

2008

An Introduction to Ambiguity and Instability: New Merit Criteria for Evaluating Classification Performance

Wei Liang

College of William & Mary - Arts & Sciences

Follow this and additional works at: <https://scholarworks.wm.edu/etd>



Part of the [Library and Information Science Commons](#)

Recommended Citation

Liang, Wei, "An Introduction to Ambiguity and Instability: New Merit Criteria for Evaluating Classification Performance" (2008). *Dissertations, Theses, and Masters Projects*. William & Mary. Paper 1539626865. <https://dx.doi.org/doi:10.21220/s2-7mg8-3v46>

This Thesis is brought to you for free and open access by the Theses, Dissertations, & Master Projects at W&M ScholarWorks. It has been accepted for inclusion in Dissertations, Theses, and Masters Projects by an authorized administrator of W&M ScholarWorks. For more information, please contact scholarworks@wm.edu.

**An Introduction to Ambiguity and Instability: New Merit Criteria for Evaluating
Classification Performance**

Liang Wei

Chaohu, Anhui Province, P.R.China

Bachelor of Science, Anhui University, 2004

**A Thesis presented to the Graduate Faculty
of the College of William and Mary in Candidacy for the Degree of
Master of Science**

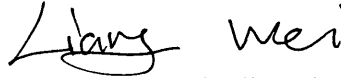
Department of Applied Science

**The College of William and Mary
January, 2008**

APPROVAL PAGE

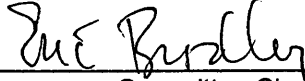
This Thesis is submitted in partial fulfillment of
the requirements for the degree of

Master of Science



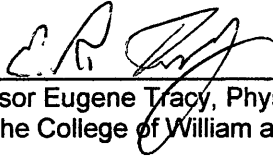
Liang Wei

Approved by the Committee, December, 2007

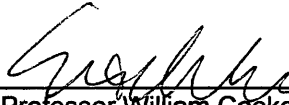


Committee Chair

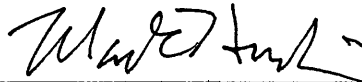
Professor Eric Bradley, Applied Science
The College of William and Mary



Chancellor Professor Eugene Tracy, Physics and Applied Science
The College of William and Mary



Professor William Cooke, Physics
The College of William and Mary



Professor Mark Hinders, Applied Science
The College of William and Mary

ABSTRACT PAGE

Different statistical classification techniques are studied on both clinical and non-clinical data sets. Traditional criteria to evaluate a specific methodology such as the misclassification error rate are discussed. New performance measures: the classification ambiguity and classification instability are introduced to evaluate the risk and the consistency of an ensemble of classifiers generated by cross-validation. Numerical experiments indicate that the new definitions are useful for understanding: 1] how the classification methodology will behave for future data and 2] the similarity and differences among various algorithms on data sets with different structures.

Contents

1	Introduction and background	1
1.1	Classification: general comments and an example	1
1.1.1	Example: detection of early cancer using TOF-MS, introduction and motivation	3
1.2	Summary of classification algorithms considered in this thesis	6
1.2.1	Linear and Quadratic Discriminant Analysis (LDA, QDA)	7
1.2.2	Decision tree: recursive partitioning (rpart)	8
1.2.3	K-Nearest Neighbors (KNN)	13
1.3	Classifier evaluation	14
1.3.1	When the training set and the test set are the same	15
1.3.2	When the data set is divided into two disjoint subsets	15
1.3.3	V-fold cross-validation	16
1.4	Decision boundaries	16
2	The Ambiguity and the Instability	22
2.1	The ambiguity	22
2.2	The instability	24
2.3	The Error Rate	26
2.4	Relationships of the error rate, ambiguity and instability	30
3	Numerical experiments	36
3.1	Introduction to the three data sets	36
3.2	Results	40
4	Variable ranking and variable selection	54
4.1	Variable ranking	55
4.1.1	Example: variable ranking using rpart	55
4.2	Optimum subset selection	61
4.2.1	Example: subset selection with McHenry's heuristic method	62
5	Visualization tool and VIBE-MS	64
5.1	Visualization of the decision boundaries	64
5.1.1	Visualization algorithm	64
5.2	VIBE-MS	68
5.2.1	Classification tools in VIBE-MS	69

6	Conclusions	72
7	Appendix: R code	74
7.1	R code to plot the decision boundaries	74
7.2	R code to plot the heatmaps of the ambiguous and unstable regions .	77
7.3	R code to compute the global ambiguity, instability and error rate . .	83
8	References	90

Dedication

To Mom and Dad for all of their unending support.

Acknowledgments

I would like to thank Professor Michael Trosset for introducing me to the interesting field of statistical data analysis.

Special thanks to Professor Eugene Tracy for his many invaluable suggestions after Professor Trosset left the College of William and Mary, and for reading, correcting and guiding me on my thesis.

Thanks to Professor William Cooke who aroused my interest in early cancer detection through his creative ideas.

Thanks to Professor Eric Bradley who gave me a lot of help in both my study and my work. Also thanks to Professor Mark Hinders who reviewed my thesis.

Many thanks to the Applied Science department in the College of William and Mary and Dr. John Semmes's proteomics research group in Eastern Virginia Medical School who funded my graduate research. Also many thanks to Incogen, Inc. for providing me a chance to integrate some of my R classification codes into their powerful bioinformatics environment: VIBE-MS which is used to solve several real-world problems.

List of Figures

1.1	An example of a mass spectrum peak list.	5
1.2	An example of decision tree using rpart algorithm	9
1.3	Three splitting criteria for rpart.	11
1.4	Decision boundaries for four classification algorithms.	18
1.5	Decision boundaries during a 5-fold cross-validation procedure using rpart algorithm.	19
1.6	Decision boundaries during a 5-fold cross-validation procedure using LDA.	20
2.1	This example shows the similarity and differences of Bayes error and the ambiguity	29
2.2	Heatmap of ambiguous regions using LDA with probabilistic method.	33
2.3	Heatmap of the ambiguous regions using LDA with deterministic method.	34
3.1	Scatterplot of <i>data.2g</i>	37
3.2	Scatterplot of <i>data.4g</i>	38
3.3	Scatterplot of <i>data.ng</i>	39
3.4	Ambiguity heatmap using probabilistic methods on <i>data.2g</i>	45
3.5	Ambiguity heatmap using deterministic methods on <i>data.2g</i>	46
3.6	Heatmap of unstable areas using probabilistic methods on <i>data.2g</i>	47
3.7	Ambiguity heatmap using deterministic methods on <i>data.4g</i>	48
3.8	Ambiguity heatmaps using probabilistic methods on <i>data.4g</i>	49
3.9	Heatmap of the unstable areas using probabilistic methods on <i>data.4g</i>	50
3.10	Ambiguity heatmap using deterministic methods on <i>data.ng</i>	51
3.11	Ambiguity heatmap using probabilistic methods on <i>data.ng</i>	52
3.12	Heatmap of unstable areas using probabilistic methods on <i>data.ng</i>	53
4.1	Decision tree plot with 2002 Leukemia data by rpart.	57
4.2	Top 10 variables ranked by rpart.	58
4.3	Scatterplot using top 2 most important variables by rpart.	60
4.4	Scatterplot using top 2 most important variables selected by Wilks Λ	63
5.1	Decision boundaries of a KNN classifier using <i>data.ng</i>	65
5.2	Decision Boundary of a KNN classifier using <i>data.2g</i>	66
5.3	VIBE Work flow.	71

List of Tables

3.1	A summary of the results in the numerical experiments.	40
3.2	Summary of the figures and the measurements they represent.	41
3.3	Comparisons of Bayes error and ambiguity.	41

Chapter 1

Introduction and background

1.1 Classification: general comments and an example

Statistical classification tries to use quantitative measurements to determine a sample's group label. For example: $D: \{\mathbf{x}_1, y_1\}, \{\mathbf{x}_2, y_2\}, \dots, \{\mathbf{x}_N, y_g\}$ is a training data set which has N samples from g groups. $\mathbf{x}_i, i=1,2,\dots,N$ are the quantitative measurements, $y_j, j = 1, 2, \dots, g$ is the group label. Each row of D represents a sample, and each column is a variable. The samples space is defined as:

Definition A sample space X is a non-empty linear Euclidean space spanned by quantitative measurements (variables) in the data. It is a q dimensional space, where q is the largest number of any set of independent variables in data D .

The classification problem can be summarized as the task of building a classifier C that maps the subjects from sample space X to a group label set $G, G = \{1,2,\dots, g\}$. A classifier is defined as:

Definition A *probabilistic classifier* or a classification rule is a function which maps an object from a q dimensional sample space X into a discrete set of group labels

G with a certain probability. A *deterministic classifier* is a special case where the probabilities are either zero or one.

Classification is one of the primary tasks in many scientific fields. Here we give some examples of common classification problems:

1. given a set of emails, building an accurate and effective statistical model to differentiate good emails from spam emails.
2. In early cancer detection research, with data having the relative abundance of different proteins, creating a classifier which can accurately predict whether an individual whether he has a cancer or not.
3. In conditions when it is difficult to identify whether the fossil belongs to a human or a chimpanzee, classification models can provide likelihood estimate for the fossil belonging to any one group by analyzing quantitative measurements such as length, width, etc.

There are many different classification approaches, introductions to *Linear Discriminant Analysis (LDA)* and *Quadratic Discriminant Analysis (QDA)* can be found in the classic textbooks by *Mardia et al.*, [1] and *Trevor et al.*, [2]; *decision tree* techniques in *Classification and Regression Tree* by Breiman et al., [3]; *K Nearest Neighbor (KNN)*, *Boosting*, *Random Forrest (RF)*, *support vector machine (SVM)* and many other modern classification methods from the machine learning community are discussed in the book *The Elements of Statistical Learning* by *Trevor et al.*, [2]. The application of these algorithms using the R programming language can be found in *THE R FAQ* [6], the textbook by *Brian Everitt: An R and S-Plus Companion to Multivariate Analysis* and the book by *Venables et al.: Modern Applied Statistics with S* [8].

1.1.1 Example: detection of early cancer using TOF-MS, introduction and motivation

According to the report *Cancer Facts and Figures 2007* [9] released by the American Cancer Society, cancer is one of the most deadly diseases in the US. The most effective way of curing a cancer is to detect it and treat it at its early stage. Many early cancer detection methods such as Pap smear, Mammography, X-ray imaging, Prostate-specific antigen (PSA) and their effectiveness for different types of cancer are discussed in [12], [17] and [19]. In this thesis, we focus on the data generated with the Matrix-assisted laser-desorption/ionization time-of-flight (MALDI-TOF) technique. The MALDI-TOF techniques together with the involved data preprocessing algorithms is discussed in [10],[11], [13], [14], [15] and [16].

Mass spectra provide the relative abundance of different proteins in body fluids or tissues such as blood samples. In our research, blood samples from people with and without a cancer are collected and mixed with solutions of assisting matrix. The samples are then put into a vacuum space, waiting to be ionized. At the beginning, a laser beam shoots the samples and the high energy transferred from the laser beam ionizes the different proteins in the complicated mixture of blood samples. They leave the surface of the sample and become ionized through a complex charge exchange process(most of the resulting ions are singly charged). They then enter an electrostatic acceleration field which gives all ions the same energy, implying they have a mass-dependent velocity. After this acceleration stage, the ions enter a field-free region, which is a long straight flight tube, and drift at constant velocity. The ions are finally stopped by striking a detector, with the intensity of the ion signal at any given time being proportional to the number of ions at the related mass. The time of flight (TOF) is measured, and from it the mass can be inferred using Equation (1.1), where V is the energy of the ion (the charge multiplied by the acceleration potential), t is its total flight time in the tube, l is the length of the

flight tube, m is the mass of the ion, and z is the ion's electric charge. The square of the flight time for an ion is proportional to its mass over z ratio. Because most of the ions are one single positively charged, their flight time becomes proportional to the root square of the mass. We can resolve the mass of an ion using its flight time from Equation 1.1.

$$\frac{2Vt^2}{l^2} = \frac{m}{z} \quad (1.1)$$

The data provides information on the intensity of ions during flight time and is recorded as the ion intensity versus flight time. Figure 1.1 is an example of peak list plotted against the m/z ratio. Peak lists are obtained after the raw data is processed in several signal processing steps including background subtraction, noise reduction, baseline subtraction, etc. The detailed explanations of peak picking and several signal processing methods in mass spectrometry can be found in [10], [11] and [17].

By comparing the ion signals of samples from healthy people and people with cancer, we hope to find a subset of important proteins that have significantly different intensities using modern statistical classification technique. Based on the assumption that the ion intensity is relative to the protein abundance, the important ion peaks will help us to find out a set of proteins as potential biomarkers for early cancer detection.

The data generally is partitioned into two disjoint subsets: training and test set. The training set is used to build the classification model (a classifier) and test set helps to evaluate the model using criteria such as specificity (which represents the classifier's ability to correctly classify a healthy people with a healthy group label), and sensitivity (which measures the classifier's ability to correctly classify patients into patient groups).

Each sample has the prevalence of different proteins in a blood or tissue sample.

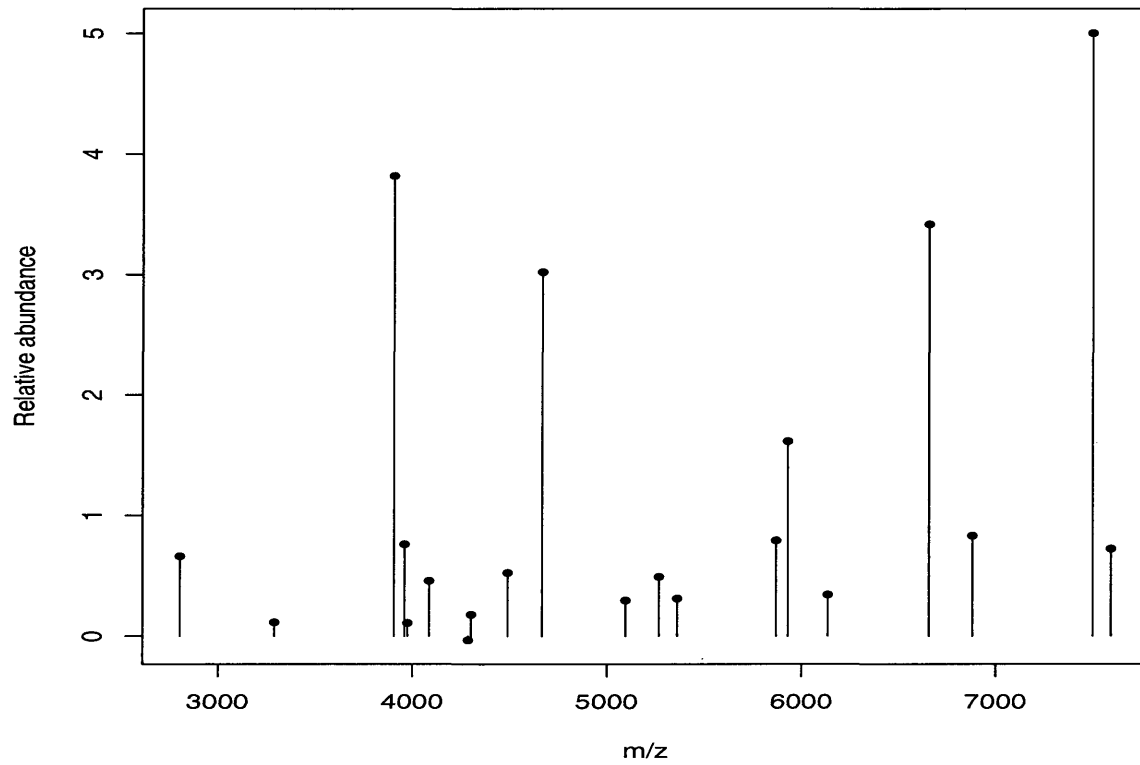


Figure 1.1: This is an example of a mass spectrum peak list, the vertical axis are relative intensity of a peak, the horizontal axis is the over charge ratio of a peak.

A sample has both the quantitative measurements x_i (the mass peaks intensities) and the group label y_i (usually provided by a pathologist using traditional diagnostic method). The quantitative measurements x_i is a vector from a p (p is the number of peaks) dimensional sample space spanned by the basis of the set of variables, y_i is from the group labels set G . Because of the so called *curse of dimensionality* problem, usually when the number of variables is large, not only is the computation consumption extraordinarily increased, but also new problems get introduced. For example: when the number of variables is greater than the number of samples, the covariance matrix of the data will be non-invertible. Thus some algorithms requiring the inverse of the covariance matrix cannot be used. Hence, before the classification procedure, it is wise to choose a smaller set of important variables if the number of variables is very large. More important is that when the number of variables is larger than the number of samples, one can *ALWAYS* create a simple perfect classifier with zero misclassification rate. But This does not mean that future data will be correctly classified, only that you do not have enough samples.

After the dimension of the data is reduced, different classification techniques could then be used to build classification models. These models will finally get evaluated. Usually the classification technique that performs the best with a certain evaluation criterion is selected for further study.

1.2 Summary of classification algorithms considered in this thesis

In this thesis, four classification algorithms are discussed: *Linear Discriminant Analysis* (LDA), *Quadratic Discriminant Analysis* (QDA), *Recursive Partitioning* (rpart) and *K Nearest Neighbors* (KNN). We will consider both probabilistic and deterministic approaches of each algorithm.

1.2.1 Linear and Quadratic Discriminant Analysis (LDA, QDA)

LDA and QDA both require that the data are normally distributed, LDA also requires that the different groups share the same covariance structure, while QDA does not have that restriction. For a classification problem with g groups data, the posterior probability that sample x belongs to group i , $i = 1, 2, \dots, g$ is given by:

$$p(x \in \text{group } i|x) = \frac{f_i(x)\pi_i}{\sum_{i=1}^g f_i(x)\pi_i} \quad (1.2)$$

Where π_i is the prior probability for group i , f_i is the probability density function for samples from group i that is normally distributed. And

$$f_i(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_i|^{1/2}} \exp -\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1}(x - \mu_i) \quad (1.3)$$

Where Σ_i and μ_i are group i 's covariance matrix and mean, respectively.

In addition to telling which group a sample x will be assigned to, the probabilistic classifier will also tell us how likely this assignment is.

1. Probabilistic version of LDA and QDA: sample x is assigned to a group i with probability $p(x \in \text{group } i|x)$, $i = 1, 2, \dots, g$.
2. Deterministic version of LDA and QDA: the deterministic method tries to find a group j , $j=1, 2, \dots, g$ such that the posterior probability $p(x \in \text{group } j|x)$ is the largest, and assigns x into group j with probability 1, other groups with probability zero.

When $g=2$, it becomes a 2-group classification problem. The discriminant function for both LDA and QDA is:

$$\delta = \log \frac{f_1(x)}{f_2(x)} + \log \frac{\pi_1}{\pi_2} \quad (1.4)$$

When we assume that the two groups have the same covariance structures, δ becomes a linear function of x (LDA). And if we don't have this assumption, δ is a quadratic function of x (QDA). If δ is larger than zero, deterministic method assigns x into group 1 with a probability 1, otherwise assigns to the second group with a probability 1.

1.2.2 Decision tree: recursive partitioning (rpart)

There are many statistical packages based on the idea of decision tree such as *Classification and Regression Tree (CART)* (discussed in [3] and [4]), and *C4.5* in [20]. *Recursive Partitioning (rpart)* is one of the decision tree algorithms. It builds a decision tree to separate data into more and more homogeneous subsets. After a decision tree is built, it can be used to predict a future sample's group label.

Figure 1.2.2 is a decision tree based on a 4-group data set. Each sample x has two quantitative variables: $X1$ and $X2$ and one group label $g(x)$. At each split, samples satisfying the inequality such as $X1 < 5.938$ go to the left sub node, the rest to the right. At each node, rpart searches all the variables exhaustively for variable that best decrease a certain splitting criterion such as misclassification rate. The samples in that node are then separated into two smaller subsets using that variable. Three splitting criteria are available for rpart, which are:

1. *Misclassification error:*

$$I_m(t) = \frac{1}{N_m(t)} \sum_i I(x_i \neq k(t)) \quad (1.5)$$

Where $I_m(t)$ represents the misclassification rate at node t . It is defined as the proportion of points that are misclassified at node t . $N_m(t)$ is the number of samples at node t . Rpart classifies samples at a node into one group by making a vote using the sample's group labels at that node. The group which

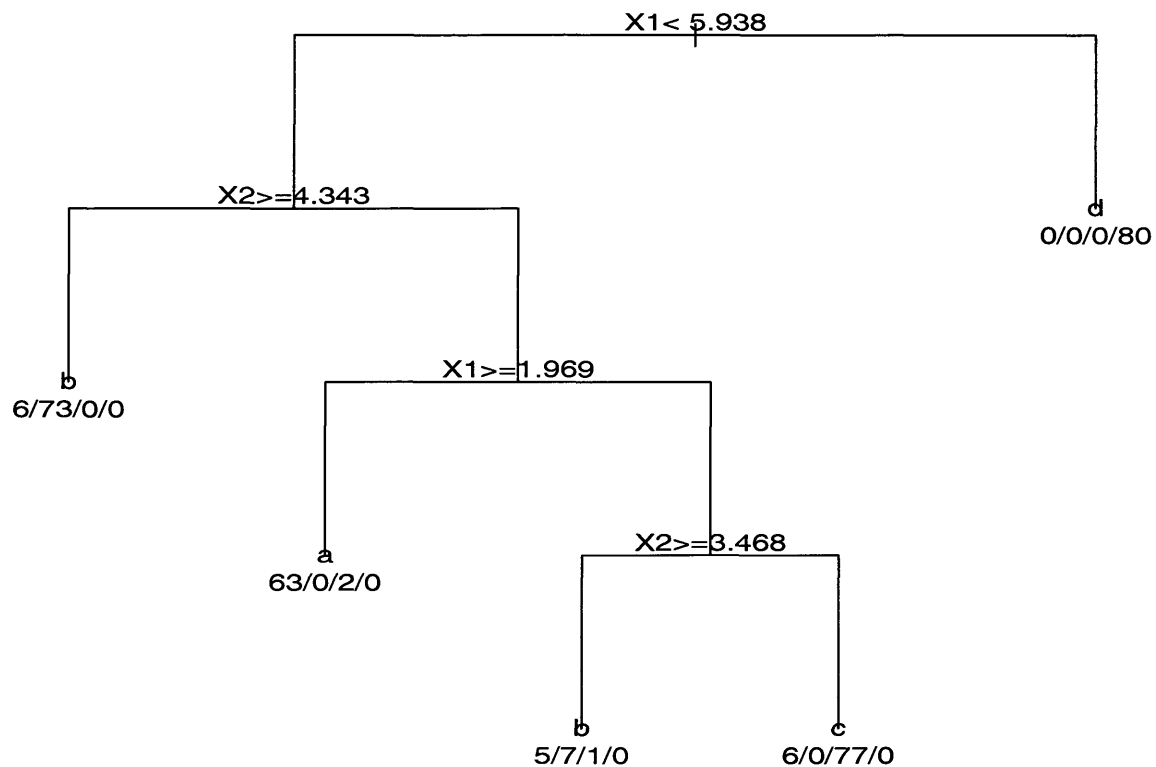


Figure 1.2: An example of decision tree using rpart algorithm. Each split in this tree diagram represents a cut that is made in the intensity value for a particular peak. For example, at the very top, the first cut is taken on the variable x_1 . If the intensity value of the peak is less than 5.938, we move to the left and perform the next cut on another variable. If it is greater than 5.938 we move to the right. Here we encounter an end of the cuts, known as a 'leaf'. The four numbers labeling each leaf show how many peaks would be classified into each of the four groups, using that particular sequence of data cuts.

has the majority of samples in node t will be the predicted group label for all samples and is represented by $k(t)$.

2. Gini index

$$I_g(t) = \sum_{k \neq k'} \hat{p}_k(t) \hat{p}_{k'}(t) \quad (1.6)$$

In the formula above, $I_g(t)$ is the Gini index at node t , $\hat{p}_{k,t}$ is the proportion of samples from group k . The Gini index reaches its maximum value of $\frac{g-1}{2g}$ when the proportions of all the groups are the same. When $g=2$, it means that at node t . In this case, the number of samples from group 1 equals the numbers of samples from group 2. It also means that the classification at node t is very poor.

3. Information entropy

$$I_e(t) = - \sum_{k=1}^K \hat{p}_k(t) \log \hat{p}_k(t) \quad (1.7)$$

Similar to the Gini index, the information entropy reaches its maximum when the proportions of all the groups are the same.

For two class problem, if $p(1,t)$ is the proportion of group 1, we have:

$$I_m(t) = 1 - \max(p(1,t), (1 - p(1,t))) \quad (1.8)$$

$$I_g(t) = 2p(1,t)(1 - p(1,t)) \quad (1.9)$$

$$I_e(t) = -p(1,t) \log(p(1,t)) - (1 - p(1,t)) \log(1 - p(1,t)). \quad (1.10)$$

Figure 1.3 shows the three splitting criteria for a 2-group classification problem.

Information Entropy, Gini index and Misclassification error

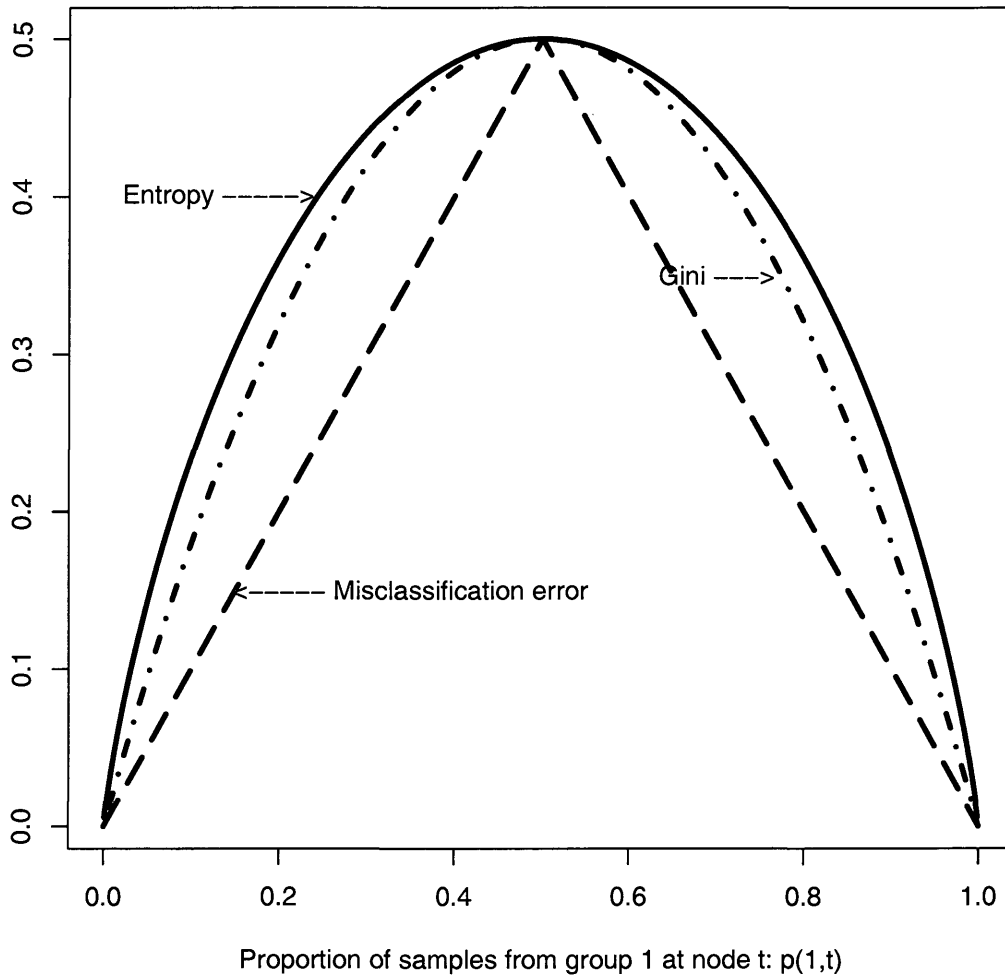


Figure 1.3: Three splitting criteria for rpart: Gini index, misclassification error and information entropy. Information entropy is scaled so that its maximum value is the same as the other two criteria. Their relationships are: information entropy \geq Gini index \geq misclassification error.

The horizontal axis is the proportion of samples from group 1 at node t : $p(1, t)$. Values from the three criteria are plotted on the vertical axis. The relationship between the three measurements can be summarized as:

$$I_m(t) \leq I_e(t) \leq I_m(t) \quad (1.11)$$

The decrease of the criterion using variable V_i at node t is defined as:

$$\Delta(I(V_i, t)) = I(t) - p(t_{left})I(V_i, t_{left}) - p(t_{right})I(V_i, t_{right}) \quad (1.12)$$

I can be any of the above splitting criteria, for example, if I is the misclassification rate, $I(t)$ will represent the misclassification rate at node t . t_{left} and t_{right} stand for the left and right subnodes after the split using variable V_i . The variable which best decreases Δ is chosen to separate the samples at node t .

By default, `rpart` uses the Gini index as its splitting criterion. The size of the tree can be determined by the following equation:

$$R_\alpha(T) = R(T) + \alpha|T| \quad (1.13)$$

Where $R(T)$ is a loss measurement such as the misclassification rate of the tree T . $|T|$ represents its size which equals to the total number of decision blocks (leaves). $|T|$ also measures the complexity of the tree. α is a nonnegative value called complexity parameter (CP). α must be fixed before running `rpart`. Equation 1.13 is an optimal problem in which our goal is to minimize $R_\alpha(T)$. For a fixed value of control parameter α , the problem becomes to determining a tree that minimizes $R_\alpha(T)$. If we choose misclassification rate as our splitting criterion, Equation 1.13 will represent a tradeoff between misclassification rate and the complexity of the tree. If α is small, we can grow a large tree T to minimize the optimal problem. When α is very small, for example, (say, it equals to zero), the optimum tree is a complete tree in

which each sample has its own decision block. The size of this complete tree is the total number of samples from the data. On the other hand, if α is very large, the optimum tree must have a small value of $|T|$.

We can also set up some other control parameters in `rpart` so that the tree growing process is stopped when:

i] The minimum number of samples in a decision block has been reached, the control parameter in `rpart` to achieve this is *minbucket*; ii] The maximum number of nodes has been created. This is controlled by parameter *maxdepth* in `rpart`.

While `rpart` is often used as a deterministic classifier, it can be used in a probabilistic fashion, too.

1. Probabilistic version of `rpart`: given a sample x , if it finally falls into leaf t , its posterior probability to be in group i is defined as:

$$P(x \in i | x \text{ in leaf } t) = \frac{\pi_i P(x \in i, x \text{ in leaf } t)}{P(x \text{ in leaf } t)} \approx \frac{\pi_i (n_{it}/n_i)}{\sum_{i=1}^g \pi_i (n_{iT}/n_i)} \quad (1.14)$$

π_i is the prior probability for samples from group i ; n_i is the total number of samples from group i in the training set; n_{it} is the number of samples from group i in leaf t .

2. Deterministic version: similarly with LDA and QDA, in equation 1.14, we find the maximum $P(x \in \text{group } i | x \text{ in leaf } t)$, assign x into group i with probability 1, probability zero to other groups.

1.2.3 K-Nearest Neighbors (KNN)

For a sample x , KNN uses a distance metric such as Euclidean distance to find K samples that are closest to x . The likelihood is calculated as the proportion of the

group labels from the K samples. We define the posterior probability for x using KNN as:

$$P(x \in \text{group } i|x) \equiv C\pi_i \frac{\text{\#of samples in the } K \text{ nearest neighbors from group } i}{K} \quad (1.15)$$

where C is a normalizer.

1. Probabilistic version: sample x is assigned into group i with probability defined in equation 1.15.
2. Deterministic version: similarly the deterministic version assigns x into group i with probability one if $P(x \in \text{group } i|x)$ is the largest, zeros to other groups. If we assume that the prior probabilities for all the groups are the same, the deterministic method equals to making a vote using the group labels in the k nearest neighborhoods.

There is no general way to find a best value for K , when $K = 1$, it is called 1-nearest neighbor classifier. K is usually selected as an odd number to avoid the cases that the vote comes to a draw, and $K=7$ or $K=5$ are widely used.

1.3 Classifier evaluation

A common criterion to evaluate a classifier's performance is the misclassification (or error) rate. After a classifier C is trained, C is applied to a test set whose group is known and the test samples are assigned group labels. The number of misclassified samples divided by the total number of samples of the test set is used to estimate the misclassification rate.

$$\text{Error rate} = \frac{\text{\#of misclassified samples}}{\text{\#of total samples in the test set}} \quad (1.16)$$

To obtain an unbiased estimate of the misclassification rate, we have to make sure that the training set and the test set are independent of each other but drawn from same populations. Generally there are three methods used to choose the training set and test set:

1.3.1 When the training set and the test set are the same

In this case, the whole data set is used to train a classifier C , before being tested against all of the samples. The error rate is estimated as the proportion of the misclassified samples, and is usually referred to as the nominal or *resubstitution* error rate. The drawback of this estimate is that the test set and the training set are the same thus not independent, and thus the estimate is biased.

1.3.2 When the data set is divided into two disjoint subsets

In this method, the data set is separated into two disjoint subsets: X_1 and X_2 at first. Only X_1 is used to train a classifier, while the samples from X_2 are used to estimate the error rate. Usually X_1 is comprised of 2/3 of randomly selected samples, the rest 1/3 becomes X_2 . This method will give us an unbiased estimate of the error rate. Although it eliminates the correlation between training and test sets, it also reduces the number of useful samples when we are training the classifier. This can sometimes be a concern, given the limited size of the sample. And because we only used 2/3 of the samples to build the classifier, this method may decrease the training set's ability in representing the whole population. Another concern about this method is that in some real problems, data are gathered expensively, in which case we may not get a large number of samples. For example, obtaining a mass spectrum sample in cancer research is very expensive, and the fact that 1/3 of the samples are not used in training tells us we should consider a method which could be more economic and use the data more efficiently.

1.3.3 V-fold cross-validation

Cross-validation is a widely used evaluation technique in estimating a classification technique's performance. The basic idea of cross-validation is to randomly partition the whole data set into several disjoint subsets, iteratively pick one of them as the test set, the rest as training set until each sample is tested exactly once. By repeating this procedure many times, we can decrease the variance from the randomly partitioning. V represents the number of subsets, it is usually selected as an integer between 5 and 10. And if the total number of samples is n , we have a limited number of different partitions for a 5-fold cross-validation:

$$N \approx \binom{n}{n/5} \binom{4n/5}{n/5} \binom{3n/5}{n/5} \binom{2n/5}{n/5} = \frac{n!}{(\frac{n}{5})^5} \quad (1.17)$$

Where n is the number of samples and thus usually a large number. Thus total number of different partitions N from the above equation will be a very large number. Let $R_1^{(k)}, R_2^{(k)}, \dots, R_V^{(k)}$ be the V different training sets at k_{th} cross-validation procedure, from equation 1.17 we know that the total number of possible training sets is a limited number N . Using the same classification technique we build an ensemble of classifiers $C^{(k)}$, $k=1,2,\dots,N$ based on the N different training sets. The posterior probability vector of a point x from the sample space given by classifier $C^{(k)}$ is: $P^{(k)}(x) = [p^{(k)}(x \in \text{group } 1|x), p^{(k)}(x \in \text{group } 2|x), \dots, p^{(k)}(x \in \text{group } g|x)]$.

1.4 Decision boundaries

Definition Decision boundaries are a set of points from a sample space X which are assigned equal probabilities for to at least two groups. The points need not to be from the data, they can be any points in the sample space.

For example: in a two class classification problem, subject x has been assigned

50% probability to group 1 and 50 % to group 2. Thus x cannot be classified into either group, as it is on the decision boundary. Decision boundaries draw special attention as they are composed of particular points in the sample space. The four classification algorithms may have quite different decision boundaries according to Figure 1.4.

Figure 1.4 shows different decision boundaries for the four algorithms using the same training set. It indicates us that some of the local decision boundaries for KNN algorithm are discrete instead of continuous. The simulated decision boundaries heatmap algorithm is summarized in algorithm 1 in section 5.

The data used in Figure 1.5 and Figure 1.6 are the same data which has four groups of samples, each group has 100 subjects, and is independently normally distributed in a 2-d Euclidean space. 5-fold cross-validation randomly choose 80% of the samples each time (320 totally, 80 from each group) and a classifier is built on the 320 samples, decision boundaries are then plotted using that classifier.

Figure 1.5 reveals the unstable nature of rpart for the specific pattern of data. We see that When 20% of the samples are replaced, the rpart decision boundaries change dramatically in a 5-fold cross-validation procedure. The discussion in the introduction section about the rpart algorithm together with Figure 1.5 clearly shows that rpart is very sensitive to the variation of samples. The default splitting criterion for rpart is the Gini index, it concerns more about how to separate the samples into more and more homogeneous subsets instead of maintaining the consistency of making a cut. Figure 1.5 also implies that sometimes there are more cuts using horizontal variable X_1 . Sometimes the vertical variable X_2 is used more than X_1 . In the future, if a sample is drawn from the highly *unstable region*, which we will give a strict mathematical definition later, it can be very difficult to assign a reliable group label to that sample using rpart, as classifiers from different training sets can be contradicting with each other.

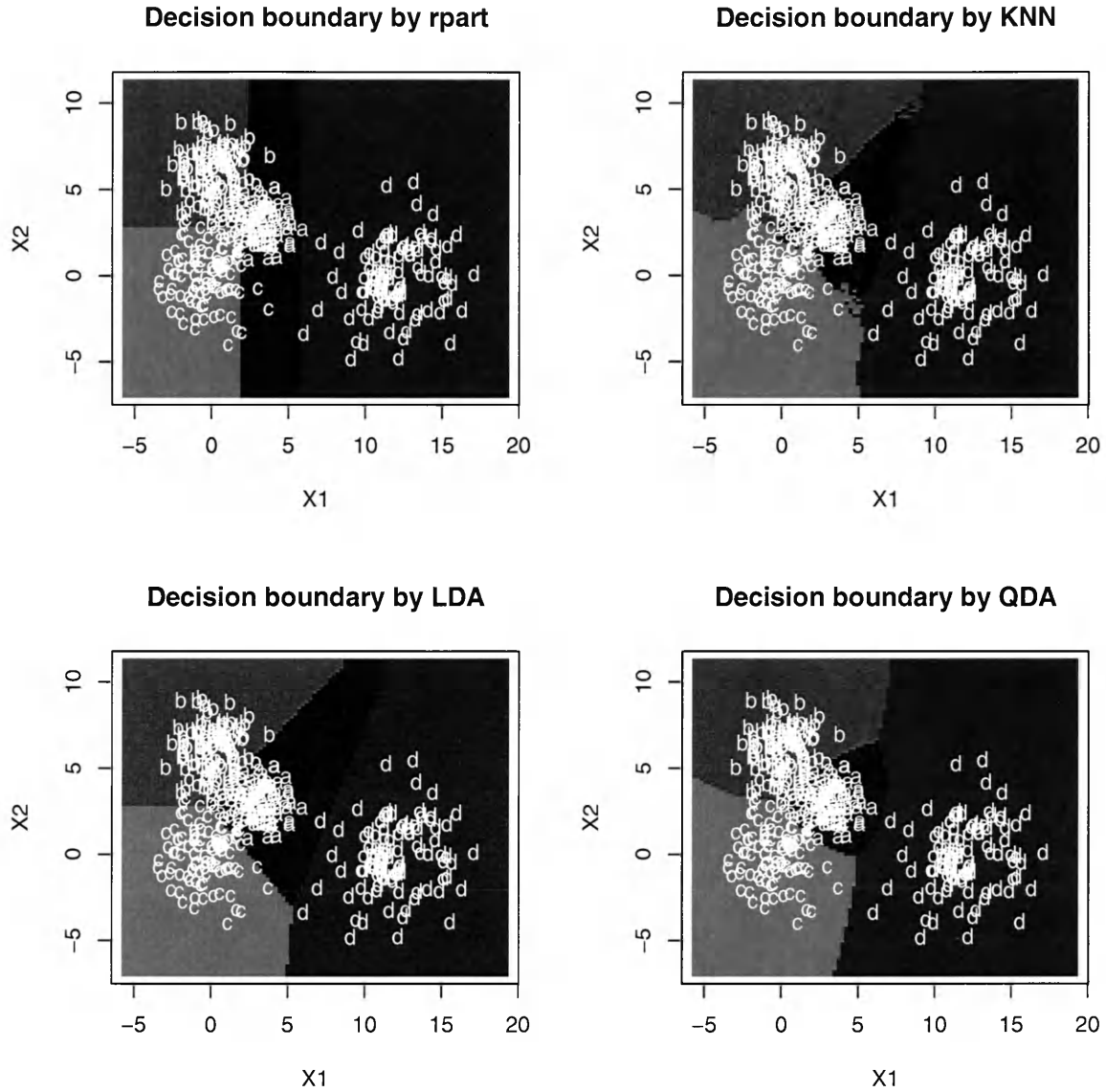


Figure 1.4: This figure shows the decision boundaries for four classification algorithms. From the top left to the right bottom are decision boundaries for: rpart, KNN, LDA and QDA. The training data set has four groups of samples. Each group is drawn from a normal distribution in a 2-dimensional Euclidean space. One group of samples is made *far* away from the other three groups, the other three groups has some overlapping.

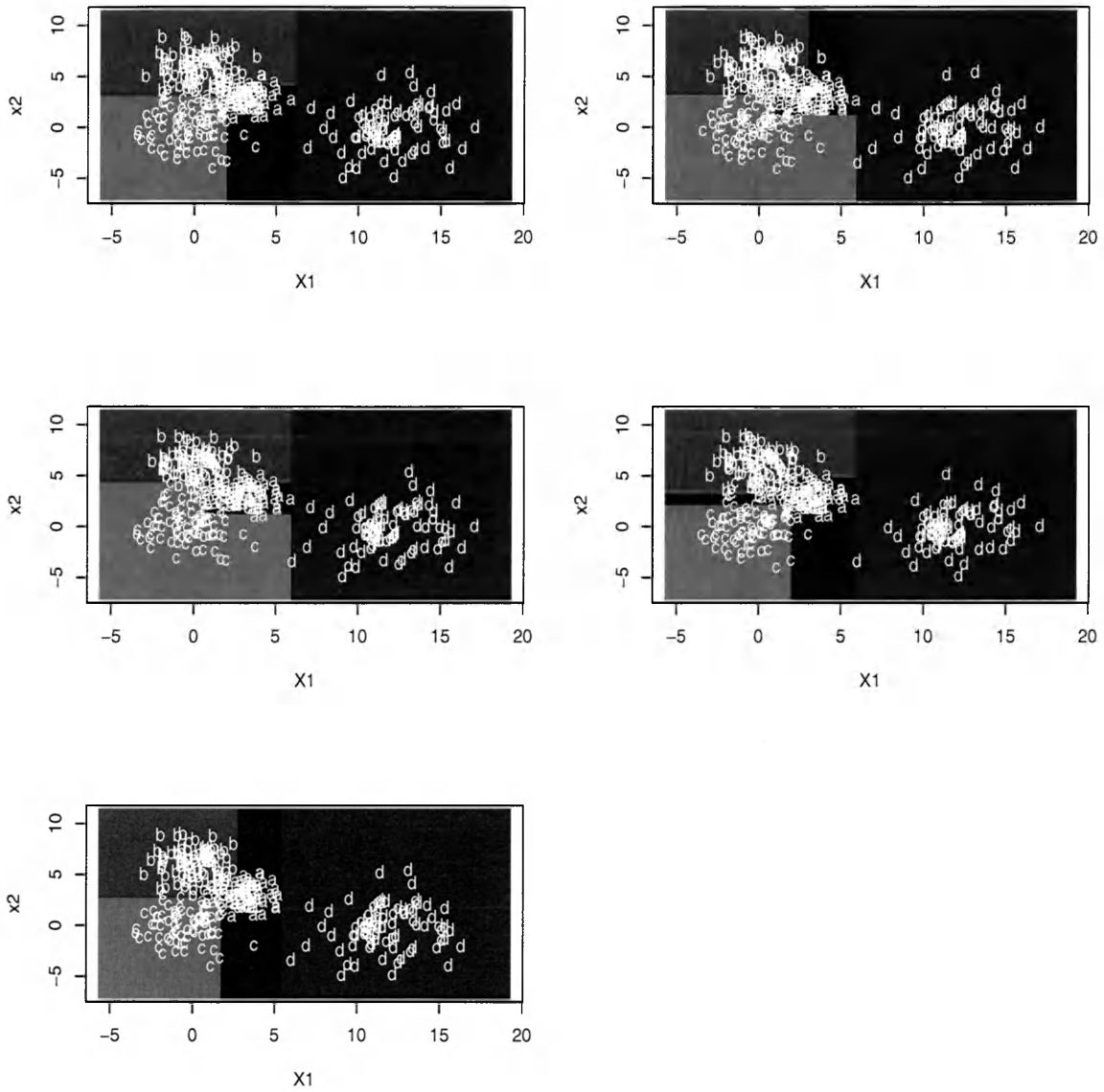


Figure 1.5: Decision boundaries during a 5-fold cross-validation procedure using rpart algorithm. Each small plot represents a different classifier based on a randomly selected 320 points from the original data. The original data has four groups of samples, each group has 100 subjects, and is independently normally distributed in a 2-d Euclidean space. This figure reveals the unstable nature of rpart for this specific pattern of data. We see that When 20% of the samples are replaced, the rpart decision boundaries change dramatically in a 5-fold cross-validation procedure.

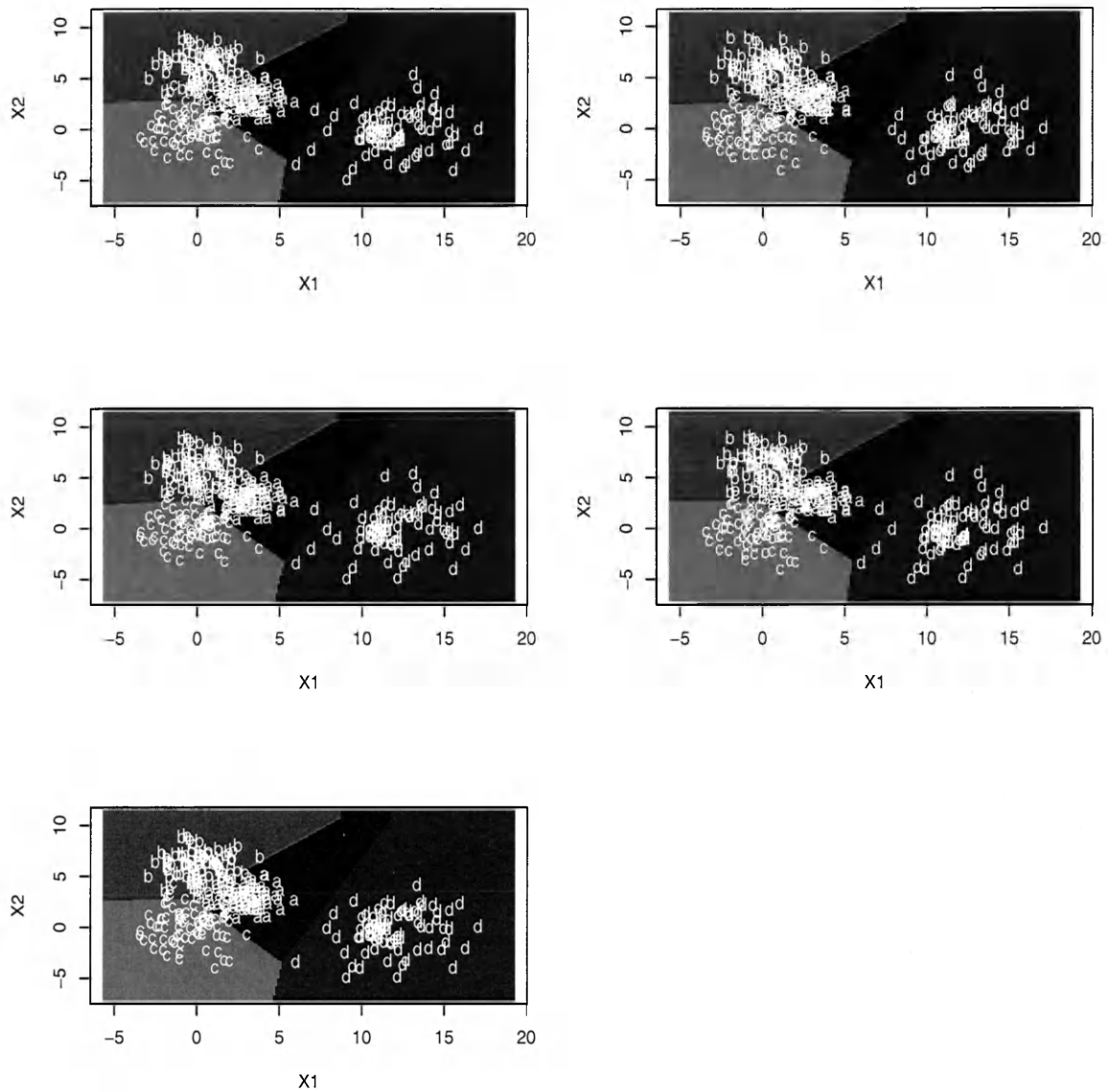


Figure 1.6: Decision boundaries during a 5-fold cross-validation procedure using LDA. Each small plot represents a different classifier based on a randomly selected 320 points from the original data. The original data has four groups of samples, each group has 100 subjects, and is independently normally distributed in a 2-d Euclidean space. Different with the dramatic variations of the decision boundaries in a cross-validation procedure in rpart, LDA's decision boundaries are very stable.

However, in Figure 1.6, the decision boundaries of LDA in a cross-validation training procedure are very stable, future samples will be classified into the same group consistently except points that are on or very close to the decision boundaries. This gives us an idea that the variance of the decision boundaries may also need to be considered when we are evaluating several classification methods.

Chapter 2

The Ambiguity and the Instability

In order to characterize the performance of classifiers, we have found it useful to introduce the new performance metrics in addition to the error rate: the *ambiguity* and *instability*.

2.1 The ambiguity

Ambiguity for a sample x gives us an idea of how difficult it is to assign a group label. For example, a point x has 0.6 vs. 0.4 probability to be classified into group a and group b . Since $p(x \in \text{group } a|x) > p(x \in \text{group } b|x)$, it will be classified into group a following a deterministic approach, but $p(x \in \text{group } a|x)$ and $p(x \in \text{group } b|x)$ are very close to each other, hence this classification is *ambiguous*. The ambiguity measures the uncertainty to classify a point into any groups. The usual misclassification rate does not contain any information about how uncertain is the classification. To rectify this, we introduce the definition of ambiguity.

Suppose x is a point from a p -dimensional Euclidean sample space X . The data set D is a subset of X . For a classification algorithm, we construct an ensemble of classifiers $C^{(k)}$, $k=1,2,\dots,N$, we require that the number of classifiers $N \gg$ the number of groups g . Each classifier $C^{(k)}$ maps points from the sample space X into the group

label set G with probability p , where p is a g dimensional real vector with positive entries that sum to one.

The posterior probability for point x from group i by classifier $C^{(k)}$ is defined as:

$$C^{(k)} : x \in X \rightarrow p^{(k)}(x \in \text{group } i|x), i = 1, \dots, g, k = 1, 2, \dots, N. \quad (2.1)$$

and

$$0 \leq p^{(k)}(x \in \text{group } i|x) \leq 1, \quad \text{and} \quad \sum_{i=1}^g p^{(k)}(x \in \text{group } i|x) = 1. \quad (2.2)$$

To decrease the variance in the ensemble of classifiers due to the random partitioning, we use the average of the posterior probability vectors to calculate the ambiguity. To measure how difficult or how uncertain point x is classified, the ambiguity function F should have the following properties:

1. It should be a nonnegative continuous real function in the sample space X ;
2. $F(p_1(x), p_2(x), \dots, p_g(x))$ reaches its peak when all $p_i(x)$ are equal to $\frac{1}{g}$, $p_i(x) = p(x \in \text{group } i|x)$;
3. F reaches its minimum when one of $p_i(x)$ is 1, and all the others are zeros.
4. F is symmetric under exchange:

$$F(p_1(x), p_2(x), \dots, p_g(x)) = F(p_{\sigma_1}(x), p_{\sigma_2}(x), \dots, p_{\sigma_g}(x))$$

One choice of F is the Shannon Entropy:

$$H(x) = - \sum_{i=1}^g \ln(p_i(x)) p_i(x), i = 1, 2, \dots, g. \quad (2.3)$$

There are many choices of ambiguity function, here we use the following definition

such that when the misclassification rate is small, the two measurements are very close to each other. It is also called the *Tsallis* entropy.

$$\Omega(x) \equiv \Omega(\bar{p}(x)) \equiv \frac{1}{2} \sum_{i=1}^g \bar{p}_i(x)(1 - \bar{p}_i(x)) = \frac{1}{2} \left[1 - \sum_{i=1}^g \bar{p}_i(x)^2 \right]. \quad (2.4)$$

$\Omega(x)$ reaches its maximum value $\frac{g-1}{2g}$ when the posterior probability for each group is the same. Note that each point in the sample space X has an ambiguity value, it does not need to be a sample from data D . $\Omega(x)$ is a function that is smooth everywhere in the sample space X . And we can measure a classification methodology's performance by integrating the ambiguity over the whole space to give us a global view of the ambiguity of that methodology. Using an appropriate prior probability $\pi(x)$, the global value of ambiguity can be measured as:

$$\Omega_\pi = \int_X \Omega(x)\pi(x)dx \quad (2.5)$$

This global value gives us a way to compare different classification algorithms. Examples are given in later sections.

2.2 The instability

The posterior probabilities for sample x , $p^{(k)}(x)$ from the ensemble of classifiers $C^{(k)}$ is stable if the N posterior probability vectors $\bar{p}^{(k)}(x)$, $k=1,2,\dots,N$ are close to each other. The *closeness* of two posterior probability vectors $p^{(i)}(x)$ and $p^{(j)}(x)$ is characterized by a normalized distance metric d . Like the usual distance metric, d satisfies the following rules:

1. Positivity: $d(p^{(i)}(x), p^{(j)}(x)) \geq 0$;
2. Symmetry: $d(p^{(i)}(x), p^{(j)}(x)) = d(p^{(j)}(x), p^{(i)}(x))$;
3. Triangular inequality: $d(p^{(i)}(x), p^{(j)}(x)) \leq d(p^{(i)}(x), p^{(k)}(x)) + d(p^{(k)}(x), p^{(j)}(x))$;

4. Uniqueness: if $d(p^{(i)}(x), p^{(j)}(x)) = 0, \Rightarrow p^{(i)}(x) = p^{(j)}(x)$.

In our definition, we want this measurement reach its maximum when the ensemble of probability vectors $p^{(1)}(x), p^{(2)}(x), \dots, p^{(N)}(x)$ are as much different as possible, such as one of the entries is 1, all the others are zeros. For example: suppose we have 2 groups of samples, and we have two classifiers which generates two posterior probability vectors for sample x : $p^{(1)}(x) = (1, 0), p^{(2)}(x) = (0, 1)$. In this case the two classification results completely disagree with each other and we consider the results *unstable*. We normalize the distance so that it reaches unity in such cases.

We can use the d_α distance as our distance metric:

$$\Delta^{ij}(x) \equiv d_\alpha(p^{(i)}(x), p^{(j)}(x)) \equiv \frac{1}{(2^{1/\alpha})} \left(\sum_{n=1}^g |p^{(i)}(x \in \text{group } n|x) - p^{(j)}(x \in \text{group } n|x)|^\alpha \right)^{1/\alpha} \quad (2.6)$$

Note that when α is 2, d is the normalized Euclidean distance. And the instability for a point x given an ensemble of classifiers $C^{(k)}$ is defined as:

$$I(x) \equiv \frac{1}{2N^2} \sum_{i,j=1}^N \Delta^{(ij)}(x). \quad (2.7)$$

The maximum value of $I(x)$ is $\frac{g-1}{2g}$, which equals the maximum value of $\Omega(x)$. And the instability reaches its minimum value of zero when all the posterior probability vectors are the same, even if the sample is classified incorrectly.

Again, $I(x)$ is defined on the whole sample space, and it is smooth everywhere. The integral of $I(x)$ using an appropriate prior probability $\pi(x)$ could give us a global view of the *stability* of a classification methodology.

$$I_\pi \equiv \int_X I(x)\pi(x)dx \quad (2.8)$$

2.3 The Error Rate

In deterministic methods, if $p(x \in \text{group } i|x)$ is the largest posterior probability for sample x , it will be classified into group i with probability one, and probability zero for all the other groups.

Definition The traditional definition of an error rate for a data set is computed as: the proportion of points that are misclassified using deterministic methods.

It is calculated as the number of samples misclassified divided by the total number of samples.

For example: the following is a confusion matrix from a classification study,

	a	b
a	80	20
b	10	90

the confusion matrix tells us that totally there are 200 samples, 20 samples from group a are misclassified into group b , 10 from group b are misclassified into group a . And the misclassification rate for this data is: $(20+10) / 200 = 15 \%$, to obtain this error rate, we require that all the samples' group labels are known.

Definition Error rate for a single point: the error rate for a sample x from group i in an ensemble of classifier $C^{(k)}$, $k=1,2,\dots,N$, using the same classification methodology is defined as:

$$E(x \in \text{group } i|x) = \frac{1}{N} \sum_{k=1}^N (1 - p^{(k)}(x \in \text{group } i|x)) = 1 - \bar{p}(x \in \text{group } i|x) \quad (2.9)$$

$p^{(k)}(i|x_n)$ is either 0 or 1, $\bar{p}(i|x)$ is the averaged posterior probability for $p^{(k)}(i|x)$, $k = 1, 2, \dots, N$; $i = 1, 2, \dots, g$.

As distinct from the traditional definition of error rate for a data set, our definition of the error rate for the entire data set is :

Definition Error rate for a data set:

$$E = 1 - \frac{1}{N} \sum_{i=1}^g \sum_{x_n \in i} \bar{p}(x \in \text{group } i | x_n) \quad (2.10)$$

The error rate for a point x whose true group label is i is defined as:

$$E(x) \equiv 1 - p(x \in \text{group } i | x), i=1,2,\dots,g \quad (2.11)$$

The Bayes error or Bayes risk can also be adopted. This estimate of the error rate can be used for any x in the sample space X , not just data points. Bayes rules use the maximum posterior probability to determine a sample's group assignment. The Bayes error for x is:

$$E_{Bayes}(x) \equiv 1 - \max_i p(x \in \text{group } i | x), i = 1, 2, \dots, g. \quad (2.12)$$

Bayes error is a nonnegative value with maximum of 0.5 which measures the *risk* to classify a sample. It helps to find the risky points and remove them before the classification. The idea of using ambiguity is very similar: points that are highly ambiguous should be removed from being classified. Figure 2.1 gives us an idea of the relationship between Bayes error and ambiguity.

Two groups of normally distributed samples are drawn from one dimensional Euclidean space, one follows normal distribution $N(0,0.1)$, one from distribution $N(1, 0.2)$. The curve that has a large peak on the left represents the population distribution for the first group, the second group's population distribution is plotted on the right with a smaller peak. In the middle of the figure there are two curves, the one with higher peak is the Bayes error, it is not differentiable at its peak, but is everywhere else. Below it is the ambiguity. Note that both Bayes error and

ambiguity reach their maximum value at the same point in this figure, where Bayes error equals 0.5 and the ambiguity equals 0.25. That point x can be derived from the following equation:

$$p(x \in \text{group 1}|x) = p(x \in \text{group 2}|x) \quad (2.13)$$

We have assumed that the prior probabilities for both groups are equal, thus

$$\frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} = \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} \quad (2.14)$$

We already know $\mu_1 = 0$, $\mu_2 = 1$, $\sigma_1 = 0.1$, $\sigma_2 = 0.2$, $\Rightarrow x \approx 0.35$.

The global Bayes error measure is given by the integral of all the *non-risky* points over the sample space X .

$$E_{Bayes} \equiv \int_{E_{Bayes} \leq \lambda_0} \pi(x) E_{Bayes}(x) dx = 1 - \int_{E_{Bayes} \leq \lambda_0} \pi(x) \max_i p(x \in \text{group } i|x) dx. \quad (2.15)$$

Where λ_0 is a threshold. We define risky points as those who have Bayes error larger than λ_0 and remove them in the integral because they are too risky to be classified. For example: if a sample x has posterior probability 0.49 for group 1, 0.51 for group 2, it will have 49% chance to be misclassified, so it is reasonable for us to remove such samples when estimating the overall Bayes error rate using equation (2.15).

Comparisons of Bayes error and Ambiguity measurements

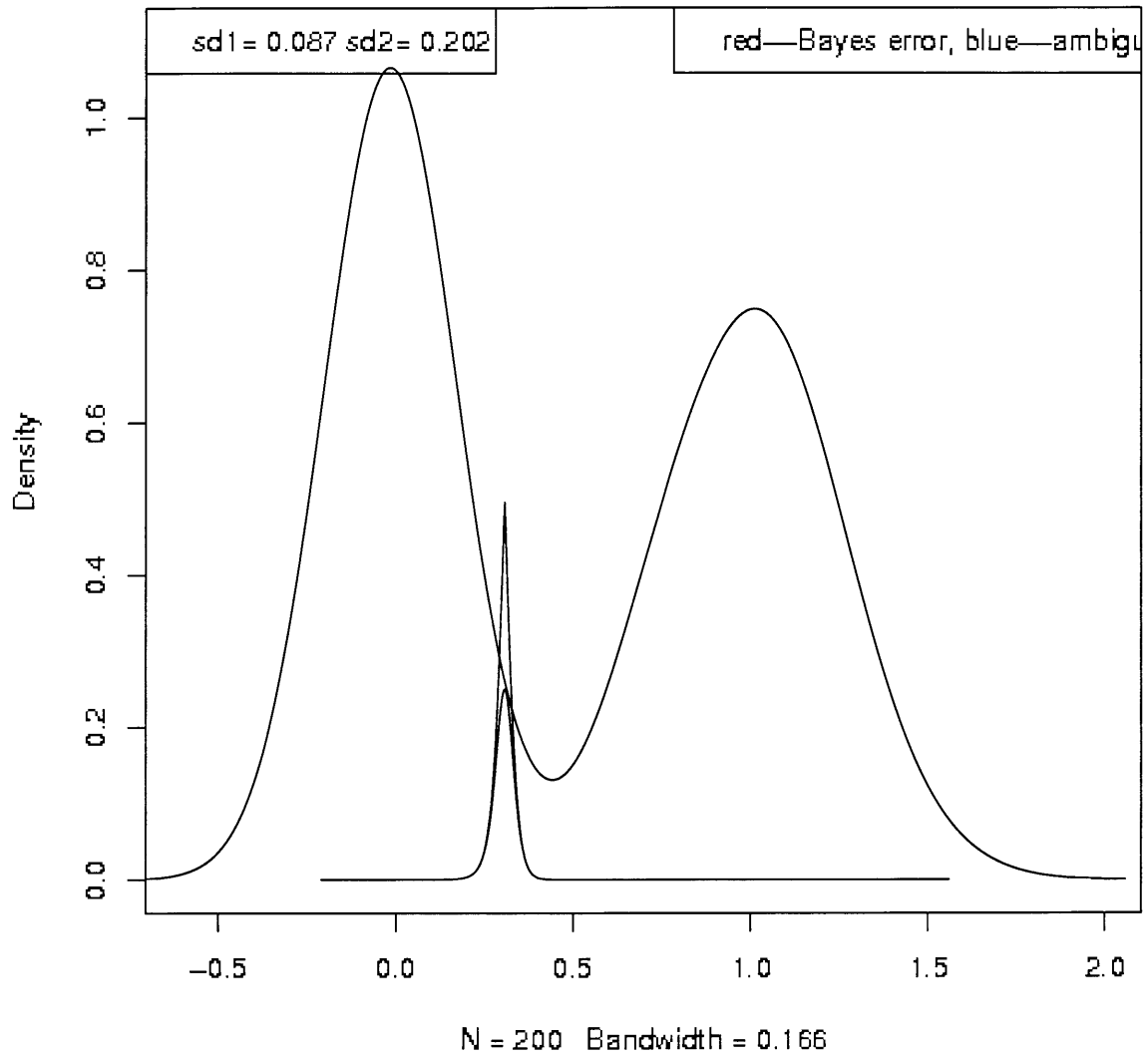


Figure 2.1: This example shows the similarity and differences of Bayes error and the ambiguity. Two groups of normally distributed samples are drawn from a one dimensional Euclidean space. The ambiguity and the Bayes error are very close to each other except for highly ambiguous (risky) points. The Bayes error is not smooth at its peak point because it contains a max function, while ambiguity is smooth everywhere.

2.4 Relationships of the error rate, ambiguity and instability

When we are choosing the definitions and the normalizations of the ambiguity, $\Omega(x)$, and the instability, $I(x)$, we have made them satisfy certain relationships. For deterministic methods, we have:

$$\Omega(x) = I(x), \forall x \in X \quad (2.16)$$

Proof of (2.16)

Suppose we have g groups and an ensemble of classifiers $C^{(k)}$, $k = 1, 2, \dots, N$. We require that $N \gg g$. Each classifier assigns a sample a g dimensional posterior probability vector $p^{(k)}(x) = (p^{(k)}(x \in \text{group } 1|x), p^{(k)}(x \in \text{group } 2|x), \dots, p^{(k)}(x \in \text{group } g|x))$. Among the N posterior probability vectors, n_i of them have maximum posterior probabilities for group i , $\sum_{i=1}^g n_i = N, 0 \leq n_i \leq N$. If we use a deterministic method, we will have n_i posterior probability vectors in the form of $(0, 0, \dots, 0, 1, 0, \dots, 0)$, where the i_{th} entry is 1 and all the others are zeros. Our ambiguity measure for x using a deterministic method is based on the averaged posterior probability vector $\bar{p}(x)$ which is:

$$\bar{p}(x) = \left(\frac{n_1}{N}, \frac{n_2}{N}, \dots, \frac{n_g}{N} \right) \quad (2.17)$$

According to our ambiguity definition,

$$\Omega(x) = \frac{1}{2} \left(1 - \sum_{i=1}^g \left(\frac{n_i}{N} \right)^2 \right) \quad (2.18)$$

Instability $I(x)$ is the normalized sum of the distances among those probability vectors. For a vector $p^{(k)}$ whose maximum posterior probability is $p_i^{(k)}(x)$, there are n_i (including itself) vectors that have zero distances with $p^{(k)}(x)$, and $N - n_i$

vectors that have of unity distances with $p^{(k)}(x)$. The sum of distances for the i_{th} is $n_i(N - n_i)$. Then the total instability I is:

$$I(x) = \frac{1}{2N^2} \sum_{i=1}^g n_i(N - n_i), \quad (2.19)$$

$$= \frac{n_1(N - n_1) + n_2(N - n_2) + \dots + n_g(N - n_g)}{2N^2}, \quad (2.20)$$

$$= \frac{1}{2} \left(1 - \sum_{i=1}^g \frac{n_i^2}{N} \right) = \Omega(x) \quad (2.21)$$

Hereby we have proved that for deterministic methods. The instability and ambiguity measurements have the same value.

Next we prove a very important relationship between the Bayes error rate and the ambiguity measurement.

$$\Omega(x) \leq E_{Bayes}(x), \forall x \in X \quad (2.22)$$

Proof Under the same assumptions in the last two proofs, we have:

$$\Omega(\bar{p}(x)) = \frac{1}{2} \left(1 - \sum_{i=1}^g \bar{p}_i(x)^2 \right) \quad (2.23)$$

$$= \frac{1}{2} \left(1 - \bar{p}_1^2 - \sum_{i=2}^g \bar{p}_i(x)^2 \right) \quad (2.24)$$

$$= \frac{1}{2} (1 + \bar{p}_1(x))(1 - \bar{p}_1(x)) - \frac{1}{2} \sum_{i=2}^g \bar{p}_i(x)^2 \quad (2.25)$$

Because $0 \leq \bar{p}_1(x) \leq 1$,

$$\Rightarrow \frac{1}{2} (1 + \bar{p}_1(x)) \leq 1, \Rightarrow \frac{1}{2} (1 + \bar{p}_1(x))(1 - \bar{p}_1(x)) \leq 1 - \bar{p}_1(x) \quad (2.26)$$

$$\Rightarrow \frac{1}{2}(1 + \bar{p}_1(x))(1 - \bar{p}_1(x)) - \frac{1}{2} \sum_{i=2}^g \bar{p}_i(x)^2 \leq 1 - \bar{p}_1(x) - \frac{1}{2} \sum_{i=2}^g \bar{p}_i(x)^2 \quad (2.27)$$

And,

$$\frac{1}{2} \sum_{i=2}^g \bar{p}_i(x)^2 \geq 0 \Rightarrow \Omega(\bar{p}(x)) \leq 1 - \bar{p}_1(x) = E_{Bayes}(x). \quad (2.28)$$

We summarize the relationships of ambiguity, instability and error rate here as:

$$I_{det}(x) = \Omega_{det}(x) \leq E_{Bayes}(x), \forall x \in X. \quad (2.29)$$

$$\Omega_{prob}(x) \leq E_{Bayes}(x), \forall x \in X. \quad (2.30)$$

Because equation 2.29 and 2.30 are true for all the points in the sample space, and if we integrate the above 2.29 over all the points in the space, we have:

$$I_{det} = \Omega_{det} \leq E_{Bayes} \quad (2.31)$$

And

$$\Omega_{prob} \leq E_{Bayes}, \forall x \in X. \quad (2.32)$$

This reflects the relationships of the three criteria in the global point of view. And we must mention that there is no fixed relationship for instability using probabilistic method with its deterministic version. Also there is no fixed relation between probabilistic ambiguity and deterministic ambiguity.

Figure 2.2 and Figure 2.3 are two heatmaps showing the ambiguous areas using the same data. Figure 2.2 uses probabilistic LDA method, and Figure 2.3 uses deterministic LDA method. The data has four different groups of samples. Each group has 100 subjects which are drawn from a 2-dimensional normal distribution. The

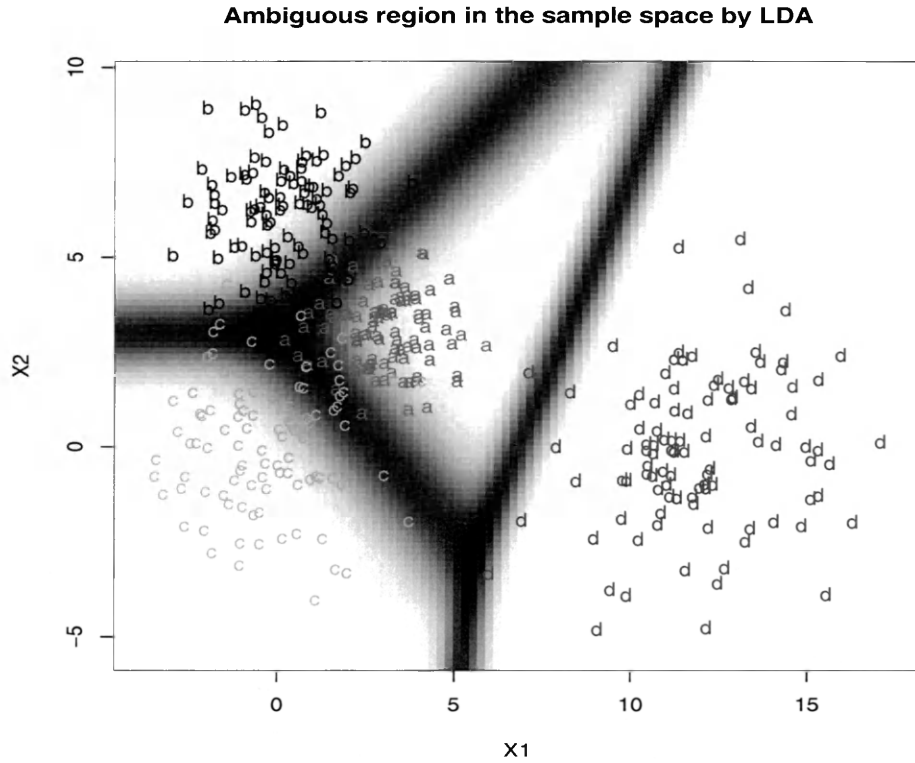


Figure 2.2: This is the heatmap that shows the ambiguous regions using LDA with probabilistic method. The data has four different groups of samples. Each group has 100 subjects which are drawn from a 2-dimensional normal distribution. 5-fold cross-validation is repeated 10 times and 5 classifiers are generated. The ambiguity are calculated based on the averaged posterior probability vector.

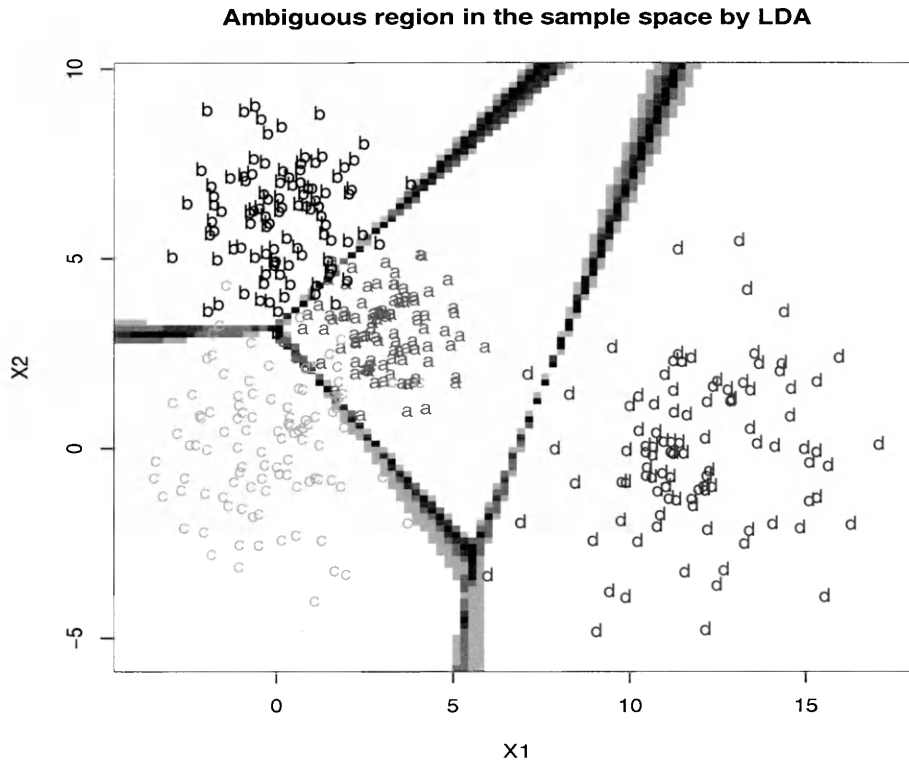


Figure 2.3: This is the heatmap that shows the ambiguous regions using LDA with deterministic method. The data has four different groups of samples. Each group has 100 subjects which are drawn from a 2-dimensional normal distribution. 5-fold cross-validation is repeated 10 times and 5 classifiers are generated. The ambiguity are calculated based on the averaged posterior probability vector. The ambiguous regions in this plot are much *sharper* than the probabilistic method.

ambiguous regions with deterministic method are much *sharper* than the probabilistic method. The volume of ambiguity areas using deterministic LDA seems to be less than probabilistic method, except in some extreme cases (such as the data is well separated that each sample is unambiguously assigned a group label), in which the two measurements are very close or even share the same values. Please also note that the fact that deterministic method has smaller ambiguous regions does not mean it is better than probabilistic method. For example, points that are ambiguous in Figure 2.2 are highly possible to be classified incorrectly. We claim that those points should be removed because they are too risky to be classified. And if we only remove the ambiguous points using deterministic method, the ambiguous points we found with probabilistic method may increase the misclassification rate a lot.

Chapter 3

Numerical experiments

To better illustrate the idea that the ambiguity and instability are useful classifier evaluation criteria, we have done a series of experiments using three different data sets. For visualization convenience, all the data sets are drawn from two dimensional Euclidean spaces. Four classification techniques: rpart, LDA, QDA and KNN are used, each generates an ensemble of classifiers after repeating cross-validation procedure several times.

3.1 Introduction to the three data sets

Figure 3.1 shows the first data which we call it *data.2g*. It is comprised of two groups of 2-dimensional normally distributed samples, each group has 1000 points. The two groups of samples share the same covariance matrix. The small overlapping of the two groups helps us to study the three properties: the ambiguity, the instability and the error rate. Figure 3.2 is the scatterplot of the second data set *data.4g* that has four normally distributed groups of samples, each group has 100 points, one group is made *far* away from the other three, while the other three groups have some overlapping. The last data set *data.ng* is plotted on Figure 3.3. *data.ng* has two groups of samples, the first group has 200 normally distributed samples, and

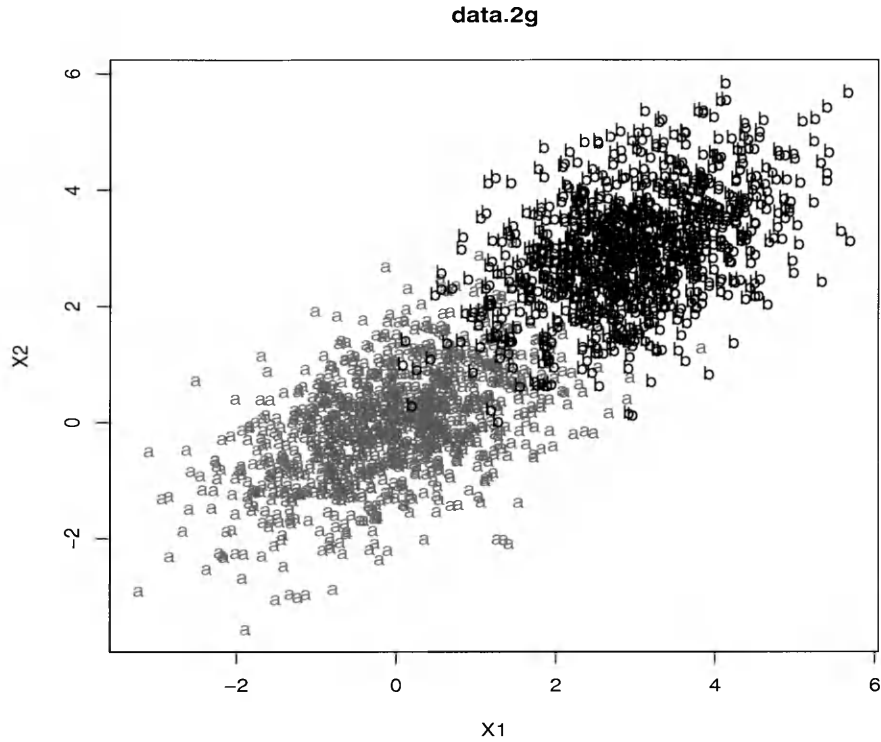


Figure 3.1: *data.2g* has 2 groups of samples, and we use letter *a* and letter *b* to represent samples from the first and the second group. They both have 1000 samples that are drawn from a normal distribution in a 2-dimensional Euclidean space and share the same covariance structure. They have a small overlapping in the middle of this figure.

second group has 200 samples that are not from a normal distribution but has a special structure. This specific pattern helps us to better understand the different behaviors of linear classification algorithm such as LDA and nonlinear methods such as QDA.

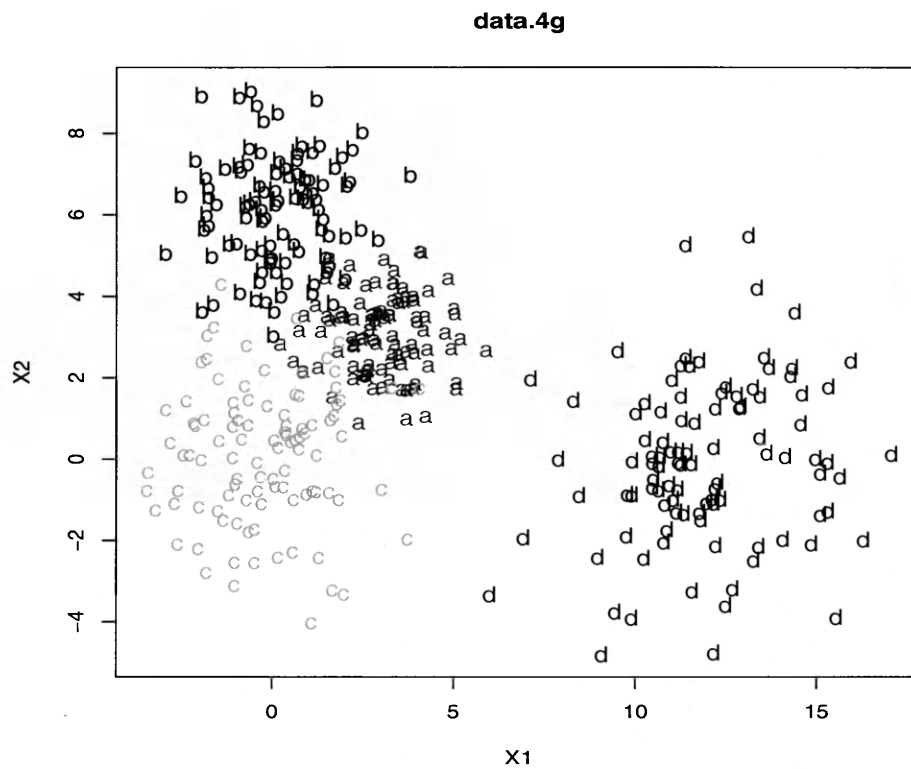


Figure 3.2: *data.4g* has four normally distributed groups of samples. Each group has 100 samples drawn from a 2-dimensional Euclidean space. They are represented by *a*, *b*, *c* and *d*. Group *d* is made far away from the other three groups, and the other three groups have a small overlapping.

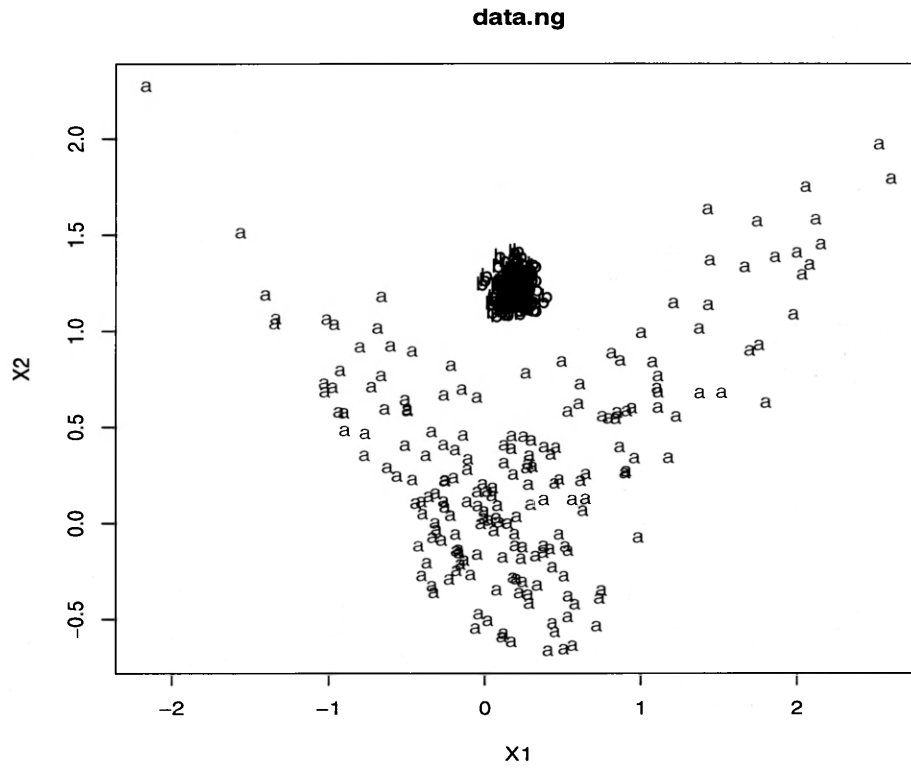


Figure 3.3: *data.ng* has two groups of samples, each of them has 200 subjects. The two groups are represented by letter *a* and letter *b* in the scatterplot. The group *b* follows a normal distribution, group *a* has a special structure that could help us to better understand the different behaviors of linear and nonlinear algorithms.

3.2 Results

We compute the error rate, ambiguity and instability based on the ensemble of classifiers generated by a 5-fold cross-validation, each cross-validation procedure is repeated 10 times such that for each classification algorithm we create 50 classifiers. Algorithms LDA, QDA, rpart and KNN are evaluated. The estimated numerical results of them are recorded in Table 3.1 and Table 3.2.

Summary of global results			
Classification method	<i>data.2g</i>	<i>data.4g</i>	<i>data.ng</i>
LDA (det)	$\Omega = 0.001$ $E = 0.0382$	$\Omega = 0.004$ $E = 0.052$	$\Omega = 0.008$ $E = 0.103$
LDA (prob)	$\Omega = 0.028$ $I = 0.001$	$\Omega = 0.060$ $I = 0.004$	$\Omega = 0.070$ $I = 0.009$
QDA (det)	$\Omega = 0.001$ $E = 0.040$	$\Omega = 0.003$ $E = 0.050$	$\Omega = 0.0$ $E = 0.0$
QDA (prob)	$\Omega = 0.028$ $I = 0.001$	$\Omega = 0.038$ $I = 0.004$	$\Omega = 0.008$ $I = 0.001$
rpart (det)	$\Omega = 0.011$ $E = 0.046$	$\Omega = 0.036$ $E = 0.106$	$\Omega = 0.001$ $E = 0.013$
rpart (prob)	$\Omega = 0.041$ $I = 0.015$	$\Omega = 0.069$ $I = 0.036$	$\Omega = 0.011$ $I = 0.002$
KNN (det)	$\Omega = 0.006$ $E = 0.040$	$\Omega = 0.007$ $E = 0.060$	$\Omega = 0.001$ $E = 0.002$
KNN (prob)	$\Omega = 0.022$ $I = 0.006$	$\Omega = 0.037$ $I = 0.010$	$\Omega = 0.001$ $I = 0.0$

Table 3.1: A summary of the results in the numerical experiments. Four classification methods: LDA, QDA, rpart and KNN are applied on *data.2g*, *data.4g* and *data.ng*. Global views of the $\Omega(x)$, $I(x)$ and $E(x)$ for a classification technique are summarized in the table while the unstable and ambiguous regions are shown in the figures.

Table 3.1 is the summary of the three measurements: ambiguity, instability and error rate. The ambiguity Ω and instability I are calculated as the global averages over the sample space. The error rate E is calculated as the proportion of misclassified test samples using deterministic methods. As we mentioned in the

Summary of figures			
<i>Deterministic methods</i>			
ambiguity = instability	Fig. (3.4)	Fig. (3.7)	Fig. (3.10)
<i>Probabilistic methods</i>			
Ambiguities	Fig. (3.5)	Fig. (3.8)	Fig. (3.11)
Instabilities	Fig. (3.6)	Fig. (3.9)	Fig. (3.12)

Table 3.2: Summary of the figures and the measurements they represent.

Summary of ambiguity vs. Bayes error			
Classification method	<i>data.2g</i>	<i>data.4g</i>	<i>data.ng</i>
LDA	$E_{Bayes} = 0.016$ $\Omega' = 0.015$	$E_{Bayes} = 0.034$ $\Omega' = 0.031$	$E_{Bayes} = 0.061$ $\Omega' = 0.055$
QDA	$E_{Bayes} = 0.016$ $\Omega' = 0.015$	$E_{Bayes} = 0.019$ $\Omega' = 0.017$	$E_{Bayes} = 0.006$ $\Omega' = 0.006$
rpart	$E_{Bayes} = 0.035$ $\Omega' = 0.034$	$E_{Bayes} = 0.059$ $\Omega' = 0.054$	$E_{Bayes} = 0.010$ $\Omega' = 0.010$
KNN	$E_{Bayes} = 0.011$ $\Omega' = 0.010$	$E_{Bayes} = 0.012$ $\Omega' = 0.011$	$E_{Bayes} = 0.000$ $\Omega' = 0.000$

Table 3.3: The Bayes error and ambiguity after highly ambiguous points are removed, in this case the averaged ambiguity and averaged Bayes error are nearly equal, which means ambiguity measure can be used as a substitute for Bayes error.

section 1, the error rate is the traditional way of evaluating a classification technique. Table 3.1 reveals the fact that although the four algorithms may have comparable misclassification rates, their ambiguity and instability may be quite different. For example, the error rates on *data.2g* for all four are very close, their ambiguity measurements are quite different. Rpart has much larger instability than the other three methods. This fact is also supported on Figure 3.6, where the heatmap suggest that algorithm LDA, QDA and KNN have smaller unstable regions on the sample space than rpart. When there are future samples drawn from the unstable regions, rpart will be the one that most likely to generate a lot of unstable classification results. For example, a sample's classified group labels change dramatically during a cross-validation process. With the help of ambiguity measures on Table 3.1 and

the heatmaps of ambiguous regions showing on Figure 3.4 and 3.5, we have found out that it is also unwise to use rpart as the classification technique to predict future samples for data that have similar structure with *data.2g*, but we can not find much difference if we only use error rate.

The third and fourth columns on Table 3.1 are the measurements for the other two data sets, *data.4g* and *data.ng*. Rpart still has the largest ambiguous and unstable areas for *data.4g* which are shown on Figure 3.7, 3.8 and 3.9. The error rate and ambiguity of those four algorithms are comparable but rpart has the largest. Rpart has much larger instability than the other three algorithms: about 10 times of LDA and QDA, and about 4 times of KNN.

The last column on Table 3.1 are the numerical results from *data.ng*. Nonlinear classification algorithms QDA and KNN are almost perfect for classifying such data. All the three measurements of QDA and KNN are very small. Linear classification algorithms LDA and rpart have much larger error rates than QDA and KNN. LDA becomes the most ambiguous and most unstable algorithm. It also has very large ambiguity and instability measures than the other three, which means for data having similar pattern with *data.ng*, we should not use LDA.

Table 3.2 is the summary of the figures and the specific measurements they represent.

Table 3.3 summarizes the Bayes error and the ambiguity after ambiguous points are removed. The threshold λ is set to be 0.1, and points that are more ambiguous than 0.1 are not included in the classification and calculation. The maximum ambiguity for any point is $\frac{g-1}{2g}$. In a two group classification problem, point with a 0.1 ambiguity has about 70% posterior probability in one group and 30% probability to be from the other group. We can see the two measures are very close, which means ambiguity can be used as a substitute for Bayes error. But analytically, the smooth feature of ambiguity makes it better than Bayes error.

From Figure 3.4 to Figure 3.12, probabilistic ambiguity, deterministic ambiguity and probabilistic instability heatmaps are plotted sequentially for each of the three data sets. As we discussed in section 2.4, it is true that the deterministic instability for any point in the sample space has the same value with its deterministic ambiguity, so we ignored the deterministic instability heatmaps. The four statistical algorithms are also in a fixed order: from left to right, top to bottom are: LDA(top left), QDA(top right), rpart(bottom left) and KNN(bottom right). They are plotted for all the points (not only the data points) on the 2-dimensional sample space, the dark regions represent the ambiguous points or the unstable points, the darker, the more ambiguous or more unstable they are.

From Figure 3.4 to Figure 3.9, we can find that LDA, QDA, and KNN algorithms seem to share the same topological structure of the ambiguity and instability areas, the rpart's ambiguous and instability regions are much larger than the other three, and its decision boundaries seem to be the most unstable.

Figure 3.6 shows where the unstable areas are for *data.2g*. Future points drawn from there can be classified by different group labels: sometimes from group *a*, sometimes from group *b*. Rpart has the largest unstable area, from the instability point of view, we should choose the other three algorithms instead of rpart for this kind of data. Comparing Figure 3.5 to Figure 3.6 we can find that in the overlapping areas, points are very ambiguous, but they are very stable, which means that the group label assignments of those points are consistent among the 50 classifiers.

Figure 3.7, Figure 3.8 are ambiguity and instability heatmaps with the data comprised of four groups of normally distributed samples, *data.4g*. This data creates very similar pattern of ambiguous and unstable regions as *data.2g*. And from Table 3.1, the two data sets' results both agree on the fact that: although the four classification techniques have comparable cross-validated error rates, rpart's instability and ambiguity measurements are much larger than the other three. In fact,

from Table 3.1 we find that for probabilistic method, rpart has about 4 times the instability of KNN and 10 times that of LDA and QDA in *data.4g*.

The dark regions represent the unstable points in Figure 3.6, 3.9 and 3.12. It give us an approximate idea of how the decision boundaries are changing of the four algorithms using the same data sets.

Figure 3.5 shows us that rpart is splitting the data either vertically or horizontally so that the probabilities for the points in the dark regions may have quite different group assignments during the 10 repeated cross-validation procedures. If we are given a sample which is in the ambiguous regions on the left bottom, rpart will classify it with a highly ambiguous probability. And rpart may not be appropriate to be used for predicting new samples.

Comparing Figure3.6 to the ambiguity heatmap from Figure 3.5, it seems that the unstable regions are smaller than ambiguous region for all the three data sets, the more data points the more stable also seems true for LDA, QDA and KNN, but not for rpart.

Figure 3.10, 3.11 and 3.12 are the results on the *data.ng*, which has two groups of samples, 200 each, one is from a normal distribution, one is not. QDA's ambiguous and unstable regions are the smallest among the four algorithms, LDA behaves poor as it has the largest ambiguous areas as well as the largest unstable areas, which is probably because one group is not following Gaussian distribution which assumed by LDA. And it is impossible for any single cut to separate the two groups in any directions for LDA. Roughly speaking, the rpart decision boundaries are still the most unstable among the four as the Table 3.1 tells us, and KNN performs well, if not as well as QDA.

Two gaussians: Ambiguity plots for four different classifiers using probability methods.

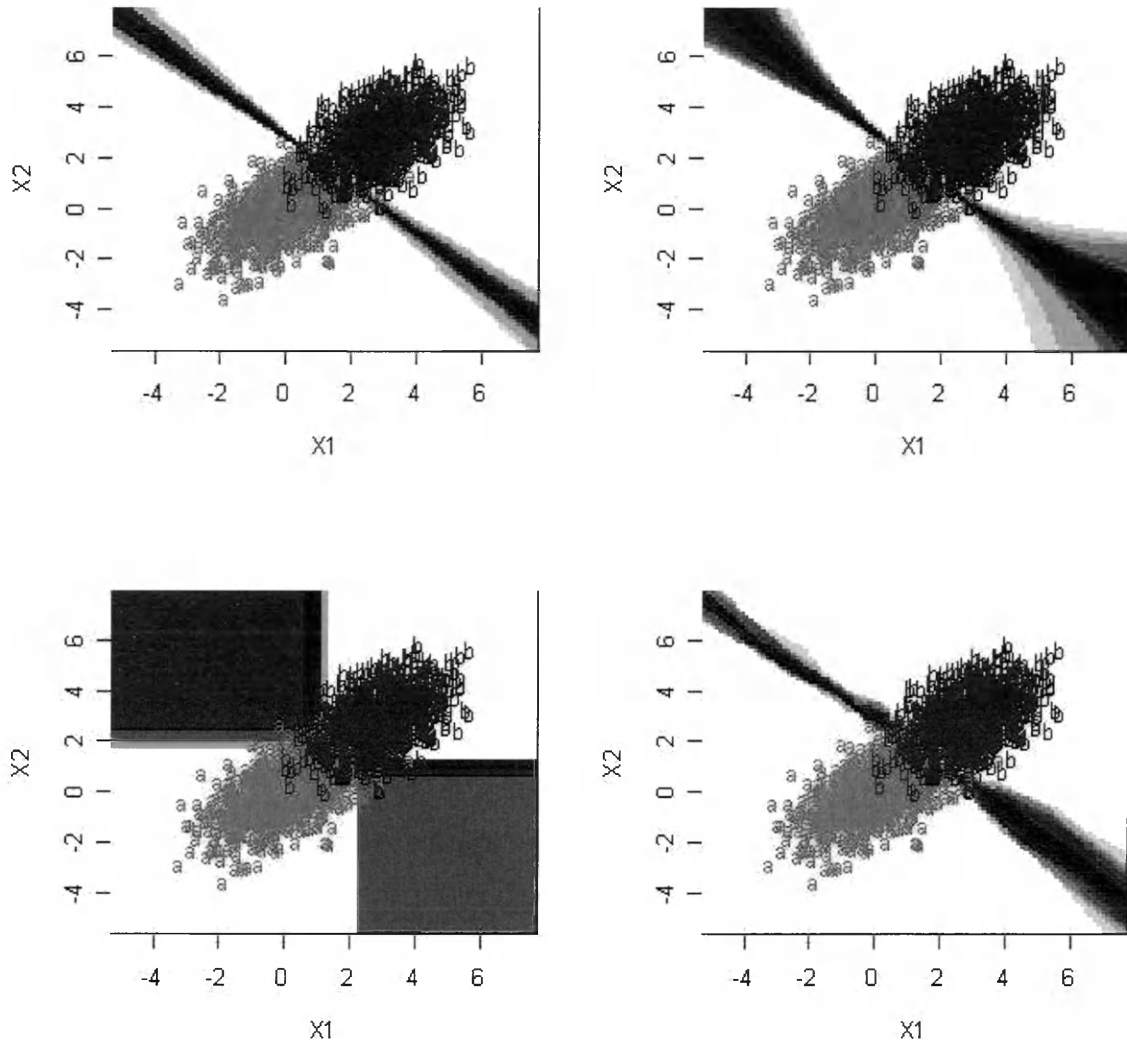


Figure 3.4: The data is composed of two groups, we used a and b to represent them in the plot. Each group has 1000 normally distributed samples that are drawn from a 2-dimensional Euclidean space. The four classification algorithms from left to right, top to bottom are: LDA (top left), QDA (top right), rpart (bottom left) and KNN (bottom right). They are plotted for all the points (not only the data points) on the 2-dimensional sample space, the dark regions represent the ambiguous points, the darker, the more ambiguous they are. Probabilistic method helps us finding out the ambiguous areas on the sample space.

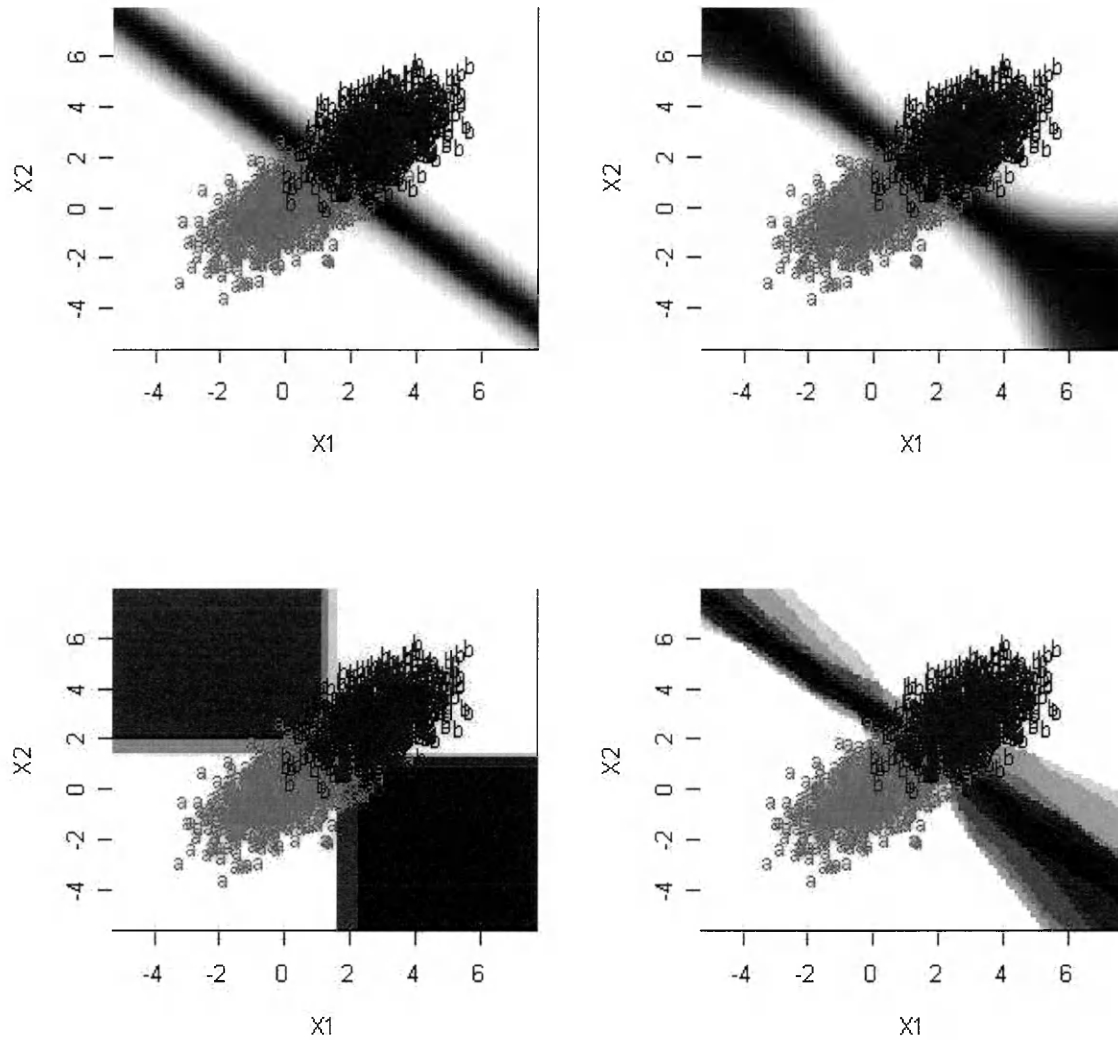


Figure 3.5: The data is composed of two groups: a and b , each with 100 samples drawn from a 2-dimensional Euclidean space. This figure shows the ambiguous regions with deterministic methods. The four classification algorithms from left to right, top to bottom are: LDA (top left), QDA (top right), rpart (bottom left) and KNN (bottom right). They are plotted for all the points (not only the data points) on the 2-dimensional sample space, the dark regions represent the ambiguous points, the darker, the more ambiguous they are.

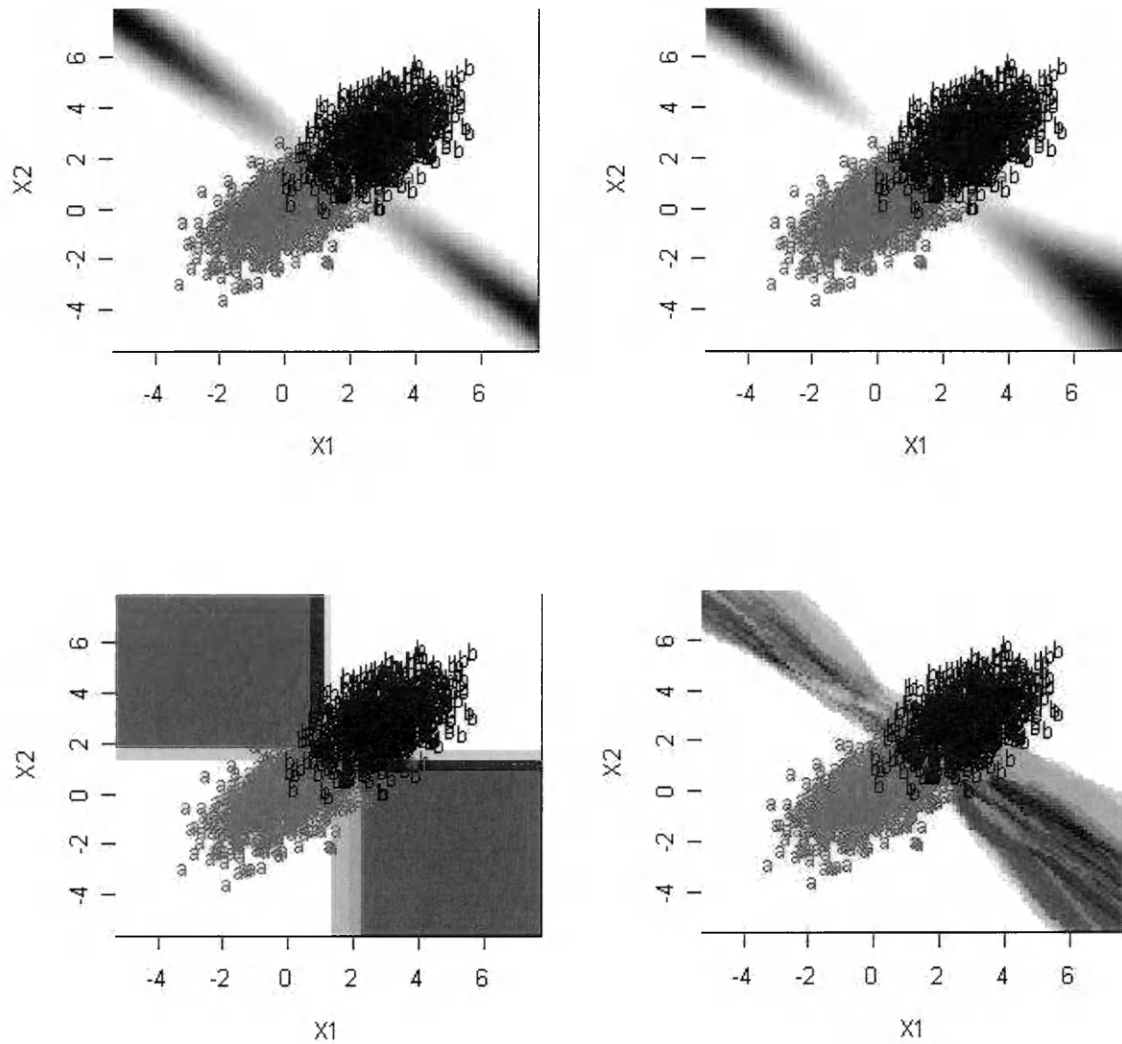


Figure 3.6: The data is composed of two groups, a and b , each with 100 samples drawn from a 2-dimensional Euclidean space. This plot shows the unstable areas which are discovered by probabilistic methods. The four classification algorithms from left to right, top to bottom are: LDA (top left), QDA (top right), rpart (bottom left) and KNN (bottom right). They are plotted for all the points (not only the data points) on the 2-dimensional sample space, the dark regions represent the unstable points, the darker, the more unstable they are.

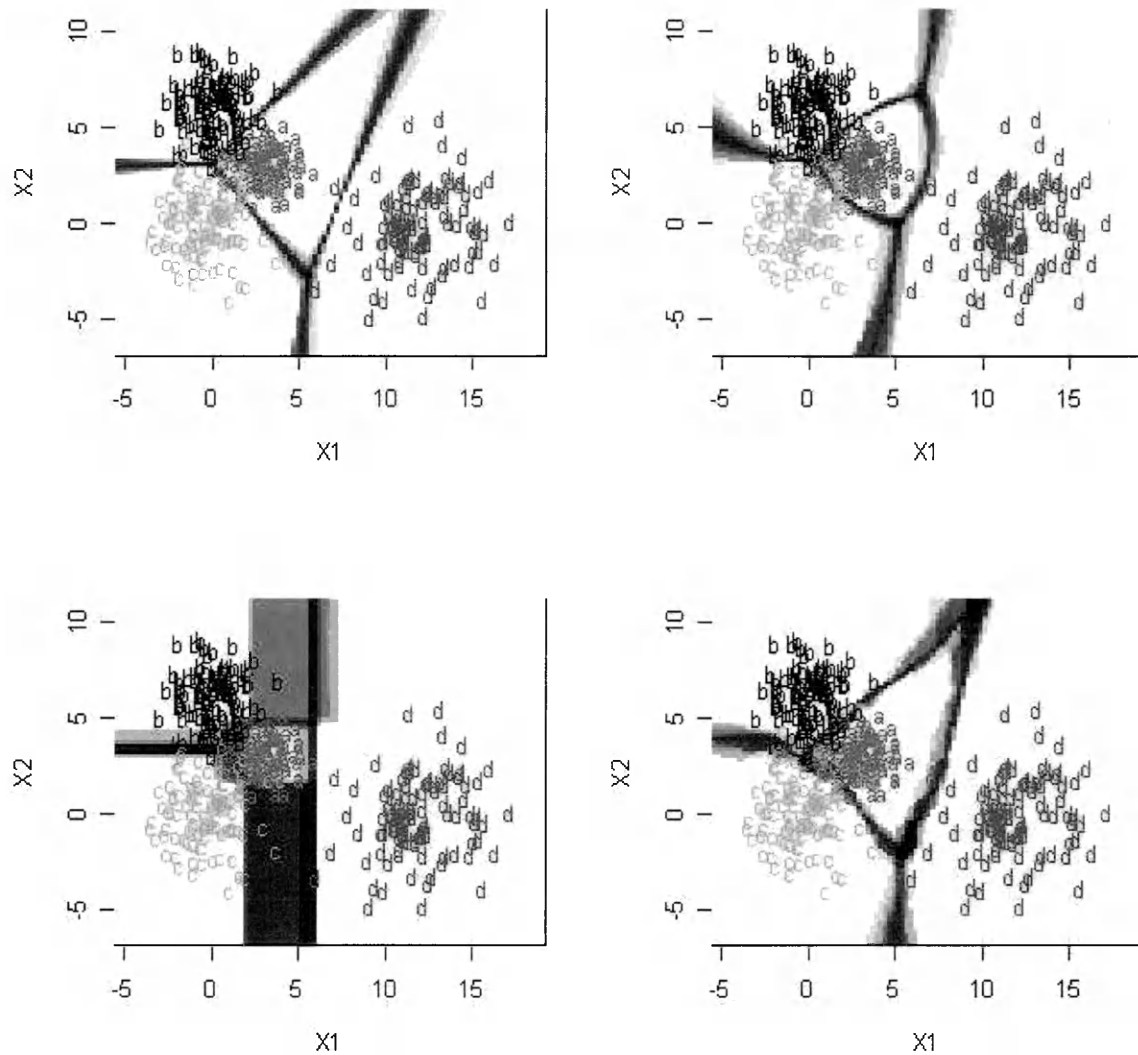


Figure 3.7: The data in this figure is comprised of four groups of normally distributed 2-dimensional samples. They are represented by letters *a*, *b*, *c* and *d*. Group *d* is made far away from the other three groups, and the other three groups have a small overlapping. Each group has 100 samples and their group labels are already known. The classification algorithms are shown as: LDA on the top left, QDA on the top right, rpart on the bottom left and KNN on the bottom right. The ambiguous areas of four algorithms are calculated using deterministic methods.

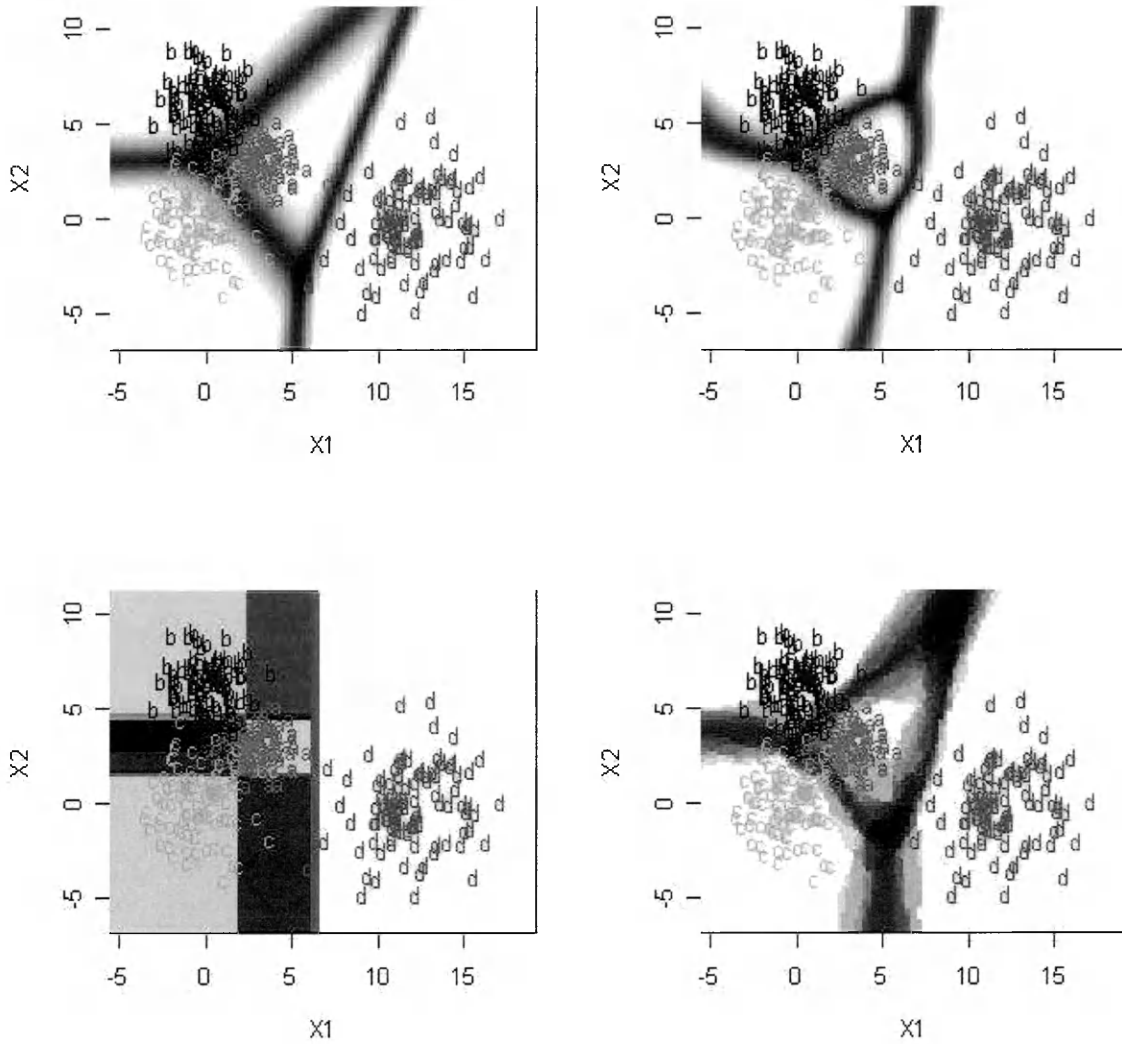


Figure 3.8: The data in this figure is comprised of four groups of normally distributed 2-dimensional samples. They are represented by letters *a*, *b*, *c* and *d*. Group *d* is made far away from the other three groups, and the other three groups have a small overlapping. Each group has 100 samples and their group labels are already known. The classification algorithms are shown as: LDA on the top left, QDA on the top right, rpart on the bottom left and KNN on the bottom right. Ambiguous points are highlighted with dark colors, their ambiguity measures are computed by probabilistic methods.

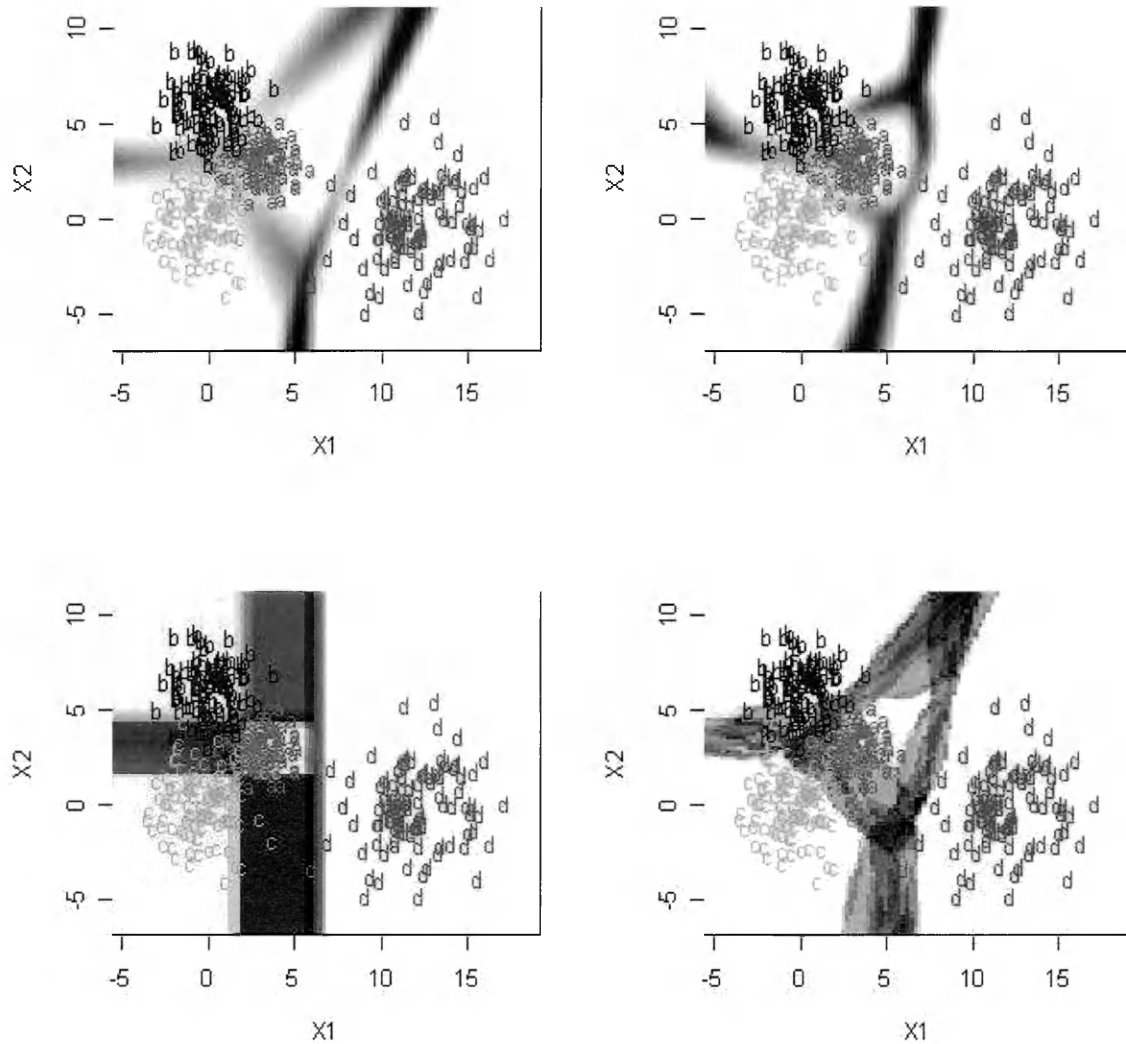


Figure 3.9: The data in this figure is comprised of four groups of normally distributed 2-dimensional samples. They are represented by letters *a*, *b*, *c* and *d*. Group *d* is made far away from the other three groups, and the other three groups have a small overlapping. Each group has 100 samples and their group labels are already known. The classification algorithms are shown as: LDA on the top left, QDA on the top right, rpart on the bottom left and KNN on the bottom right. Four algorithms' unstable areas are plotted with different grayscales which are proportional to their instability values. Comparing to other three algorithms, rpart has a wide unstable areas, which means for this pattern of data, if we cross-validate rpart to predict future samples, the group assignments could be very unstable. And we should be very careful using rpart on this kind of data, although the misclassification rate are very similar with the other three algorithms.

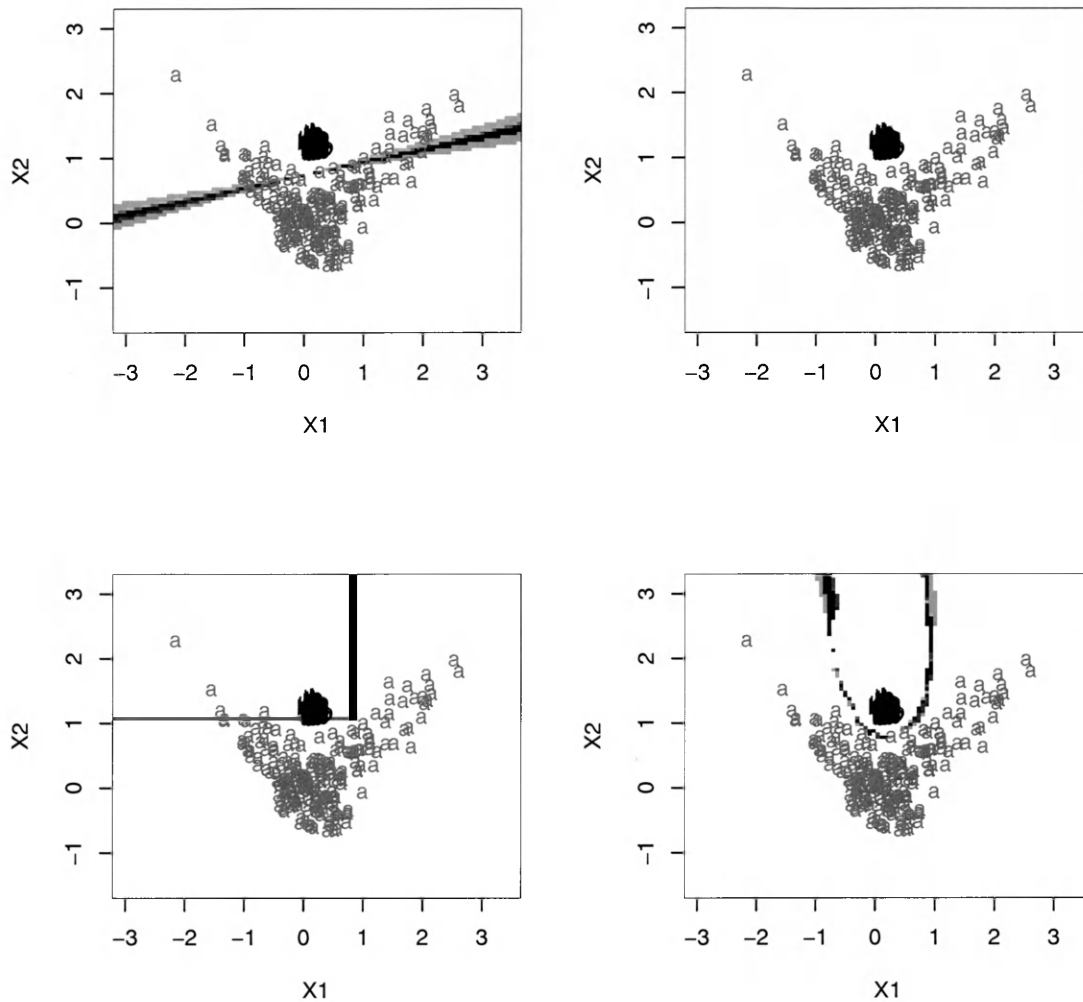


Figure 3.10: The data in this figure is *data.ng*, which has two groups of samples, each of them has 200 subjects. The first group is represented by letter *a* in the figure, second by letter *b*. Group *a* follows a normal distribution, group *b* has a special structure that could help us to better understand the different behaviors of linear and nonlinear algorithms. The classification algorithms are shown as: LDA on the top left, QDA on the top right, rpart on the bottom left and KNN on the bottom right. This figure highlight the ambiguous areas by deterministic methods, the size of the most ambiguous areas is much smaller (*sharper*) than probabilistic method but they have very similar topological pattern.

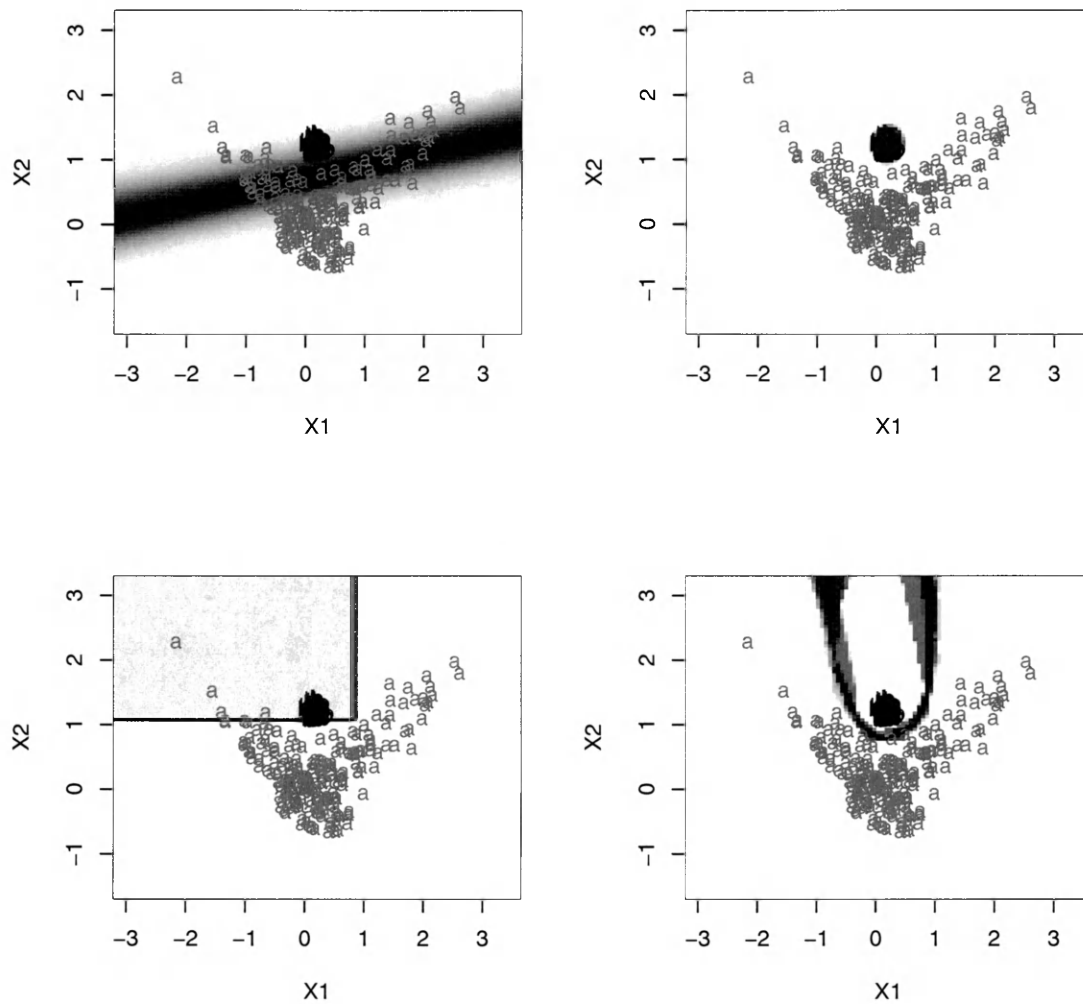


Figure 3.11: The data in this figure is *data.ng*, which has two groups of samples, each of them has 200 subjects. The first group is represented by letter *a* in the figure, second by letter *b*. Group *a* follows a normal distribution, group *b* has a special structure that could help us to better understand the different behaviors of linear and nonlinear algorithms. The classification algorithms are shown as: LDA on the top left, QDA on the top right, rpart on the bottom left and KNN on the bottom right. The ambiguous areas are highlighted by dark colors and their ambiguity measures are computed by probabilistic methods.

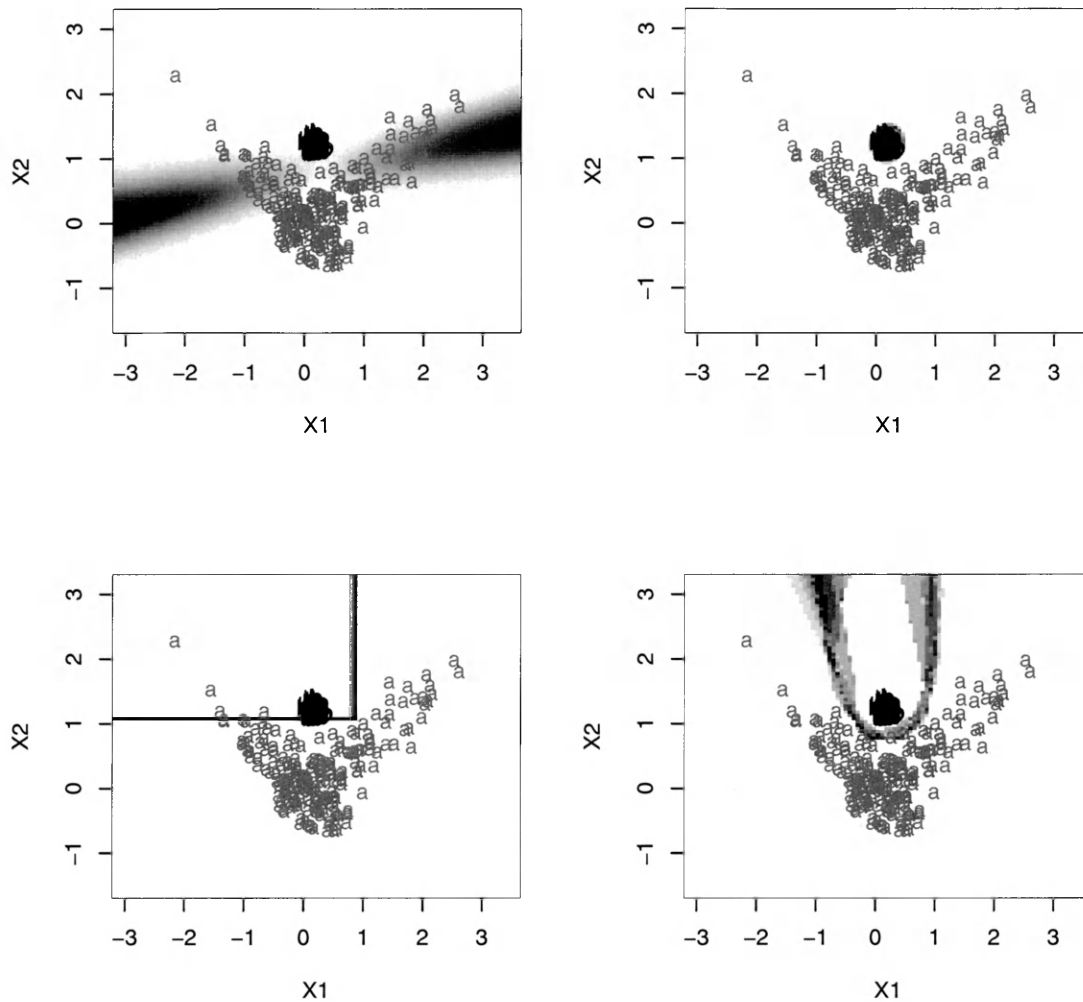


Figure 3.12: The data in this figure is *data.ng*, which has two groups of samples, each of them has 200 subjects. The first group is represented by letter *a* in the figure, second by letter *b*. Group *a* follows a normal distribution, group *b* has a special structure that could help us to better understand the different behaviors of linear and nonlinear algorithms. The classification algorithms are shown as: LDA on the top left, QDA on the top right, rpart on the bottom left and KNN on the bottom right. This figure reveals an important fact: some highly ambiguous regions can be very stable. Those points are classified with same group labels in the ensemble of classifiers consistently, but they are very difficult to classify as they are very close to the decision boundaries of different groups.

Chapter 4

Variable ranking and variable selection

Variable selection, sometimes called feature selection, is a common task in machine learning, classification, dimension reduction and linear regression studies. As we discussed in the introduction section of this thesis, when the number of variables is very large, we will have the so called *curse of dimensionality* problem. The computation time for a classification problem can be tremendous. Variable selection tries to find out a way to restrict or project the data to a low dimensional space that best represents the original high dimensional data. After the variable selection procedure, a data's dimension is reduced but, hopefully its structure is retained. It benefits researchers that:

1. The computation time can be dramatically decreased. Sometimes when the number of variables is extremely large, we may even not be able to solve the classification problem in a limited time. Variable selection makes solving such problems possible;
2. It helps researchers to better understand the underlying data structure. Ideally

when we are projecting the original data into low dimensional space, we have retained the important data structure information. For example, after we have found the most important 2 or 3 variables using their discriminating power, we can make a scatterplot in the 2 or 3 dimensional space, and visually see the data's distribution in those important dimensions. This job is almost impossible for data with higher than 3 dimensions.

One of the goals for early cancer detection study, as well as the mass-spectrometry research group in the College of William and Mary, is to find a few of the most important proteins that have significantly different relative intensities for people with cancer and people without cancer. A variable selection technique can help us to achieve this goal. In this thesis, we will discuss two different approaches to select the most important variables, we call them *variable ranking* and *optimal subset selection*, respectively.

4.1 Variable ranking

Variables ranking focuses more on searching the each individual variable's discriminant power. It can be measured in `rpart` as the decrease of Gini index for a variable to make a best split in a node on a decision tree.

4.1.1 Example: variable ranking using `rpart`

For `rpart`, we use the variable ranking criterion which was suggested by Leo Breiman *et al.*, in their book *Classification and Regression Trees*. The idea is that, at each node, decision tree tries all the variables exhaustively to find out which variable can best decrease a certain splitting rule (such as the misclassification rate or Gini index) by separating the samples in the original node into left and right sub nodes. Let's call the Gini index at node t as $I(t)$, after the splitting, node t is separated

into two disjoint smaller nodes, left node t_{left} and right node t_{right} . The decrease of the Gini index using variable V_i is defined in equation (4.1:)

$$\Delta(I(V_i, t)) = I(t) - p(t_{left})I(V_i, t_{left}) - p(t_{right})I(V_i, t_{right}) \quad (4.1)$$

where $p(t_{left})$ is the proportion of samples which have been separated into the left node, if the number of samples at node t before splitting is n_t and after splitting left node has $n_{t_{left}}$ samples, $p(t_{left}) \approx \frac{n_{t_{left}}}{n_t}$. Similarly for $p(t_{right})$ which $\approx \frac{n_{t_{right}}}{n_t}$.

And the importance for a variable V_i can be calculated as the sum of the maximum decrease of a certain criteria such as Gini index for that variable through all of the nodes:

$$M(V_i) = \sum_{t=1}^N \Delta I(V_i, t), N \text{ is the total number of nodes.} \quad (4.2)$$

By default `rpart` retains the top 5 variables instead of all variables' $\Delta(I(V_i, t))$ at each node, this can be changed by resetting the `surrogate` parameter in the `rpart` function. Note that according to this particular variable ranking method, some of the variables that even may not show up in the decision tree plot can be ranked among the top few most important variables. That's because they may be the second most important variables but the tree always select the best variable. For example, in Figure 1.2.2, *X5929.9* does not appear on the tree, but after summing up its decrease of Gini index power all through the tree, it ranks number two, which we can find from Figure 4.2.

The data used in Figure 4.1 and Figure 4.2 is the same. The data has two patients groups: *Adult T-cell Leukemia (ATL)* and *HTLV-1-associated-myelopathy/tropical spastic paresis (HAM/TSP)*, and one healthy group marked as *control*. The three groups have 42, 49, 38 subjects, respectively. Each subject has 60 variables that represent the relative abundance of various proteins from the subject's blood sam-

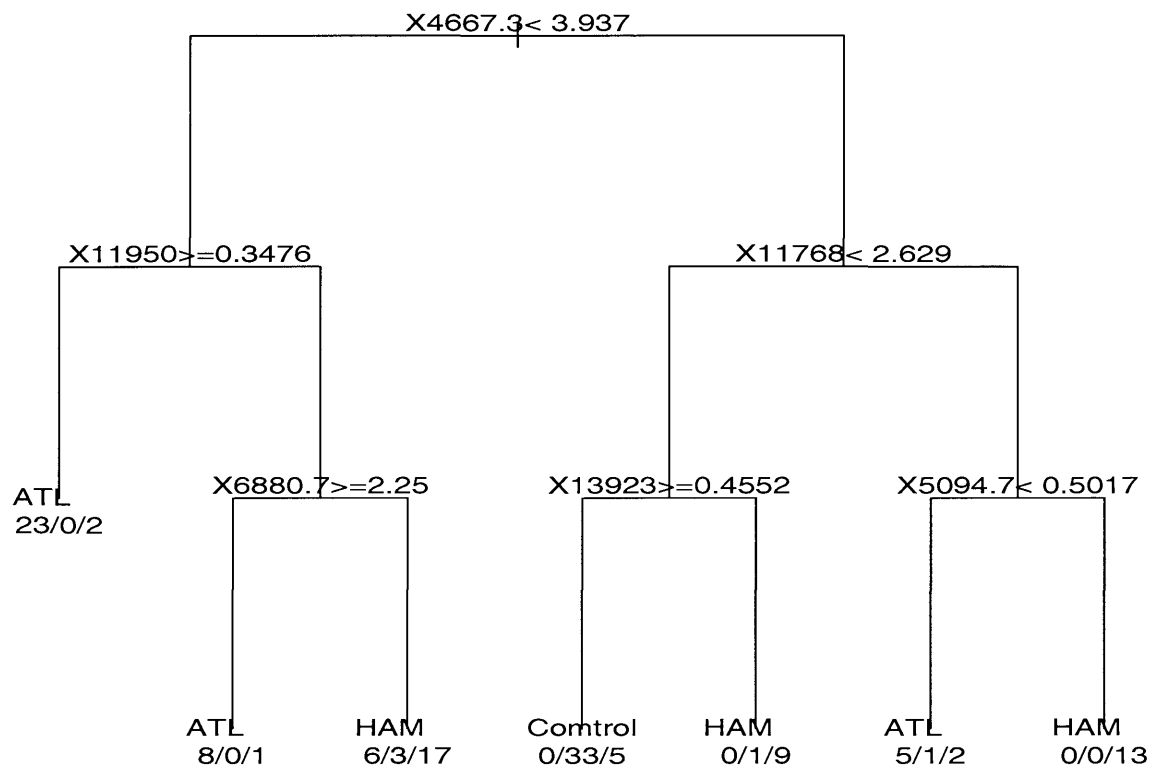


Figure 4.1: In this decision tree, 2002 Leukemia data is classified. The data contains two patients' groups samples: *ATL* (*Adult T-cell Leukemia*) and *HTLV-1-associated-myelopathy/ tropical spastic paresis* (*HAM/TSP*), and one group from healthy people marked as *control*. The three groups have 42,49,38 subjects, respectively. Each subject has 60 variables that represent the relative abundance of various proteins from people's blood sample. At the root node, rpart uses peak $X_{4667.3}$ and try to separate the three groups using threshold 3.937. Samples whose intensity at peak $X_{4667.3}$ less than 3.937 go to the left node, else go to the right node. The samples separated to the left are then checked by their intensities at peak X_{11950} to determine whether they are larger or equal than 0.3476. Then samples satisfy the inequality again go to the left and they are 23 ATL patients and 2 from HAM. The tree stopped there at the left decision block which we call a leaf. Totally there are 21 out of 129 samples misclassified. About 16 % nominal error rate. (The compute of nominal error rate is introduced in the introduction section of this thesis.)

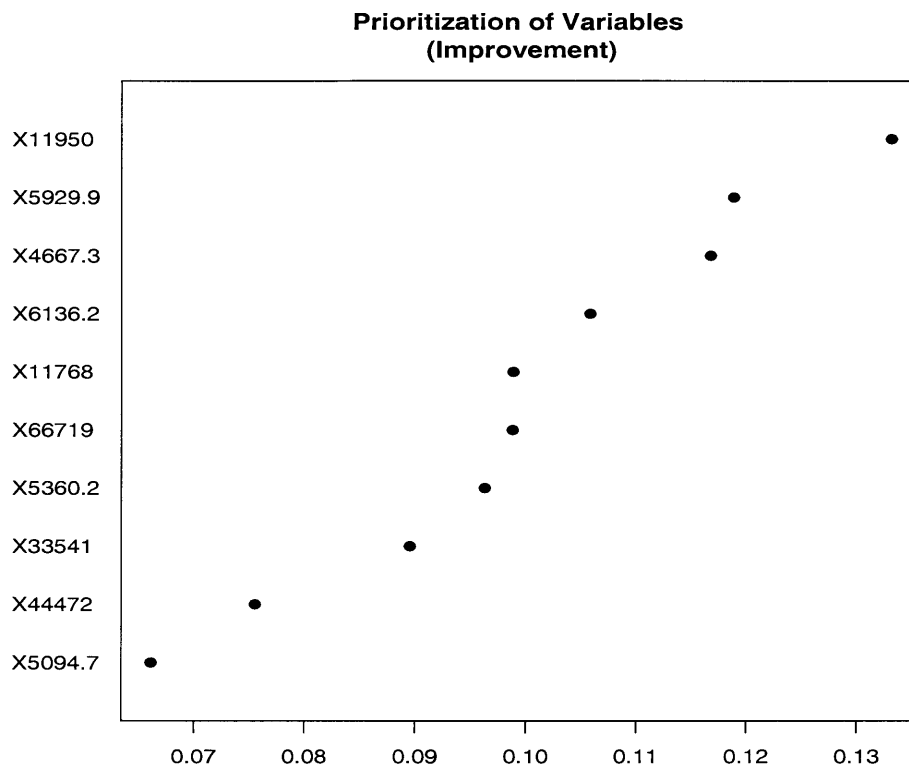


Figure 4.2: This is a dot plot which shows the top 10 most important variables using the ranking method with `rpart`. The names of the top 10 most important variables are plotted vertically, and their total decrease of the Gini index on all of the nodes is normalized and plotted in the horizontal axis in percentage. The most important variable is *X11950*, its discriminant power is about 13.5% from the top 10 most important variables, followed by variable *X5929.9*, about 12%

ple. The goal of this analysis is to find which variables have significantly different intensities in the three different groups.

Figure 4.1 shows that if we use the variables in the tree, we can separate the 129 samples into 7 more homogeneous decision blocks with about 84% accuracy, but what about the variables do not appear in the tree, are those variables not important?

Figure 4.2 is a dot plot showing the variables importance using the method discussed above, the names of the top 10 most important variables are plotted vertically, and their total decrease of the Gini index on all of the nodes is normalized and plotted in the horizontal axis in percentage. The most important variable is *X11950*, its discriminant power is about 13.5% from the top 10 most important variables, followed by variable *X5929.9*, about 12%.

The horizontal axis in the top 10 most important variables selected are plotted against their importance which is calculated by each variable's individual importance divided by the sum of their importance. The horizontal axis is the measure of their importance in percentage which sum up to 100% .

Figure 4.3 is the scatterplot using the two most important variables, *X11950* and *X5929.9*. In the plot, triangles are samples from HAM, circles are from healthy people, and the squares are from ATL. We can see that using the two variables can separate the groups of Leukemia (*ATL*, blue squares) and Healthy people (*control*, blue circles) well but the HAM/TSP group has lots of overlapping with the two groups and thus is very hard to separate.

The drawback of variable ranking is that this method is based on the discriminant power from a single variable, we have not considered the correlation between the important variables extracted. For example, *X11950* is the most important variable, and if we add a new variable which has the same value of *X11950*, the rankings of variables will be *X11950*, the new variable, and the rest of the variables. We have

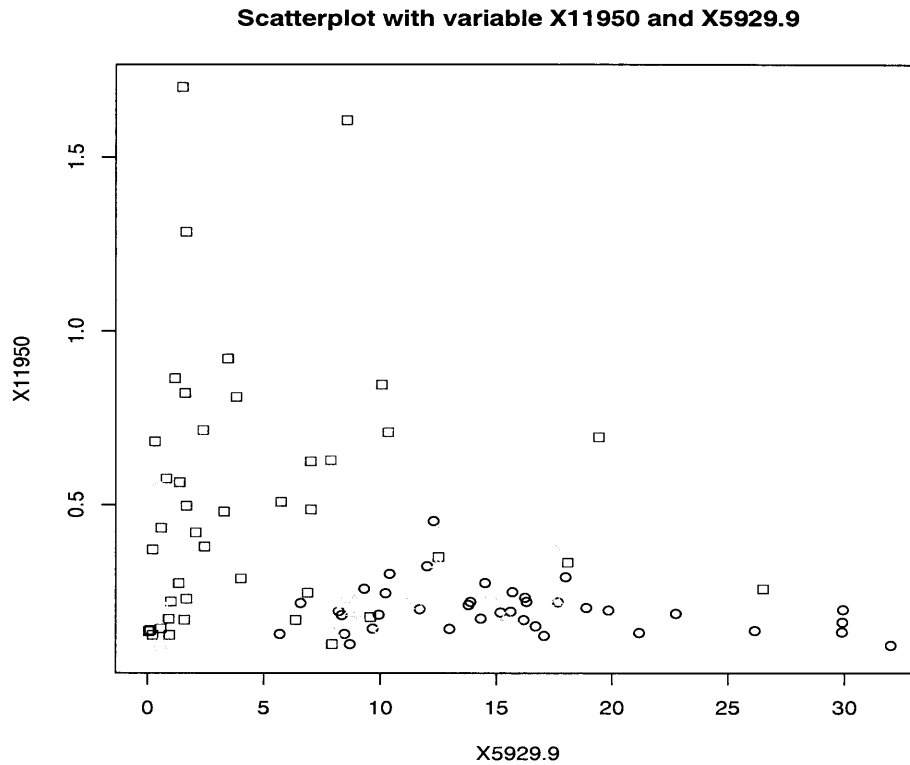


Figure 4.3: This is the scatterplot using the two most important variables, $X11950$ and $X5929.9$ which are chosen by rpart. In the plot, triangles are samples from *HAM/TSP*, circles are from healthy people, and the squares are from ATL. The two variables are good enough if we want to separate the ATL group from the control group well. The *HAM/TSP* group has lots of overlapping with the rest two groups. Samples from *HAM/TSP* are very hard to be isolated. We need other variables to separate HAM/TSP group.

not reduced the redundancy of important variables, so that if we classify the samples using the top 5 most important ranked variables, we may get very close results with using only 2 or 3 among the five. This problem urges us to consider the second approach, the optimal subset selection.

4.2 Optimum subset selection

Generally speaking, subset selection is a method which aims at finding out the best combination of variables for a certain criteria such as Wilks Lambda statistic. For example, we have a data set that includes two classes of samples, each of the samples has 100 variables. We want to choose five variables that have the best capacity of differentiating the two groups. The exhaustive way is to define a criteria such as the ratio of the between-group sum of square (B) with the sum of the between-group sum of squares and the within-group sum of squares (W). We can try all of the possible different combination of five variables and calculate the ratio respectively, then we find out which combination of 5 variables has the minimum value of λ .

$$\Lambda = |W|/|W + B| \tag{4.3}$$

When the number of variables is high, we can not calculate all the combinations exhaustively because the computation time required makes this job impossible. Instead, we use a heuristic method which can not promise always finding the best subset, but most of the time it can find out the most important variables which can discriminate the samples into different group with a small misclassification rate.

For a fixed number of variables, say, 10, an exhaustive way of finding the best subset of 10 variables is to try all of the combination of 10 different variables, computing their Λ , and find the ones with the minimum Λ . The computation time increases exponentially with the number of variables so that when the total number

of variables becomes very large, it is almost impossible for us to find out the best combination of 10 variables. Thus we need to find a heuristic method to choose 10 variables which has a Wilks Λ value comparable to the best answer.

4.2.1 Example: subset selection with McHenry's heuristic method

One way to find a heuristic computation method was introduced by McHenry in his paper *Computation of a Best Subset in Multivariate Analysis* [18]. According to McHenry, his method, "usually, but not invariably, finds the best solution". His strategy is to first select a subset of variables and then iteratively replace a single variable to a variable outside of the preselected set. And try to compare the two subsets' Wilks Λ value. If after change, the Wilks Λ becomes smaller, the new variable will replace the old one. The procedure does not stop until there is no improvement of the Wilks λ . Using this method, we can largely decrease the computation time if we exhaustively search all the combinations of variables for the best answer. The data Leukemia 2002 contains 60 quantitative variables and it takes only a few seconds to return the best 10 variables combination. The algorithm almost immediately outputs the answer when we are trying to find three best variables, which are: $X_{4490.7}$, $X_{8471.0}$ and X_{11768} . The experiments are done on a computer with a 2G RAM and 2 Intel(R) Core(TM) 1.8 GHz CPUs.

Scatterplot in Figure 4.4 uses two variables X_{11768} and $X_{8471.9}$. The squares represent samples from the ATL group, circles and triangles are samples from are samples from the control and HAM/TSP groups. Similar to the scatterplot in Figure 4.3, we can use a classifier to differentiate samples from ATL and control well, as they are well separated. HAM/TSP group has a big overlapping with ATL and control, which makes it almost impossible to differentiate it from the other two groups.

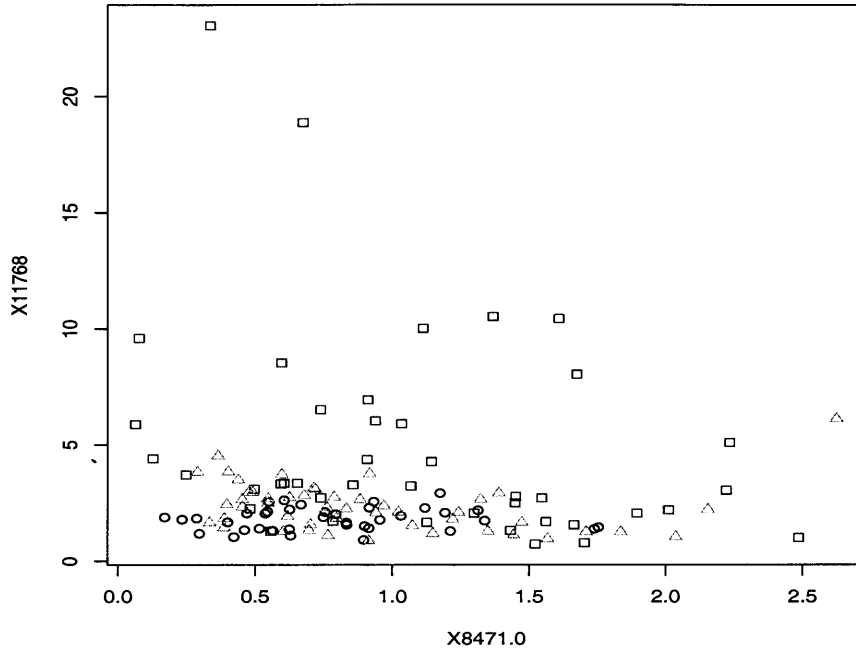


Figure 4.4: This is the scatterplot using the two most important variables, X_{11768} and $X_{8471.9}$ which are chosen by McHenry's Wilks Λ heuristic method. In the plot, triangles are samples from HAM, circles are from healthy people, and the squares are from ATL. Again, the two variables are good enough if we want to separate the ATL group from the control group well. But HAM/TSP group is very hard to be isolated.

Chapter 5

Visualization tool and VIBE-MS

5.1 Visualization of the decision boundaries

Decision boundaries for a classification problem draw special interest as they are comprised of special points with equal probability to be classified into different groups. Some of the methodologies for example: LDA and QDA are such that one can compute their decision boundaries analytically. But for algorithms like KNN, it is very hard or impossible to find the solutions for their decision boundaries analytically. The boundaries for KNN are quite irregular and non-smooth as we can see from Figure 5.1 and Figure 5.2, with two different data sets: *data.2g* and *data.4g*. Its decision boundaries are quite complicated in both figures, thus a method for visualization is useful.

5.1.1 Visualization algorithm

We suggest a decision boundary visualization algorithm in here:

we have not specified which particular classification technique in this algorithm. So it is applicable for any classification algorithms. We already showed some ex-

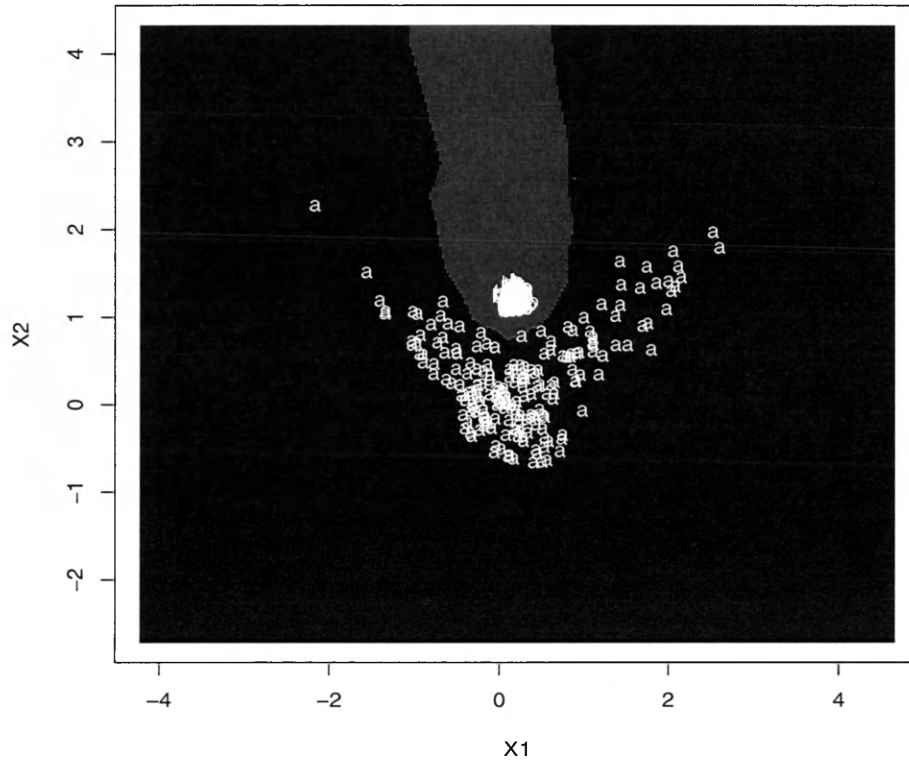


Figure 5.1: Some of the local decision boundaries of KNN classifier are quite irregular, non-smooth as we can see from this Figure. The data has two groups of sample. One has 200 samples represented by letter *a*, samples from this group together has a shape as a letter *V*. Second group is normally distributed which also has 200 samples.

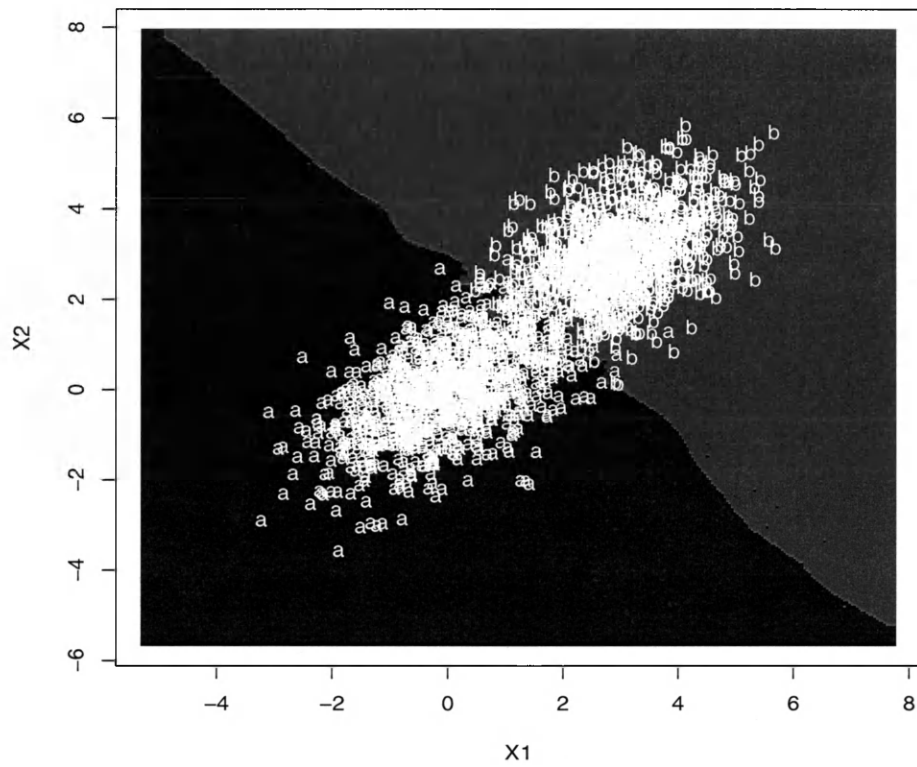


Figure 5.2: This figure again tells us that the decision Boundary of a KNN classifier can be very irregular and non smooth. The data we used to train the KNN classifier has two group of normally distributed samples, each group with 1000 samples.

Algorithm 1 Generate heatmap for decision boundaries

1. Build up a classifier C with the given training set X ;
 2. Select n_1 by n_2 points on a grid in the sample space. The grid should cover all the points from the training set. For example: we find two points: y_1 and y_n whose projections on Y-axis(vertical axis) are the minimum and the maximum among all the points in the data set. Similarly we find x_1 and x_m which are the minimum and the maximum in the horizontal direction. Then we restrict our grid to be within this square region: $(x_1 - \delta_1, x_m + \delta_2)$ by $(y_1 - \delta_3, y_n + \delta_4)$, where δ_i is a small positive real number, $i=1, 2, 3, 4$;
 3. Classify those points on the grid using the classifier C with deterministic method;
 4. Represent the points on the grid using a symbol with different colors according to their predicted group labels. The symbol's size should be large enough to cover the spaces among those points;
 5. Make a scatterplot of the training set against the plot using a different color, and we can see how the classifier separates the training set.
-

amples in section 1.4 when we are introducing the definition of decision boundaries using the four methods: recursive partitioning (rpart), K Nearest Neighbor (KNN), Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA).

Studying the decision boundaries by highlighting them in a plot gives us a lot of help in better understanding the behaviors of different classification algorithms. We have illustrated this point in Figure 1.4. Four algorithms' decision boundaries are highlighted with the same data set data.4g, which has four groups of normally distributed samples, and one group samples are well separated from the others.

It also helps us to realize the fact that even using the same classification method, the decision boundaries can change dramatically within an ensemble of classifiers. As we have shown in Figure 1.5: rpart's decision boundaries change a lot in a 5-fold cross-validation procedure. It gives us an idea of why the instability measure is important. Comparing Figure 1.5 to Figure 1.6, it is highly possible that when new samples are drawn from the unstable areas, their group assignments may be quite

different by rpart algorithm.

By looking at Figure 5.1 and Figure 5.2, we also find out that some of KNN's local decision boundaries are very unstable, and can be highly influenced by noise and outliers.

In summary, decision boundaries contain very important information: as it enables us to find out some of the basic nature of a classification technique. Building an ensemble of classifiers using the same classification technique, we can measure the decision boundaries' variance by ambiguity and instability and capture those unstable and ambiguous areas in the sample space.

5.2 VIBE-MS

As we discussed in the first section, cancer is one of the most deadly diseases in the US. Currently the best way to cure a cancer is to detect it in its early stage. Recent advances in proteomics have brought us opportunity to understand the possible solutions of cancer biomarkers in molecular levels, conventional life science knowledge alone is not enough. And knowledge from different approaches such as mass-spectrometry, signal processing, clinical research, computational science, physics, and statistics should be incorporated to achieve the goal of finding biomarkers and treat cancer in its early stages, which will enable us to detect a cancer efficiently and minimize the pain for the patients for testing.

Based on this motivation, Visual Integrated Bioinformatics System Mass Spectrometry (VIBE-MS) is created and developed cooperatively by Incogen, Inc., the William and Mary Mass Spectrometry group and Eastern Virginia Medical School.

The VIBE-MS workflow gives researchers an easy-to-control, integrated modular environment. The *drag-and-drop* interface allows researchers from different backgrounds to easily select what the data sets they want to analyze, in which way they

want the data to be preprocessed, and how to statistically analyze the data. VIBE-MS has provided a set of parameters that allows the researchers to choose their own. For examples, in the cross-validation module, user can select how many subsets they want to use in a cross-validation procedure; in KNN classification module, they can choose a value for K .

The goal of VIBE-MS is to integrate new and existing methods into a whole pipeline that makes a complete process from raw data preprocessing to the classifier evaluation. Signal processing and statistical analysis workflows are integrated into a whole pipeline. First the raw data (in the form of a mass-spectrum) is selected using a spectrum selector module and then the optimization modules starts to work, then the processed data sets are sent to statistical classification modules to build classifiers, finally those classifiers are evaluated by the cross-validation module.

5.2.1 Classification tools in VIBE-MS

In cooperation with computer scientists at Incogen, this thesis' author has incorporated several classification modules and evaluation tools into VIBE-MS. Each classification module has a cross-validation mode and non-cross-validation mode. The non-cross-validation mode uses all of the data in training and testing the classifier on all of the data, whose pros and cons have been discussed in the first section. In the cross-validation mode, two types of the most popular cross-validation algorithms are available, and they are:

1. V -fold cross-validation. The original data set is partitioned into V disjoint subsets, each time $V-1$ subsets are randomly selected as training data, the classification model is built on it, then the remaining single subset is used for testing. This process is repeated V times until all of the samples are tested exactly once. To minimize the variation of the partitioning, the whole V -fold cross-validation should be repeated N times.

2. Leave-one-out cross-validation. In a leave-one-out cross-validation procedure, a single sample is extracted as a test set, the other samples are used as training, to make sure the independence of training sets and test sets, if a test sample has more than one replicates, the training set does not contain its other replicates.

Four classification algorithms: LDA, QDA, rpart and KNN are integrated into these cross-validation packages. After the cross-validation procedure, the posterior probabilities for each sample will be reported. Visualization tools such as the heatmap can also be used on posterior probabilities to show important information such as the group assignments of those samples.

After each run, classifiers (statistical models) and important variables and certain criterion such as misclassification rate are output to help researchers to understand and evaluate the whole process. The flexibility of VIBE-MS pipeline enables users to change the parameters and finally achieve the optimum results.

What's more important, the R code for calculating and visualizing ambiguity, instability and decision boundaries can be integrated into VIBE-MS's workflow. By simply copying and pasting the codes into a *Generic R* module, researchers will be able evaluate a classification technique's performance using the two new metrics: ambiguity and instability we introduced in this thesis. Some of the R codes are attached in the appendix section.

Figure 5.3 shows a VIBE-MS mass-spectra pipeline. It includes a data selection module (the *Spectra* square on the top left of the workspace) to obtain data, then the data is preprocessed using background subtraction and several other signal processing methods, then most important variables are selected using variable selection module, finally the data is passed to a generic R module on the right bottom of the workspace to build classifiers and have them evaluated. Two textview modules are used to record important information such as the misclassification rate of different classification algorithms.

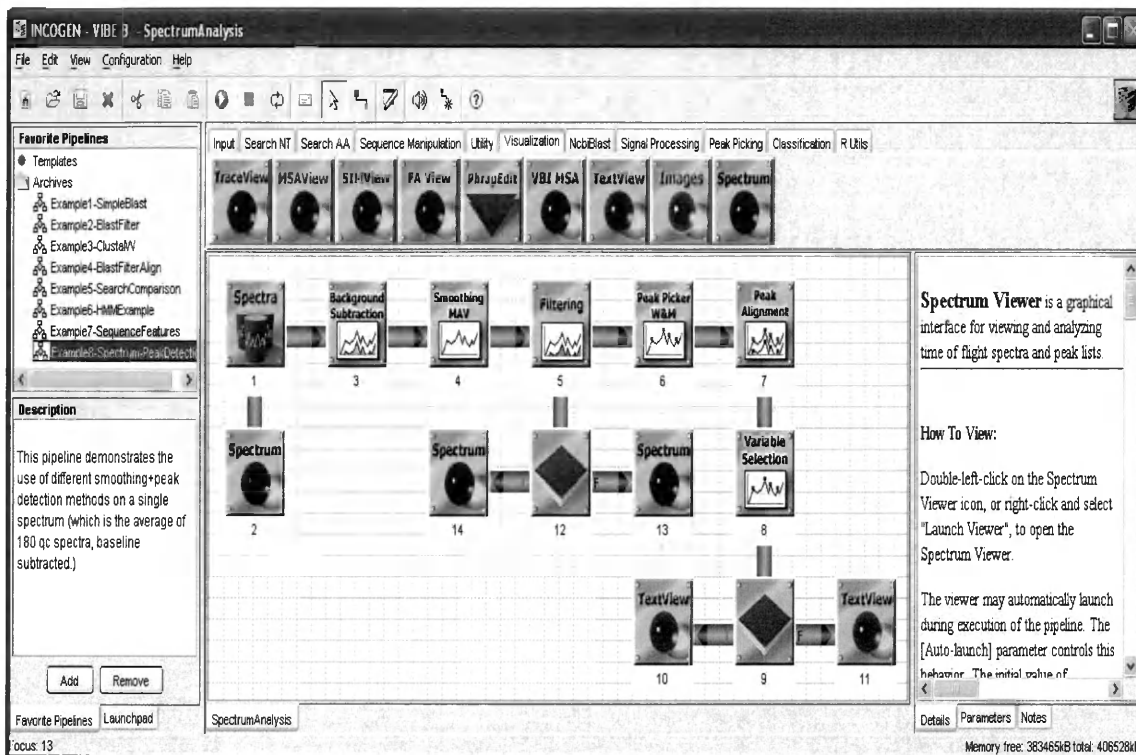


Figure 5.3: This figure shows a VIBE-MS mass-spectra pipeline, it contains a data selection module (the Spectra square on the top left of the workspace) to obtain data, then the data is preprocessed using background subtraction and several other signal processing methods, then most important variables are selected using variable selection module, finally the data is passed to a generic R module on the right bottom of the workspace to build classifiers and have them evaluated, two textview modules are used to record important information such as the misclassification rate of different classification algorithms.

Chapter 6

Conclusions

From section 1 to section 3 we have discussed the motivations of using ambiguity and instability as two evaluation criteria together with error rate. The numerical experiments in section 3 suggest that both of the new definitions: ambiguity and instability can help us in evaluating a classification technique. They give deeper information which helps us better judge a classifier than conventional methods such as error rate. They benefit us in:

1. Better understand how the decision boundaries change within an ensemble of classifiers, unstable and ambiguous regions can be found in a straightforward manner where the classifier's outputs are generated by different cross-validation random partitioning.
2. Classifiers with comparable error rates can be quite different in instability measurement. When we have future samples need to be classified, we should choose the classification technique with smaller instability measurement, and error rate should not be the only criterion when we are selecting a classification technique. This conclusion is supported by Table 3.1 and Figure 3.6, 3.9 and 3.12.
3. Figure 3.4, 3.5, 3.7, 3.8, 3.10 and 3.11 suggest that some regions in the sample

space maybe very ambiguous, which means points from a probabilistic distribution in there are very hard to differentiate from other distributions thus they should be removed from classifying. But they can be very stable in the cross-validation process.

4. Table 3.3 indicates that ambiguity can help us finding *risky* points which are not appropriate to be classified, as the information Bayes error contains. Ambiguity can be used as a substitute of Bayes error. And what's more important, analytically ambiguity measurement has better properties than Bayes error. Ambiguity function is a smooth function in the sample space, while Bayes error is not smooth in the local maximum points.

Chapter 7

Appendix: R code

Part of the R code used in this thesis are shown below.

To use these R code, the data must have such strict structure:

1. The first column is the subject identifier, such as 1,2,..., N (N is the total number of different subjects)
2. The second column must be the replicate identifier, which differentiates the same subject.
3. The third column is the group identifier, to differentiate subjects from different groups.
4. From the fourth column to the last column are the numeric variables.

7.1 R code to plot the decision boundaries

The following codes are written in R(a statistical computing language) and are used to generate the decision boundaries for : KNN and LDA.

```
library(MASS)
```

```

library(class)

#####

# This function plots the decision boundaries using KNN algorithm
f.knn.db<- function(Train,Test,Labels,K=5)
{
  #First column is the subject id,
  #Second column is the replication id,
  #Third column is the group id

  group.id<-as.character(Train[,3])
  X<-as.numeric(Train[,4]) # X1
  Y<-as.numeric(Train[,5]) # X2
  Xlim<-c(min(c(X,Test[,4]))-2, max(c(X,Test[,4]))+2)
  Ylim<-c(min(c(Y,Test[,5]))-2, max(c(Y,Test[,5]))+2)
  x.plot<-seq(Xlim[1],Xlim[2],length=100)
  y.plot<-seq(Ylim[1],Ylim[2],length=100)
  x.plot<-sort(rep(x.plot,100))
  xy.data<-data.frame(X1=x.plot,X2=y.plot)
  Labels<-knn(train=Train[,-c(1:3)], test=xy.data, cl=group.id,k=K)
  plot(as.numeric(xy.data[,1]),as.numeric(xy.data[,2]),
       col=as.numeric(as.factor(Labels)),
       pch=rep(15,nrow (xy.data)),xlab='X1',ylab='X2' )
  points(as.numeric(Train[,4]),as.numeric(Train[,5]),
        pch=as.character(group.id),col='white')
}

```

```
#####
# This function plots a decision boundary using lda algorithm
f.lda.db<- function(Train,Test,Labels,K=5)
{
  group.id<-as.character(Train[,3])
  X<-as.numeric(Train[,4]) # X1
  Y<-as.numeric(Train[,5]) # X2
  Xlim<-c(min(c(X,Test[,4]))-2, max(c(X,Test[,4]))+2)
  Ylim<-c(min(c(Y,Test[,5]))-2, max(c(Y,Test[,5]))+2)
  x.plot<-seq(Xlim[1],Xlim[2],length=100)
  y.plot<-seq(Ylim[1],Ylim[2],length=100)
  x.plot<-sort(rep(x.plot,100))
  xy.data<-data.frame(X1=x.plot,X2=y.plot)
  Fit <- lda(as.factor(group.id)~.,Train[,4:5])
  Labels<-predict(Fit,xy.data)$class
  plot(as.numeric(xy.data[,1]),as.numeric(xy.data[,2]),
       col=as.numeric(as.factor(Labels)),
       pch=rep(15,nrow (xy.data)),xlab='X1',ylab='X2' )
  points(as.numeric(Train[,4]),as.numeric(Train[,5]),
        pch=as.character(group.id),col='white')
}
```

7.2 R code to plot the heatmaps of the ambiguous and unstable regions

Below are the R code used to draw ambiguity and instability areas using rpart algorithm. The parameter *method* specifies whether it is a probabilistic model or a deterministic model, the *measure* can be either *ambi* or *instab*, which determines it is a ambiguity plot or instability plot.

```
f.plot.rpart <-  
function(Data, V=5, no.cv=10,no.p=100, measure='ambi',method='prob')  
{  
#####  
# Data has such structure:  
# 1st column as subject id, 2nd column rep id, 3rd column group id,  
# 4th column is the first variable, 5th column the second variable,  
# only two dim data sets are applicable for this function.  
# 'no.p' is the number of points selected # horizontally or  
# vertically in the grid, by default,  
# 10,000 points are selected.  
  
#####  
# Create the points on the grid  
  X<-as.numeric(Data[,4])  
  Y<-as.numeric(Data[,5])  
  Xlim<-c(min(X)-1, max(X)+1)  
  Ylim<-c(min(Y)-1, max(Y)+1)
```

```

x.plot<-seq(Xlim[1],Xlim[2],length=no.p)
# x values on the grid, used in drawing the heatmap
X.grid <- x.plot
y.plot<-seq(Ylim[1],Ylim[2],length=no.p)
Y.grid <- y.plot # same as X.grid
x.plot<-sort(rep(x.plot,no.p))
xy.data<-data.frame(X1=x.plot,X2=y.plot)

#####

library(rpart)
Nrow <- nrow(xy.data)
group.id <- Data[,3]
no.group <- length(levels(as.factor(group.id)))
all.groups <- levels(as.factor(group.id))
sub.id <- as.character(Data[,1])

#####

f.binary<-
function(x)
{
Max<-which.max(x)
x[Max]<-1
x[-Max]<-0
return(x)}

#####

f.geo.sub<-

```



```

function(sub.p){
  return(sum(sub.p*(1-sub.p))/2) }

#####

f.liang.edist<-
  function(x,Y,no.group)
  {
    X<-matrix(rep(x,length(Y)/no.group),byrow=TRUE,
              nrow=length(Y)/no.group)
    X1<-apply((X-Y)^(2),1,sum)
    Edist<-sum(sqrt(X1))
    return(Edist)
  }

#####

f.liang.edist.2 <-
  function (Data,no.group=2,V=V,Const=Const)
  {
    Dist<-numeric()
    for(i in 1:(nrow(Data)-1)){
      X<-Data[i,]
      Y<-Data[-c(1:i),]
      Dist[i]<-f.liang.edist(X,Y,no.group)
    }
    return(sum(Dist)/Const)
  }

```

```
#####
```

```
f.plot.rpart.2<-function(Data,V=5,xy.data=xy.data,no.p=no.p)
{
  cv.part <- numeric()
  Prob.matrix <- array(0, dim=c(Nrow, no.group, V))
  for(i in 1: no.group)
  {
    reps <- which( is.element( group.id,all.groups[i] ) )
    # the subject name
    subjects <- levels (as.factor(sub.id[reps] ) )
    # number of different subjects
    no.sub <- length(subjects)
    r <- round(no.sub/V)
    Res<- r*V- no.sub
    r.interval <- c(rep(r, V-abs(Res)), rep(r-sign(Res), abs(Res)) )
    r<-rep(0, ( length(r.interval)+1 ) )
    r[1]<-0
    for(i in 2: (length(r.interval)+1))
    {
      r[ i ] <- r[i-1]+r.interval[i-1]
    }
    perm <- sample(1:no.sub)
    for ( j in 1: V)
    {
      subj.j <- perm[(r[j]+1) : r[j+1] ]
      reps <- which(is.element(sub.id, subjects[subj.j]))
    }
  }
}
```

```

    cv.part [reps] <- j
  }
}
for (i in 1:V) {
  data.test <- xy.data
  data.train <- Data[cv.part!=i, ]
  fit.full <- rpart(as.factor(data.train[,3])~.,
                  data.train[,-c(1:3)] )
  # Check whether it is a root tree,
  # otherwise prune it back using 1 se rule.
  fit.cptable <- fit.full$cptable
  if(ncol(fit.cptable)>=5){
    min.error.index<- which.min( fit.cptable [,4] )
    one.se <- sum (fit.cptable[min.error.index, 4: 5] )
    cp.index <-as.numeric( names(fit.cptable[,4][ fit.cptable [,4] <
                                                one.se] [1] ) )
    fit.prune <- prune(fit.full, cp = fit.cptable[cp.index])
  } else {
    fit.prune<-fit.full
  }
  Prob.matrix[, ,i]<-predict(fit.prune,data.test,type='prob')
}
return(Prob.matrix)
}

```

```
#####
```

```
# Main function
```

```

#Convert character into position of the group id
index.group.id<-as.numeric(as.factor(group.id))
Const<-(no.cv)^(2)*sqrt(2)
X<-array(0,dim=c(no.p*no.p,no.group,no.cv*V))
for(i in 1:no.cv){
  X[,,( (i-1)*V+1):(i*V) ]<-f.plot.rpart.2(Data=Data,V=V,
                                             no.p=no.p,xy.data=xy.data)
}
if (measure=="ambi") {
  ambi<-numeric( )
  for( i in 1: (no.p*no.p)){
    Y<-t(matrix(as.numeric(X[i,1:no.group,]),nrow=no.group))
    # calculate the average of the posterior probability for
    # sample 'i'
    if (method=="det")
      { Y<-t(apply(Y,1,f.binary)) }
    # f.binary is another function which round a number
    # between (0,1) to the 0 or 1 according to which
    # is closer.
    ProbBar<-apply( Y, 2 ,mean)
    ambi[i]<- f.geo.sub(ProbBar)
  }
  Measure <- ambi
}

if(measure=="instab"){
  instab<-numeric()
}

```

```

for(i in 1: ( no.p*no.p) ){
  Y<-t(matrix(as.numeric(X[i,1:no.group,]),nrow=no.group))
  instab[i]<-f.liang.edist.2(Y,no.group=no.group,V=V,Const=Const)
  # f.liang.edist.2 is a function calculate the euclidean distance
}
Measure <- instab
}

image(X.grid, Y.grid, matrix(Measure,nrow=no.p,byrow=TRUE),
      xlab='X1',ylab='X2',col=gray(256:1/256))
points(Data[,4:5], col=(2+as.numeric(as.factor(Data[,3]))),
       pch=(2+as.numeric(as.factor(Data[,3]))))
}

```

7.3 R code to compute the global ambiguity, instability and error rate

The following R code calculates the global ambiguity, instability or error rate for QDA algorithm.

```

f.comp.qda <- function(Data,V=5,no.cv=10,method='prob')
{
#####
f.binary<-

```

```

function(x)
{
Max<-which.max(x)
x[Max]<-1
x[-Max]<-0
return(x)}

#####

f.geo.sub<-
function(sub.p){
return(sum(sub.p*(1-sub.p))/2) }

#####

f.liang.edist<-
function(x,Y,no.group)
{
X<-matrix(rep(x,length(Y)/no.group),
byrow=TRUE,nrow=length(Y)/no.group)
X1<-apply((X-Y)^(2),1,sum)
Edist<-sum(sqrt(X1))
return(Edist)
}

#####

f.liang.edist.2 <-
function (Data,no.group=2,V=V,Const=Const)
{
Dist<-numeric()

```

```

    for(i in 1:(nrow(Data)-1)){
      X<-Data[i,]
      Y<-Data[-c(1:i),]
      Dist[i]<-f.liang.edist(X,Y,no.group)
    }
    return(sum(Dist)/Const)
  }

```

```
#####
```

```

f.comp.qda.2<-
function(Data,V)
{
  library(MASS)
  Nrow <- nrow(Data)
  Data<- data.frame(Index=(1:Nrow),Data)
  Data[,4]<-as.character(Data[,4])
  group.id <- Data[,4]
  no.class <- length(levels(as.factor(group.id)))
  all.groups <- levels(as.factor(group.id))
  sub.id <- as.character(Data[,2])
  no.group<-length(levels(as.factor(group.id)))
  cv.part <- numeric()
  for(i in 1: no.class)
  {
    reps <- which( is.element( group.id,all.groups[i] ) )
    # the subject name
    subjects <- levels (as.factor(sub.id[reps] ) )

```

```

# number of different subjects
no.sub <- length(subjects)
r <- round(no.sub/V)
Res<- r*V- no.sub
r.interval <- c(rep(r, V-abs(Res)),
rep(r-sign(Res), abs(Res)) )
r<-rep(0, ( length(r.interval)+1 ) )
r[1]<-0
for(i in 2: (length(r.interval)+1))
  {
    r[ i ] <- r[i-1]+r.interval[i-1]
  }
perm <- sample(1:no.sub)
for ( j in 1: V)
  {
    subj.j <- perm[(r[j]+1) : r[j+1] ]
    reps <- which(is.element(sub.id,
                           subjects[subj.j]))
    cv.part [reps] <- j
  }
}
phase2.qda.prob <- rep(0, no.group+1)
for (i in 1:V)
  {
    data.test <- Data[cv.part==i, ]
    data.train <- Data[cv.part!=i, ]
    fit.qda <- qda(as.factor(data.train[,4])~.,

```



```

        data.train[,-c(1:4)] )
    X<-predict(fit.qda,data.test[,-c(1:4)])$posterior
    phase2.qda.prob <- rbind(phase2.qda.prob,
                             cbind(data.test[,1],X))
  }
  phase2.qda.prob <- phase2.qda.prob[-1,]
  phase2.qda.prob <- phase2.qda.prob[order(
    as.numeric(phase2.qda.prob[,1])),][,-1]
  return(phase2.qda.prob)
}

```

```
#####
```

```
#Main function
```

```

  Nrow<-nrow(Data)
  group.id<-as.character(Data[,3])
  no.group<-length(levels(as.factor(group.id)))
  #Convert character into position of the group id
  index.group.id<-as.numeric(as.factor(group.id))
  Const<-(no.cv)^(2)*sqrt(2)
  X<-array(0,dim=c(Nrow,no.group,no.cv))
  for(i in 1:no.cv) {
    X[, ,i]<-f.comp.qda.2(Data=Data,V=V)
  }

```

```
#####
```

```
#Calculate the Bayes error
```

```
Y<-apply(X,1:2,mean)
```

```

B.err<-numeric()

index.group.id.2<-apply(Y,1,which.max)

for(i in 1:Nrow) {
  B.err[i] <- 1- Y[i,][index.group.id.2[i]]
}

B.err<-sum(B.err/Nrow)

#####

# Calculate the instability, the instability under
# deterministic method has the same value of ambiguity,
# so we only compute the ambiguity for simplicity.

instab<-numeric()

for(i in 1: Nrow){
  Y<-t(matrix(as.numeric(X[i,1:no.group,]),nrow=no.group))
  instab[i]<-f.liang.edist.2(Y,no.group=no.group,
                           V=V,Const=Const)
}

instab<-sum(instab)/Nrow

ambi<-numeric()

ambi.det<-numeric()

err<-numeric()

for( i in 1: Nrow){
  Y<-t(matrix(as.numeric(X[i,1:no.group,]),
              nrow=no.group))
  ProbBar<-apply( Y, 2 ,mean)

```

```

err[i]<- 1-ProbBar[index.group.id[i]]
ambi[i]<- f.geo.sub(ProbBar)
Y1<-t(apply(Y,1,f.binary))
ProbBar1<-apply( Y1, 2 ,mean)
ambi.det[i]<-f.geo.sub(ProbBar1)
}

err<-sum(err)/Nrow
ambi<-sum(ambi)/Nrow
ambi.det<-sum(ambi.det)/Nrow
return( list ( B.err=B.err, err=err,
               instab=instab, ambi=ambi,
               ambi.det=ambi.det ) )
}

```

Chapter 9

References

- [1] Mardia, K., Kent, J. and Bibby, J. (2006). *Multivariate Analysis*, Academic Press.
- [2] Hastie, T., Tibshirani, R., Friedman, J. (2003). *The Elements of Statistical Learning*, Springer.
- [3] Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classification And Regression Tree*, Chapman & Hall/CRC.
- [4] Salford Systems, (November, 2007). <http://www.salford-systems.com/cart.php>.
- [5] Atkinson and E., Thernequ, T. (1997). An Introduction to Recursive Partitioning Using the RPART Routines, *Technical report*, Mayo Clinic Section of Biostatistics.
- [6] Hornik, K. (2007). *The R FAQ*, <http://cran.r-project.org/doc/FAQ/R-FAQ.html>, ISBN 3-900051-08-9.
- [7] Everitt, B. (2004). *An R and S-PLUS Companion to Multivariate Analysis*, Springer.

- [8] Venables, W. and Ripley, B. (2002). *Modern Applied Statistics with S*, Fourth Edition, Springer.
- [9] Cancer Facts and Figures 2007 (2007). Atlanta: American Cancer Society.
- [10] Tracy, M., Chen H., Malyarenko, D., Sasinowski, M., Cazares, L., Drake, R., Semmes, O., Tracy, E. and Cooke, W. (2007). Precision enhancement of MALDI-TOF-MS using high resolution peak detection and label-free alignment, to appear in *Proteomics*.
- [11] Chen, H., Tracy, E., Cooke, W., Semmes, O., Sasinowski, M., and Manos, D. (2007). Automated peak identification in a TOP-MS spectrum, in *Quantitative medical data analysis using mathematical tools and statistical techniques*, Hong, D. and Shyr, Y., World Scientific, Singapore.
- [12] Davis, C., Gerick, F., Hintermair, V., Friedel, C., Fundel, K., Kuffner, R. and Zimmer, R. (2006). Reliable gene signatures for microarray classification: assessment of stability and performance, *Bioinformatics*, **22**: 2356-2363.
- [13] Aebersold R., Goodlett D. (2001). Mass spectrometry in proteomics, *Chemical Review*, **101**: 269-95.
- [14] Gygi S., Aebersold R. (2000). Mass Spectrometry and proteomics, *Current Opinion in Chemical Biology*. **4**: 489-94.
- [15] Yates 3rd JR (2000). Mass spectrometry From genomics to proteomics, *Trends*

Genet, **16**: 5-8.

[16] Keller, B., Li, L. (2000) Discerning matrix-cluster peaks in matrix-assisted laser desorption/ionization time-of-flight mass spectra of dilute peptide mixtures. *J Am Soc Mass Spectrum*, **13**:129-34.

[17] Shin, H., Markey, M. K., (2005). A machine learning perspective on the development of clinical decision support systems utilizing mass spectra of blood samples, *Journal of Biomedical Informatics*, 227-248.

[18] McHenry, C., (1978). Computation of a Best Subset in Multivariate Analysis, *Applied Statistics* , **27**: 291-296.

[19] Handl, J., Knowles, J. and Kell, D. (2005). Computational cluster validation in post-genomic data analysis, *Bioinformatics*, **21** 3201-3212.

[20] Quinlan, R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo.

[21] Turner, K. and Ghosh, J. (2000). Robust combining of disparate classifiers through order statistics, in H. Kargupta and P. Chan, editors, *Advances in Distributed and Parallel Knowledge Discovery*, pp. 185-210, AAAI/MIT Press.

[22] K. Turner and J. Ghosh (1996). Analysis of decision boundaries in linearly combined neural classifiers, *Pattern Recognition*, **29**: 341-348

[23] K. Turner and J. Ghosh (2003). Bayes Error Rate Estimation using Classifier En-

sembles, *International Journal of Smart Engineering System Design*, **5(2)**: 95:110.

[24] Devijver, P and Kittler, J. (1982). *Pattern Recognition: a Statistical Approach*, Prentice/Hall, New Jersey.

[25] Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*, Academic Press, New York.

[26] Han., J. and Kamber, M. (2006). *Data Mining: Concepts and Techniques*, Elsevier, New York.