figure 3.8. Nonparametric semi-supervised classification results Mahalanobis distance = 2.

Error Rates (Mahalanobis Distance = 4)				
	Unlabeled			
Labeled	**100**	**500**	**1000**	**10000**
2	0.0460	0.0343	0.0275	0.0277
5	0.0446	0.0330	0.0283	0.0232
10	0.0454	0.0317	0.0277	0.0229
20	0.0425	0.0323	0.0280	0.0231
50	0.0394	0.0327	0.0273	0.0229
100	0.0384	0.0321	0.0267	0.0230
200	0.0363	0.0300	0.0267	0.0231
500	0.0316	0.0274	0.0263	0.0233
1000	0.0265	0.0259	0.0245	0.0232
	margin of error = .01			

Table 3.15. Nonparametric semi-supervised classification results Mahalanobis distance = 4.

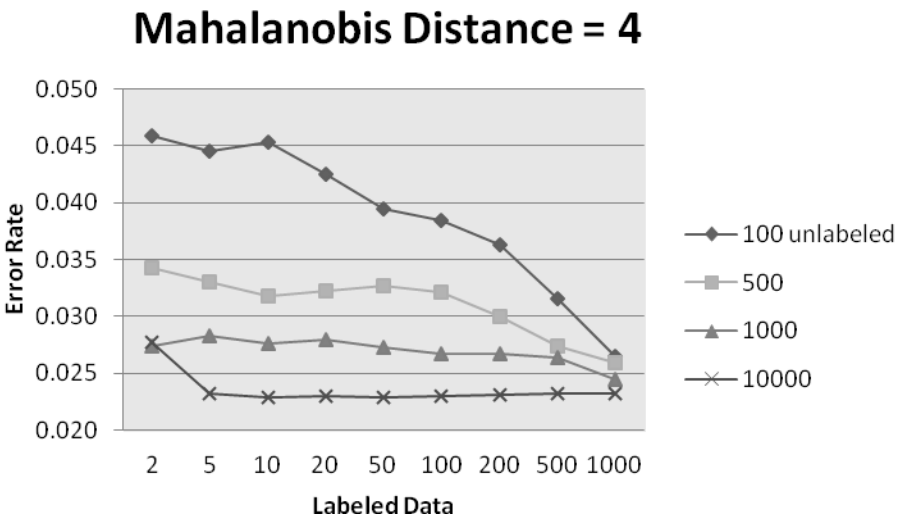

Figure 3.9. Nonparametric semi-supervised classification results Mahalanobis distance = 4.

3.5. Parametric vs. Nonparametric Comparison

We have discussed in detail the methodology for implementing semi-supervised learning via the EM algorithm for the purpose of extending the use of neutral zone classifiers beyond the supervised setting. Our discussion covered both the parametric and nonparametric cases. The EM algorithm for the parametric semi-supervised case is well established, however in the nonparametric semi-supervised case we detailed a procedure drawing on the semi-supervised k-means algorithm (Basu et al., 2002) and the nonparametric EM algorithm (Benaglia et al., 2009a). The results are fairly similar with the parametric case having slightly better performance when the data is simulated from two Normal distributions, however as both the labeled and unlabeled sample sizes increase the difference in classification accuracy becomes negligible.

The most notable difference in performance is in the nonparametric case when there is only 100 unlabeled observations. In this situation the classification error rate is much higher than in any of the other situations. This is because the nonparametric EM algorithm requires more observations to accurately estimate each class density. When the sample size increases to 500 unlabeled observations the performance of the nonparametric classification aligns more closely the parametric classification results. All of these results show us that when we are dealing with data from an unknown distribution that the nonparametric EM procedure for semi-supervised learning outlined in this chapter provides an effective method for class density estimation that will yield useful classification results when using the Bayes classifier.

Chapter 4

Neutral Zone Classification Clustering Effectiveness

4.1. Introduction

An OFRG fingerprint vector is a representation of the binding between a polony and a set of hybridization probes. Both a polony and probe are sequences of DNA. Probes are very short sequences. If they bind to the polony we gain information about it through knowledge of embedded subsegments. Binding experiments measure the intensity of the binding between the polony and several probes leaving us with a vector of intensity values, or fingerprint. Obtaining accurate fingerprints, however, is often challenging for several reasons. Quantifying the intensity of the probe-polony binding is often difficult and can be subject to noise from various factors.

Once we have a vector of intensity values we want to cluster similar polonies. Ideally we would like to translate the intensity values into binary values where 0 represents no binding and 1 represents binding. Because of the noise in the intensity values it is not always easy to determine whether a polony binded to a probe. Therefore one method to do this would be to cluster directly on the intensity values using a clustering algorithm such as *k*-means. Another method is to first classify the intensity

values into groups that represent the degree of the probe-polony binding (i.e. no binding, partial binding, complete binding) and then to cluster the result of classification fingerprints. It is not entirely clear which approach is better and in this chapter we seek to study this question.

Performing three-class neutral zone classification on the intensity values gives us a fingerprint representation of a polony to multiple probes that is represented by 0, 1, 2 and N values. Here, 0, 1 and 2 represent no binding, partial binding and complete binding, respectively. It is these 0, 1, 2 and N values that we can cluster on. An algorithm that is used to cluster the fingerprint vectors that result from neutral zone classification is the greedy clique clustering algorithm developed by Figueroa et al. (2003). This algorithm seeks to resolve the *N* values in the fingerprint produced by the neutral zone classification step in order to cluster similar sequences.

The rest of this chapter is organized as follows. In Section 4.2. we provide an overview of the greedy clique clustering algorithm developed by Figueroa et al. (2003). In Section 4.3. we perform a simulation study to investigate if first performing neutral zone classification on the intensity values will lead to better cluster results than if we performed clustering only on the intensity values. In Section 4.4. we summarize the findings of our study.

4.2. Greedy Clique Clustering Algorithm

In order to cluster the fingerprints for each polony we use the greedy clique clustering algorithm developed by Figueroa et al. (2003). The algorithm uses graph theory to find suitable clusters and has been developed to work with neutral zone classification. For the three-class neutral zone classifier fingerprints are vectors consisting of 0, 1, 2 or N values which represent no binding, partial binding, complete binding and neutral, respectively. For a set of n polonies there will be a fingerprint that corresponds to each polony and the set of all fingerprints will be denoted by $F = \{ f_1, f_2, \ldots, f_n \}$. Two relationships between fingerprints that are defined for the greedy clique clustering algorithm are resolved fingerprints and compatible fingerprints. Fingerprints f_i and f_j are considered to be resolved if they do not differ in any location and contain no N values. For example if $f_i = (0,1,0,0,2)$ and $f_j = (0,1,0,0,2)$ then f_i and f_j are resolved. Additionally, fingerprints f_i and f_j are considered to be compatible if they differ only at locations with N values. For example if $f_i = (0,1,N,0,2)$ and $f_j = (0,1,0,0,2)$ then f_i and f_j are compatible. After the greedy clique clustering algorithm is run on the set of fingerprints F the result is a set of clusters $C = \{C_1, C_2, \ldots, C_m\}$ where C_i is a set of mutually compatible fingerprints.

Figure 4.1 represents a possible scenario for fingerprint relationships. In order to understand this figure we must define some terms. In Figure 4.1 the vertices represent

the fingerprints, the edges represent the relationship between compatible fingerprints and a clique is the portion of the graph where every two vertices are connected. Additionally, a maximum clique is defined as a clique that contains the largest number of vertices and a unique maximal clique is defined as a clique that has all compatible vertices.

The greedy clique clustering algorithm can be defined using the following steps:

1. Search and remove a unique maximal clique C_u from the graph; add C_u to C
2. Repeat step 1 until no more unique maximal cliques are left.
3. Search and remove a maximum clique C_m from the graph; add C_m to C.
4. Repeat step 1 and step 3 until all fingerprints are added to C.

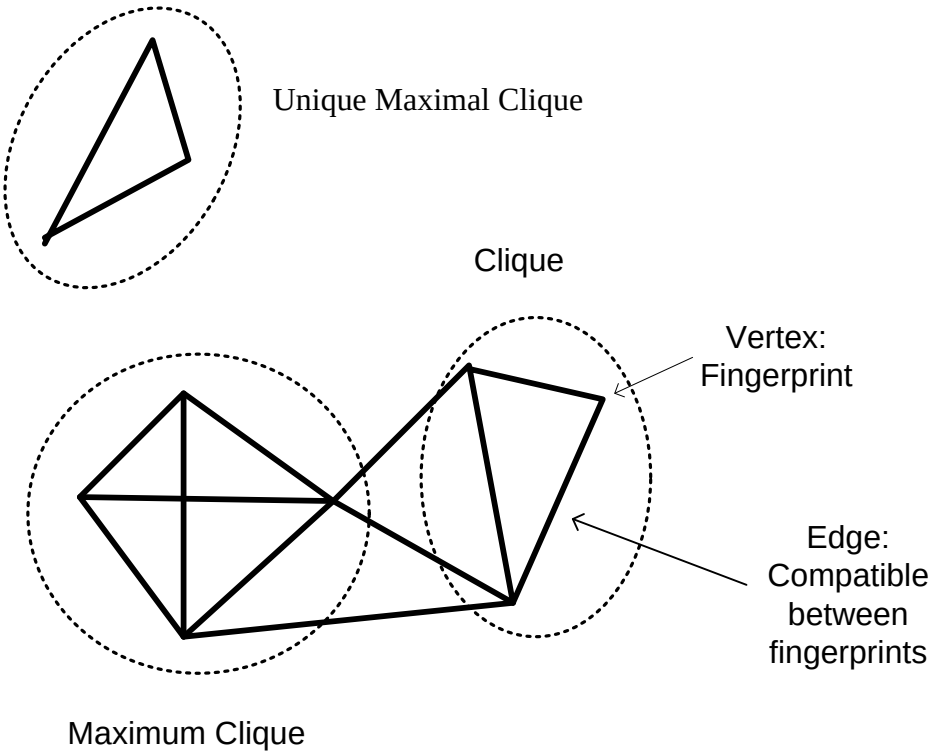

Figure 4.1. Sample clique graph.

4.3. Simulation Study

In order to determine if there is a benefit to performing neutral zone classification before the clustering step on the intensity data the following simulation was performed. First we choose the number of fingerprints, or expected clusters, as well as the fingerprint length. In this case we chose the number of fingerprints to be $c = 5$ and the fingerprint length to be $K = 5$. Therefore each of these c fingerprints of length $K = 5$ corresponds to a true cluster. We then randomly generate the $c = 5$ fingerprints where

$C_i = (C_{i1}, C_{i2}, \ldots, C_{iK})$ denotes each randomly generated sequence and $C_{ij} \in \{0,1,2\}$. An example of one of the randomly generated fingerprints is as follows:

i	C_{i1}	C_{i2}	C_{i3}	C_{i4}	C_{i5}
1	0	0	2	0	1
2	0	2	1	1	2
3	0	2	0	1	2
4	1	0	2	0	1
5	0	2	1	1	1

Table 4.1. Simulation study fingerprint example.

Then for each C_i randomly generate n_i vectors of intensity measurements $\{X_{ij}\}_{j=1}^{n_i}$, where $X_{ij} = (X_{ij1}, X_{ij2}, \ldots, X_{ijK})$ and $X_{ijl} \sim N(\mu_{C_{il}}, \sigma^2)$ for $l = 1,2,\ldots,K$. Since $C_{ij} \in \{0,1,2\}$ we need to select values for μ_i where $i \in \{0,1,2\}$ and σ^2. For this particular simulation we select $\mu_0 = 1, \mu_1 = 3, \mu_2 = 5$ and $\sigma^2 = 0.5$. Also, n_i, the number of intensity vectors we simulate from each fingerprint, is chosen to be $n_i = (5,5,5,5,2)$.

Once the intensity measurements are randomly generated, Bayesian neutral zone classification is performed in order to obtain the predicted fingerprint, $\{\hat{C}_{ij}\}_{i=1,j=1}^{c,n_i}$, for each intensity vector, $\{X_{ij}\}_{i=1,j=1}^{c,n_i}$. The predicted fingerprints of each intensity vector are then run through the greedy clique clustering algorithm to obtain predicted clusters. We then evaluate the accuracy of the predicted clusters by comparing the results to the known, simulated clusters for each vector in the following way. For each grouping size (i.e. pairs, triples, quads, etc.), we count the number of incorrect groupings. Take each pair of

intensity vectors, for example $\left(X_{12,}X_{13}\right)$, and compare whether the predicted clusters agree with the true clusters in terms of whether the pair should belong to the same cluster. In the case of the pair $\left(X_{12,}X_{13}\right)$ both vectors are from cluster 1 so we would expect them to be grouped together, whereas the pair $\left(X_{12,}X_{33}\right)$ one vector is from cluster 1 and one vector is from cluster 3 therefore we would expect them to not be grouped together. For the simulated example we have 22 choose 2, or 231, total cluster pairs and we evaluate the accuracy of the classifier by how many incorrect pairings are made. We perform the same process for all grouping sizes (i.e. triples, quads, etc.).

To establish whether performing Bayesian neutral zone classification does in fact improve clustering results, comparison to clustering on the intensity vectors is performed through the *k*-means algorithm. For Bayesian neutral zone classification we use the parametric setting and assume known densities. The cost structure used is given in Table 4.2.

True Class Label	Predicted Class Label			
	0	1	2	N
0	0	6	8	1
1	6	0	6	1
2	8	6	0	1

Table 4.2. Cost structure for neutral zone study.

For the *k*-means clustering step, each intensity vector $\left\{X_{ij}\right\}_{i=1,\,j=1}^{c,n_i}$ is clustered directly on the simulated values without first performing any classification methods. In order to not predetermine the number of clusters, k, in the *k*-means algorithm a search for the optimal

value of k is performed using average silhouette width (Kaufman and Rousseeuw, 2005). The procedure is performed by executing the *k*-means algorithm over a range of k and choosing the k value which gives the smallest average silhouette width. Silhouette width is defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

where $a(i)$ is the average dissimilarity of i to all other objects in the cluster containing i and $b(i)$ is the smallest average dissimilarity of i to any cluster created by the *k*-means algorithm. The results of the *k*-means clustering algorithm are then compared for accuracy in the same manner as the clique clustering results, by comparing the predicted cluster of each X_{ij} grouping size.

The results of the simulation are based on 50 trials of the experiment where each trial consists first of randomly generating $c = 5$ new fingerprints of length $K = 5$ before simulating the vectors of intensity measurements, $\{X_{ij}\}_{j=1}^{n_i}$. The values of each μ_i where $i \in \{0,1,2\}$ and σ^2 remain unchanged for each trial. For each group size the clique clustering on the Bayesian neutral zone classified values outperforms the *k*-means clustering results. For example, in a group size of two, clique averages 8.5 mistakes over the 50 simulations while *k*-means averages 14.6 mistakes. In a group size of three, clique averages 14.12 mistakes over the 50 simulations while *k*-means averages 65.52 mistakes. The complete results of each trial are in the following table:

Grouping Size	Total Groupings	Mean Incorrect Groupings	
		Clique	*K*-Means
2	231	8.50	14.60
3	1,540	14.12	65.52
4	7,315	12.12	175.88
5	26,334	6.34	359.82
6	74,613	2.60	608.20
7	170,544	0.92	863.08
8	319,770	0.20	1020.10
9	497,420	0.02	991.80
10	646,646	0.00	783.38
11	705,432	0.00	496.00
12	646,646	0.00	247.60
13	497,420	0.00	95.20
14	319,770	0.00	27.20
15	170,544	0.00	5.44
16	74,613	0.00	0.68
17	26,334	0.00	0.04

Table 4.3. Number of incorrect intensity vector pairings for the clique and *k*-means clustering algorithms.

In addition to the simulation study described above we varied σ^2 while keeping $\mu_0 = 1, \mu_1 = 3, \mu_2 = 5$ and the cost structure the same as in Table 4.2 in order to see how additional variance in data will affect the clustering results. We used five choices for $\sigma^2 = \{0.2, 0.3, 0.4, 0.5, 0.6\}$ and ran the previously described simulation for 10 trials at each σ^2 value. The incorrect number of pairs and triples are shown in Table 4.4. As we can see from the table that as the variability is introduced into the data the number of incorrect pairs and triples increases.

Std. Dev.	Pairs		Triples	
	Clique	K-Means	Clique	K-Means
0.2	0	2.5	0	10
0.3	0	7	0	25
0.4	8	12.4	9.6	48.4
0.5	8.5	14.6	12.1	65.5
0.6	14	18.3	24.3	82.4

Table 4.4. Number of incorrect groupings for various σ^2 values.

4.4. Summary

We demonstrated that clustering performance of intensity values can be improved by first applying Bayesian neutral zone classification rather than only clustering the intensity values directly. For the example presented in Section 4.3. we demonstrated the using the greedy clique clustering algorithm is substantially better than performing clustering using *k*-means on the intensity values. While the simulation does not provide evidence that this will always be the case, it does give encouraging results. Currently, Bayesian neutral zone classification and greedy clique clustering algorithm are being used in microbial community profiling applications. In these applications the size of each cluster is typically not equal. This is the setting where using Bayesian neutral zone classification proved most effective, which provides encouraging results for the usefulness of neutral zone classification in microbial community profiling applications.

Chapter 5

Summary

5.1. Summary

Neutral zone classifiers allow for a region of neutrality when the data is too ambiguous to confidently assign a predicted class. Previous versions of neutral zone classifiers have involved computationally complex methods for finding the boundaries for classification that minimize expected cost. In Chapter 2, we developed a neutral zone classifier from a Bayes point of view that significantly reduces the computation time for classification while at the same time reducing the expected cost of the classifier. Also in Chapter 2, we extended the original application of neutral zone classifiers to cover the paradigm of unsupervised classification. To do this we, we incorporated use of an EM algorithm for NDE into the development of the Bayes neutral zone classifier.

In Chapter 3 we discussed in detail the methodology for implementing semi-supervised learning via the EM algorithm for the purpose of extending the use of neutral zone classifiers in a semi-supervised setting. The discussion covered both the parametric and nonparametric cases. We demonstrated, using a simulation study, the benefits of adding labeled data to unlabeled data and performing semi-supervised learning as opposed to unsupervised learning.

Chapter 4 addresses the benefits of utilizing neutral zone classification methods to reduce the noise in a data set. This is an important issue for the microbial community profiling application. We demonstrated that under certain scenarios that there are benefits to using neutral zone classification to reduce noise.

5.2. Future Work

In Chapter 2 we introduced a method to perform neutral zone classification in the unsupervised nonparametric setting. Potential future work in this area could develop rigorous guidelines of when it is appropriate to use nonparametric methods in order to avoid any distribution assumptions.

For the semi-supervised work in Chapter 3 we focused the analysis mainly on a mixture of normal distributions to determine the value of labeled observations. Future work could extend this analysis to other distributions. A mixture of exponential distributions was briefly examined, however, this could be extended to many families of distributions to determine how much labeled data is needed to yield the best classification improvement.

In Chapter 4 we explored through a simulation study whether neutral zone classification improves the clustering results over performing clustering on solely the intensity values. Future work could investigate this further by performing extensive simulation studies over various distributions to determine when neutral zone classification is most effective.

Appendix

A. Proof of Theorem 1

Proof:

First we will prove for $\rho_0 > \rho_1/(\rho_1 - 1)$. Consider the case when $p_i(y) > p_j(y)$ and suppose $\rho_j \geq 2$ then we have $I_j > 0$ from **Lemma 1**. Therefore the Bayes classifier in (2.3) reduces to

$$\hat{C}_B(y) \in \begin{cases} i & \text{if} \quad I_i < 0 \\ N & \text{if} \quad I_i > 0 \end{cases}$$

which is equivalent to

$$\hat{C}_B(y) \in \begin{cases} i & \text{if} \quad p_j(y) < \dfrac{1}{\rho_i} \\ N & \text{if} \quad p_j(y) > \dfrac{1}{\rho_i} \end{cases} \tag{6.1}$$

The neutral zone classifier in **Definition 1** is defined as

$$\hat{C}_{NZ}(y; L_i) = \begin{cases} i & \text{if} \quad p_i(y) - p_j(y) > L_i \\ N & \text{if} \quad p_i(y) - p_j(y) < L_i \end{cases}$$

which, since $p_i(y) + p_j(y) = 1$, can be rewritten as

$$\hat{C}_{NZ}\left(y;L_i\right)\in\begin{cases} i & \text{if} \quad p_j\left(y\right)<\dfrac{1-L_i}{2} \\ N & \text{if} \quad p_j\left(y\right)>\dfrac{1-L_i}{2} \end{cases} \tag{6.2}$$

Therefore, for $\hat{C}_B\left(y\right)\equiv\hat{C}_{NZ}\left(y;L_i\right)$ then

$$\frac{1-L_i}{2}=\frac{1}{\rho_i} \quad \Leftrightarrow \quad L_i=1-\frac{2}{\rho_i}$$

and since the Bayes classifier is, by definition, optimal it follows that $L_i^*=1-2/\rho_i$.

Next suppose $\hat{C}_B\left(y\right)\equiv\hat{C}_{NZ}\left(y;L_i\right)$. Let us first consider the case when $p_0\left(y\right)>p_1\left(y\right)$ which implies that $p_1\left(y\right)<\frac{1}{2}$. We know from **Definition 1** our neutral zone classifier when $p_0\left(y\right)>p_1\left(y\right)$ is defined as

$$\hat{C}_{NZ}\left(y;L_0\right)\in\begin{cases} 0 & \text{if} \quad p_1\left(y\right)<\dfrac{1-L_0}{2} \\ N & \text{if} \quad p_1\left(y\right)>\dfrac{1-L_0}{2} \end{cases}$$

Let us now look at the Bayes classifier in (2.4). Since $\rho_0>\rho_1/\left(\rho_1-1\right)$ then $\frac{1}{\rho_0}<1-\frac{1}{\rho_1}$ and the Bayes classifier in (2.4) becomes

$$\hat{C}_B\left(y\right)\in\begin{cases} 0 & \text{if} \quad p_1\left(y\right)<\dfrac{1}{\rho_0} \\ 1 & \text{if} \quad p_1\left(y\right)>1-\dfrac{1}{\rho_1} \\ N & \text{if} \quad p_1\left(y\right)>\dfrac{1}{\rho_0} \text{ and } p_1\left(y\right)<1-\dfrac{1}{\rho_1} \end{cases}$$

However, $\hat{C}_B(y) \neq 1$ since $\hat{C}_B(y) \equiv \hat{C}_{NZ}(y; L_0)$. This implies that $1 - \frac{1}{\rho_1} \geq \frac{1}{2}$, or $\rho_1 \geq 2$, since $p_1(y) < \frac{1}{2}$. Also, since $\hat{C}_B(y) \equiv \hat{C}_{NZ}(y; L_0)$ we have that

$$\frac{1 - L_0}{2} = \frac{1}{\rho_0} \quad \Leftrightarrow \quad L_0 = 1 - \frac{2}{\rho_0}$$

and since the Bayes classifier is, by definition, optimal it follows that $L_0^* = 1 - 2/\rho_0$. We can show similarly for the case when $p_1(y) > p_0(y)$ that $L_1^* = 1 - 2/\rho_1$.

Next we will prove for $\rho_0 \leq \rho_1/(\rho_1 - 1)$. We have $\frac{1}{\rho_0} \geq 1 - \frac{1}{\rho_1}$ and the Bayes classifier in (2.4) becomes

$$\hat{C}(y) \in \begin{cases} 0 & \text{if} \quad p_1(y) < 1 - \frac{1}{\rho_1} \\ 0 & \text{if} \quad p_1(y) < \frac{1}{\rho_0} \text{ and } p_1(y) > 1 - \frac{1}{\rho_1} \text{ and } p_1(y) < \frac{\rho_1}{\rho_0 + \rho_1} \\ 1 & \text{if} \quad p_1(y) > \frac{1}{\rho_0} \\ 1 & \text{if} \quad p_1(y) < \frac{1}{\rho_0} \text{ and } p_1(y) > 1 - \frac{1}{\rho_1} \text{ and } p_1(y) > \frac{\rho_1}{\rho_0 + \rho_1} \end{cases}$$

Also, since

$$\frac{1}{\rho_0} \geq 1 - \frac{1}{\rho_1}$$

$$\Rightarrow \quad \rho_1 \geq \rho_0 \rho_1 - \rho_0$$

$$\Rightarrow \quad \frac{1}{\rho_0} \geq \frac{\rho_1}{\rho_0 + \rho_1}$$

and

$$\frac{1}{\rho_0} \geq 1 - \frac{1}{\rho_1}$$

$$\Rightarrow \quad \rho_1 \geq \rho_0 \rho_1 - \rho_0$$

$$\Rightarrow \quad 1 \geq \rho_0 - \frac{\rho_0}{\rho_1}$$

$$\Rightarrow \quad 1 + \rho_1 \geq \rho_0 - \frac{\rho_0}{\rho_1} + \rho_1$$

$$\Rightarrow \quad \rho_1 \geq \rho_0 + \rho_1 - \frac{\rho_1}{\rho_1} - \frac{\rho_0}{\rho_1}$$

$$\Rightarrow \quad \rho_1 \geq \rho_0 + \rho_1 - \left(\frac{\rho_0 + \rho_1}{\rho_1} \right)$$

$$\Rightarrow \quad \frac{\rho_1}{\rho_0 + \rho_1} \geq 1 - \frac{1}{\rho_1}$$

Therefore we have that $1 - \frac{1}{\rho_1} \leq \frac{\rho_1}{\rho_0 + \rho_1} \leq \frac{1}{\rho_0}$ and our Bayes classifier in (2.4) when

$\frac{1}{\rho_0} \geq 1 - \frac{1}{\rho_1}$ becomes

$$\hat{C}_B(y) \in \begin{cases} 0 & \text{if} \quad p_1(y) < \dfrac{\rho_1}{\rho_0 + \rho_1} \\ 1 & \text{if} \quad p_1(y) > \dfrac{\rho_1}{\rho_0 + \rho_1} \end{cases}$$

First assume $\rho_0 = \rho_1$ then for $\hat{C}_B(y) \equiv \hat{C}_{NZ}(y; L_0, L_1)$ we have that

$$\frac{1-L_0}{2} = \frac{1}{2} \quad \Leftrightarrow \quad L_0 = 0$$
$$\frac{1-L_1}{2} = \frac{1}{2} \quad \Leftrightarrow \quad L_1 = 0$$

and since the Bayes classifier is, by definition, optimal it follows that $\left(L_0^* = 0, L_1^* = 0\right)$.

Next assume $\hat{C}_B(y) \equiv \hat{C}_{NZ}(y; L_0, L_1)$. When $p_0(y) > p_1(y)$ then $\hat{C}_B(y) \neq 1$ which implies that $\frac{\rho_1}{\rho_0 + \rho_1} \geq \frac{1}{2}$, or $\rho_1 \geq \rho_0$, since $p_1(y) < \frac{1}{2}$. Also, when $p_1(y) > p_0(y)$ then $\hat{C}_B(y) \neq 0$ which implies that $\frac{\rho_1}{\rho_0 + \rho_1} \leq \frac{1}{2}$, or $\rho_1 \leq \rho_0$, since $p_1(y) > \frac{1}{2}$. Therefore, for $\hat{C}_B(y) \equiv \hat{C}_{NZ}(y; L_i)$ we again must have

$$\frac{1-L_0}{2} = \frac{1}{2} \quad \Leftrightarrow \quad L_0 = 0$$
$$\frac{1-L_1}{2} = \frac{1}{2} \quad \Leftrightarrow \quad L_1 = 0$$

Therefore we have $\rho_1 \geq \rho_0$ when $p_0(y) > p_1(y)$ and $\rho_1 \leq \rho_0$ when $p_1(y) > p_0(y)$ which implies that $\rho_1 = \rho_0$ for $\hat{C}_B(y) \equiv \hat{C}_{NZ}(y; L_0, L_1)$. Also, since the Bayes classifier is, by definition, optimal it follows that $L_0^* = 0$ and $L_1^* = 0$.

B. Proof of Theorem 2

Proof:

Because $p_i(y) > p_j(y) > p_k(y)$ we must have $p_j(y) < 1/2$, and thus $p_i(y) + p_k(y) > 1/2$. Hence,

$$\begin{aligned} I_j(y) &= C_N \left\{ \pi_i \rho_{ji} f(y)_i + \pi_k \rho_{jk} f_k(y) - \left(\pi_i f_i(y) + \pi_j f_j(y) + \pi_k f_k(y) \right) \right\} \\ &\geq C_N \left\{ 2\left(\pi_i f(y)_i + \pi_k f_k(y) \right) - \left(\pi_i f_i(y) + \pi_j f_j(y) + \pi_k f_k(y) \right) \right\} \\ &= \frac{C_N}{\pi_i f_i(y) + \pi_j f_j(y) + \pi_k f_k(y)} \left\{ 2\left(p_i(y) + p_k(y) \right) - 1 \right\} \\ &> 0. \end{aligned}$$

Similarly, we must have $p_i(y) + p_j(y) > 1/2$, and thus $I_k(y) > 0$. Consequently, it follows that on the branch $p_i(y) > p_j(y) > p_k(y)$, the Bayes neutral zone classifier in (2.11) simplifies to

$$\hat{C}_B(y) \in \begin{cases} i & \text{if} \quad I_i(y) < 0 \\ N & \text{if} \quad I_i(y) > 0. \end{cases}$$

Referring to Figure 2.2 each of the three regions shown represents the union of two of the six branches defined by the conditions $p_i(y) > p_j(y) > p_k(y)$. For example, the lower right region in Figure 2.2 corresponds to the union of the two branches $p_0(y) > p_1(y) > p_2(y)$ and $p_0(y) > p_2(y) > p_1(y)$. On each of these two branches, the Bayes neutral zone classifier is the same; namely $\hat{C}_B(y) \in \begin{cases} 0 & \text{if} \quad I_0(y) < 0 \\ N & \text{if} \quad I_0(y) > 0 \end{cases}$. The other two regions in Figure 2.2 are justified in a similar way.

C. Proof of Theorem 3

Proof:

If $\min[\, I_0(y)\,,\, I_1(y)\,,\, I_2(y)\,] < 0$, for all *y*, then at least one of the integrands in (2.10) is negative and (2.11) can be alternatively expressed as

$$\hat{C}_B(y) = \begin{cases} 0 & \text{if} \quad I_0(y) < \min[I_1(y)\,,\, I_2(y)\,] \\ 1 & \text{if} \quad I_1(y) < \min[I_0(y)\,,\, I_2(y)\,] \\ 2 & \text{if} \quad I_2(y) < \min[I_0(y)\,,\, I_1(y)\,] \end{cases}$$

which from (2.10) we see is equivalent to

$$\hat{C}_B(y) = \begin{cases} 0 & \text{if} \quad c < \min[a\,,\, b\,] \\ 1 & \text{if} \quad a < \min[c\,,\, d\,] \\ 2 & \text{if} \quad d < \min[e\,,\, f\,] \end{cases}$$

where

$$\begin{aligned} a &= \rho_{10}\pi_0 f_0(y) + \rho_{12}\pi_2 f_2(y) \\ b &= \rho_{20}\pi_0 f_0(y) + \rho_{21}\pi_1 f_1(y) \\ c &= \rho_{01}\pi_1 f_1(y) + \rho_{02}\pi_2 f_2(y) \\ d &= \rho_{20}\pi_0 f_0(y) + \rho_{21}\pi_1 f_1(y) \\ e &= \rho_{01}\pi_1 f_1(y) + \rho_{02}\pi_2 f_2(y) \\ f &= \rho_{10}\pi_0 f_0(y) + \rho_{12}\pi_2 f_2(y) \end{aligned}$$

Or, equivalently, we classify as $k = 0,1,2$ based on whichever of $\sum_{\substack{i=0 \\ i \neq k}}^{2} \pi_i f_i(y) C_{ki}$ is the smallest**.**

D. Proof of Theorem 4

Proof:

We will prove for case m where $p_i(y) > p_j(y) > p_k(y)$. First suppose $I_j > 0$ and $I_k > 0$ then the Bayes classifier in (2.11) reduces to

$$\hat{C}_B(y) \in \begin{cases} i & \text{if} \quad h(y) = I_i < 0 \\ N & \text{if} \quad h(y) = I_i > 0 \end{cases}.$$

The neutral zone classifier in **Definition 2** is defined as

$$\hat{C}_{NZ}(y; L_m) = \begin{cases} i & \text{if} \quad g(y) = p_i(y) - p_j(y) > L_m \\ N & \text{if} \quad g(y) = p_i(y) - p_j(y) < L_m \end{cases}.$$

Therefore, under condition 1 illustrated in Figure 5.1, for $\hat{C}_B(y) \equiv \hat{C}_{NZ}(y; L_m)$ the following must hold:

$$\begin{array}{lrcl} & \hat{C}_B(y) & = & i \\ \Leftrightarrow & h(y) & < & 0 \\ \Leftrightarrow & y & > & y^* \\ \Leftrightarrow & g(y) & > & g(y^*) \\ \Leftrightarrow & \hat{C}_{NZ}(y; L_m) & = & i \quad \text{provided we choose } L_m = g(y^*) \end{array}.$$

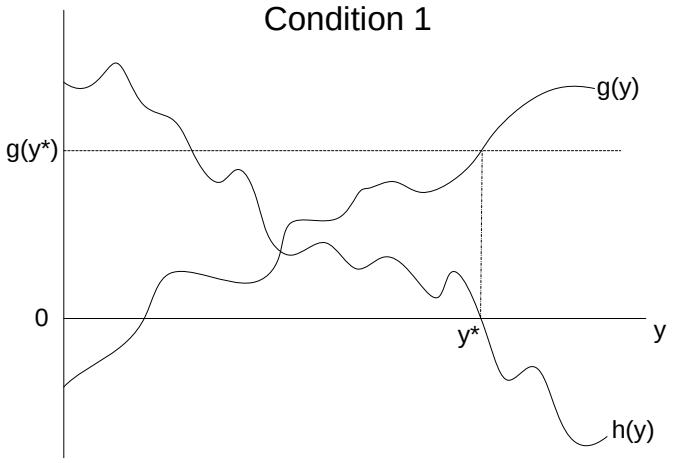

Figure 5.1. Sample plot of condition 1 for Theorem 2.

And under *Condition 2* illustrated in Figure 5.2, for $\hat{C}_B(y) \equiv$ he following must hold:

$$
\begin{array}{lrcll}
 & \hat{C}_B(y) & = & i & \\
\Leftrightarrow & h(y) & < & 0 & \\
\Leftrightarrow & y & < & y^* & \\
\Leftrightarrow & g(y) & > & g(y^*) & \\
\Leftrightarrow & \hat{C}_{NZ}(y) & = & i & \text{provided we choose } L_m = g(y^*)
\end{array}
$$

.

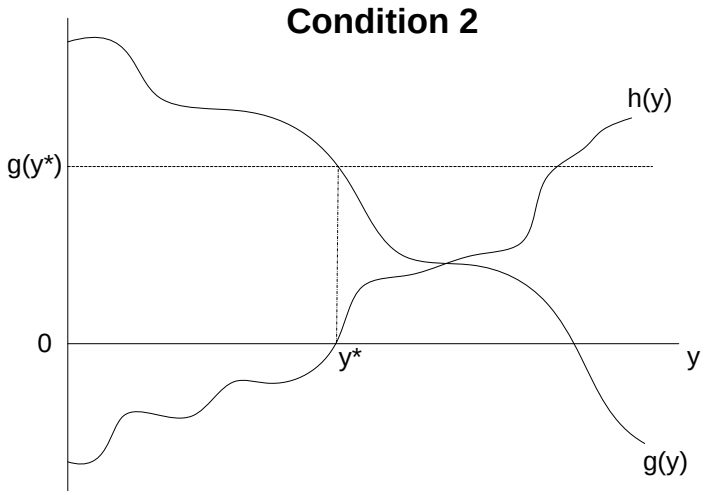

Figure 5.2. Sample plot of condition 2 for Theorem 2.

Therefore, since the Bayes classifier is, by definition, optimal it follows that $L_m^* = g\left(y^*\right)$

Next suppose $\hat{C}_B\left(y\right) \equiv \hat{C}_{NZ}\left(y; L_m\right)$ then $\hat{C}_B\left(y\right) \neq j$ and $\hat{C}_B\left(y\right) \neq k$ which implies from (2.11) that $I_j > 0$ nd $I_k > 0$ Also since, $\hat{C}_B\left(y\right) \equiv \hat{C}_{NZ}\left(y; L_m\right)$ we again have that $L_m = g\left(y^*\right)$ and since the Bayes classifier is, by definition, optimal it follows that $L_m^* = g\left(y^*\right)$.

Bibliography

Aach J, Church GM. J. (2004). Mathematical models of diffusion-constrained polymerase chain reactions: basis of high-throughput nucleic acid assays and simple self-organizing systems. *Theor Biol*, 228 (1), 31-46.

Basu, S., Banerjee, A., Mooney, R. (2002). Semi-supervised Clustering by Seeding. In Proceeding of 19th International Conference on Machine Learning.

Benaglia, T., Chauveau, D., Hunter, D. R., (2009a). An EM-like algorithm for semi- and non-parametric estimation in multivariate mixtures. *Journal of Computational and Graphical Statisitics* 18 (2), 505-526.

Benaglia T., Chauveau D., Hunter D.R. (2009b). Bandwidth Selection in an EM-like Algorithm for Nonparametric Multivariate Mixtures, in DR Hunter, DSP Richards, and JL Rosenberger (eds.), *Nonparametrics Statistics and Mixture Models: World Scientific, Singapore*. 15-27.

Box, G.E.P., Cox, D.R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society* 26:211-252.

Castelli, V., Cover, T. (1995). On the exponential value of labeled samples. *Pattern Recognition Letters* 16, 105-111.

Day, N. E. (1969). Estimating the components of a mixture of normal distributions. *Biometrika* 56, 463-474.

Figueroa, A, Borneman, J, Jiang T, Clustering binary fingerprint vectors with missing values for DNA array data analysis. IEEE Computer Society Bioinformatics Conference, 2003.

Givens, G.H., Hoeting, J.A. (2005). Computational Statistics. Wiley-Interscience.

Hastie, T., Tibshirani, R., Friedman, J. (2001). The Elements of Statistical Learning. Springer, New York, NY, USA.

Hosmer Jr., D.W. (1973). A Comparison of Iterative Maximum Likelihood Estimates of the Parameters of a Mixture of Two Normal Distributions Under Three Different Types of Sample. *Biometrics*, Vol. 29, No. 4, pp. 761-770.

Hosmer Jr., D.W., Dick, N.P. (1977). Information and mixtures of two normal distributions. *Journal of Statistical Computation and Simulation*, 6: 2, 137-148

Jeske, D. R., Liu, Z., Bent, E., and Borneman, J. (2007). Classification Rules that Include Neutral Zones and their Application to Microbial Community Profiling, *Communication in Statistics – Theory and Methods*, 36 (10), 1965-1980.

Johnson, R.A., Wichern, D.W. (2007). Applied Multivariate Statistical Analysis. Prentice Hall, Upple Saddle River, NJ, USA.

Kaufman, L., Rousseeuw, P.J. (2005). Finding Groups in Data: An Introduction to Cluster Analysis. Wiley-Interscience, Hoboken, NJ, USA.

Lo, Y., Mendell, N. R. and Rubin, D. B. (2001). Testing the number of components in a normal mixture. *Biometrika* 88, 767-778.

Louis, T. A. (1982). Finding the Observed Information Matrix when Using the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 44, No. 2. pp. 226-233.

McLachlan, G. J. and Basford, K. E. (1988). Mixture Models: Inference and Applications to Clustering. Marcel Dekker.

McLachlan, G.J., Krishnan, T., (1997). The EM Algorithm and Extensions. Wiley, New York.

Mitra RD, Church GM. (1999) In situ localized amplification and contact replication of many individual DNA molecules. *Nucleic Acids Res*. 27 (24), e34.

Schork, N. J. and Schork, M. A. (1988). Skewness and mixtures of normal distributions. *Communications in Statistics, Series A* 17, 3951-3969.

Silverman, B. W., (1998). Density Estimation for Statistics and Data Analysis. Chapman & Hall/CRC.

Valinsky, L., Vedova, G. D., Jiang, T., and Borneman, J., (2002). Oligonucleotide fingerprinting of rRNA genes for analysis of fungal community composition. *Applied and Environmental Microbiology* 68 (12): 5999-6004.

Valinsky, L., Vedova, G.D., Scumpham, A.J., Alvey, S., Figueroa, A., Yin, B., Hartin, J., Chrobak, M., Crowley, D.E., Jiang, T., and Borneman, J. (2002). Analysis of bacterial community composition by oligonucleotide fingerprinting of rRNA genes. *Applied and Environmental Microbiology* 68 (7): 3243-3250.

Yakowitz, S.J. (1970). Unsupervised Learning and the Identification of Finite Mixtures. *Transactions on Information Theory* 5: 330-339.

Yu, H., Jeske, D. R., Ruegger, P., and Borneman, J. (2010). A Three-Class Neutral Zone Classifier Using a Decision-Theoretic Approach with Application to DNA Array Analyses, to appear in *Journal of Agricultural*, *Biological and Environmental Statistics*, 15, 474-490.