Robust Nonparametric Discriminant Analysis Procedures

Except where reference is made to the work of others, the work described in this dissertation is my own or was done in collaboration with my advisory committee. This dissertation does not include proprietary or classified information.

Sai Vamshidhar Nudurupati

Certificate of Approval:

Nedret Billor
Associate Professor
Mathematics and Statistics

Asheber Abebe, Chair
Associate Professor
Mathematics and Statistics

Peng Zeng
Assistant Professor
Mathematics and Statistics

George T. Flowers
Dean
Graduate School

Robust Nonparametric Discriminant Analysis Procedures

Sai Vamshidhar Nudurupati

A Dissertation

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Doctor of Philosophy

Auburn, Alabama
May 9, 2009

Robust Nonparametric Discriminant Analysis Procedures

Sai Vamshidhar Nudurupati

Signature of Author

Date of Graduation

Vita

Sai Vamshidhar Nudurupati, son of Sri. Sai Baba Nudurupati and Smt. Sarada Nuduruapti, was born on April 8, 1980, in Visakhapatnam, India. He graduated from Little Flower Junior College, Hyderabad, India in May 1997. He joined MJCET, Osmania University, Hyderabad, India in November 1997 and graduated with a Bachelor of Engineering in Mechanical Engineering in May 2001. He graduated with a Masters in Industrial Engineering from Arizona State University in May 2003. He joined the doctoral program in Department of Mathematics and Statistics at Auburn University in January 2004 under the guidance of Dr. Asheber Abebe. He is married to Sheetal Paliwal, daughter of Sri. Hansraj Paliwal and Smt. Savita Paliwal.

Dissertation Abstract

Robust Nonparametric Discriminant Analysis Procedures

Sai Vamshidhar Nudurupati

Doctor of Philosophy, May 9, 2009
(M.S., Arizona State University, 2003)
(B.S., MJCET - Osmania University, 2001)

130 Typed Pages

Directed by Asheber Abebe

In this study, a nonparametric discriminant analysis procedure that is less sensitive than traditional procedures to deviations from the usual assumptions is proposed. The procedure uses the projection pursuit methodology where the projection index is the two-group transvariation probability. Montanari (2004) proposed and used this projection index to measure group separation but allocated the new observation using simple Euclidean distances from projected centers. Our procedure employs a method of allocation based on the centrality of the new point measured using two versions of the transvariation probability: a symmetrized two-group transvariation and a smooth version of point-group transvariation. It is shown by simulation that the procedures proposed in this study provide lower misclassification error rates than classical procedures such as linear discriminant analysis and quadratic discriminant analysis and recent procedures like maximum depth and Montanari's transvariation-based classifiers under a variety of distributional settings. A different rank-based procedure for classification is considered where ranking is applied on classical classifiers as well as recently introduced classifiers such as the maximum L_1 depth and quadratic discriminant function based on the minimum covariance determinant (MCD) estimates of the mean and covariance. An extensive simulation study shows that not only does the ranking method provide balance between misclassification error rates for each group but also yields lower total probabilities of misclassification and higher consistency of correct classification for heavy-tailed distributions. A theoretical evaluation of the influence function shows that this new procedure is robust against local infinitesimal contaminations.

Acknowledgments

I would like to express my earnest gratitude to Dr. Asheber Abebe for his continuous support and guidance throughout the course of my degree. I would like to acknowledge my committee members, Dr. Nedret Billor and Dr. Peng Zeng for their valuable time and support. My sincere gratitude to Dr. Curtis Jolly for acting as an external reader. I am deeply indebted to my family, who went far beyond their normal boundaries to provide me with all comforts. I thank them for all the sacrifices they have made and their devotion. The unconditional love and support of my siblings, Kiran and Chaitanya helped me accomplish this degree. My parents-in law, deserve a special mention for their immense support and affection. Words fail to express my appreciation to my wife, Sheetal, whose commitment and persistence have helped me realize my aspirations. Her endless love and patience kept me focused and always steered me in the right direction. I consider myself fortunate to have her as my life partner. Thanks to one and all. Finally, thanks to Lord Sri Sairam for his grace and blessings.

Style manual or journal used Journal of Approximation Theory (together with the style known as "aums"). Bibliograpy follows van Leunen's *A Handbook for Scholars.*

Computer software used The document preparation package TeX (specifically LaTeX) together with the departmental style-file `aums.sty`.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Discriminant analysis is the process of devising rules to assign a new individual data point into one of K ($K > 1$) known groups. The method is usually based on previously known information related to the K groups, known as *training data* whose correct classification information is known. In this dissertation, the focus would be on the two-group discrimination problem ($K = 2$). Classification or discrimination has a wide range of applications. A very small list of applications includes: spam filters for an email engine that sends good emails to the inbox and bad emails to a spam folder; voice recognition software used to distinguish the source of the voice from among several speakers; methods to distinguish who is a bad risk for credit and who is credit worthy; methods to classify a patient's tumor as cancerous or benign.

A discriminant analysis procedure uses all the variables that the training data contains and use their correct classification information to create a discriminant model, a discriminant rule or a classifier. Classification is done by feeding new observations into this classifier or model and getting the group membership to which the new observation belongs. A classifier is said to be a good classifier if it provides low misclassification error rates not just for the best of the situations but also under various conditions such as shape of the groups, number of groups, size of the groups etc. Fisher (1936) came up with what is considered to be the first scientific discriminant model. His classifier performs well for data that follow normal distributions, the groups share the same covariance structure and if there are no outliers. Violations of any of these assumptions may make Fisher's method unstable. There has been a lot of research done in the field of classification ever since then in an effort to come up with classifiers that are robust, ones that are not sensitive to violations of certain assumptions. A wide range of techniques have been used to develop various such classifiers that are robust to deviations in the parametric, semi-parametric and nonparametric realms. Most of the parametric methods become sensitive with deviations from their underlying assumptions. Researchers devised methods that are more robust to such deviations using semi-parametric and nonparametric techniques which address issues that existing methods did not consider.

In this dissertation, we propose modifications to existing semi-parametric classifiers in an effort to create classifiers that are robust to deviations. The first proposed method provides two nonparametric alternatives for the allocation process provided in the classifier by Montanari (2004). The second method is a rank based method where discriminant functions are ranked and we wish to show that ranking can provide equal misclassification error rates in each group which might be very pertinent in some situations.

The optimality of our classifiers is shown by comparing the proposed classifier with existing classifiers. This is done by the use of real data sets and also via extensive Monte Carlo simulation studies. Sensitivity curves will be used to show the effect of local and gross perturbations on the probability of misclassification error rate.

Influence functions are used to ascertain the robustness of the rank based discriminant function theoretically.

Chapter 2

Background

2.1 Introduction

Let us start with the univariate two sample location problem. Suppose $X_1, \ldots, X_{n_x}$ are independent and identically distributed (iid) random variables that are normally distributed with mean μ_x and variance σ^2 ($N(\mu_x, \sigma^2)$) and $Y_1, \ldots, Y_{n_y}$ iid $N(\mu_y, \sigma^2)$. Suppose also that X_i is independent of Y_j for $i = 1, \ldots, n_x$ and $j = 1, \ldots, n_y$. Our desire is to test the hypothesis $H_0 : \mu_x = \mu_y$ versus the alternative hypothesis $H_1 : \mu_x \neq \mu_y$. One then may employ the two sample t test that uses the test statistic

$$t_{xy} = \frac{\bar{X} - \bar{Y}}{S_p \sqrt{\frac{n_x + n_y}{n_x n_y}}} \ ,$$

where S_p is the pooled standard deviation, to obtain an estimate of the standardized separation between the two locations. The null is rejected in favor of the alternative if $|t_{xy}|$ is larger than $t_\alpha(df)$, the critical value, where α is the allowable type-I error rate and *df* are the degrees of freedom. Alternatively, one may use the non-parametric counterpart of the two-sample t test known as the Mann-Whitney test (Hollander and Wolfe, 1999) which uses the test statistic

$$u_{xy} = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \phi(X_i, Y_j) \ , \tag{2.1}$$

where $\phi(r,s) = I\{r < s\}$ and $I\{A\}$ is the indicator function of the set A. Then one rejects the null if $|u_{xy} - (n_x n_y)/2|$ is larger than a given critical value.

Now consider two d-dimensional populations Π_x and Π_y with underlying distributions F and G, respectively, each defined on $\mathbb{R}^d$ for $d \geq 1$. Suppose we have a random sample of size n_x from Π_x given by $\mathbb{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_{n_x}\}$ and, independent of the first sample, a random sample of size n_y from Π_y given by $\mathbb{Y} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_{n_y}\}$. Let F_{n_x} and G_{n_y} represent the empirical distribution functions of $\mathbb{X}$ and $\mathbb{Y}$, respectively. We are now interested in the multivariate two sample location problem of testing the null hypothesis $H_0 : \mu_x = \mu_y$ against the alternative hypothesis $H_1 : \mu_x \neq \mu_y$. H_0 is true iff $\mathbf{u}'\mu_x = \mathbf{u}'\mu_y$ for all $\mathbf{u} \in \mathbb{R}^d$ and H_0 is false if $\mathbf{u}'\mu_x \neq \mathbf{u}'\mu_y$ for at least one $\mathbf{u} \in \mathbb{R}^d$. Under the assumption that the two populations are normal that differ only in the location parameters, $\mathbf{u}'\mu_x = \mathbf{u}'\mu_y$ can be tested using the univariate two-sample test statistic

$$t_{xy}(\mathbf{u}) = \frac{\mathbf{u}'(\bar{\mathbf{X}} - \bar{\mathbf{Y}})}{\sqrt{\mathbf{u}'\mathbf{S}_p\mathbf{u}\left(\frac{n_x+n_y}{n_x n_y}\right)}} \; . \tag{2.2}$$

The rejection region for $H_0 : \mu_x = \mu_y$ versus $H_1 : \mu_x \neq \mu_y$ is $\bigcup_{\|\mathbf{u}\|=1}\{(\mathbf{X}, \mathbf{Y}) \in \mathbb{R}^d \to \mathbb{R}^d : |t_{xy}(\mathbf{u})| \geq c\}$ for a chosen constant c. Thus, we reject H_0 if $\max_{\|\mathbf{u}\|=1} |t_{xy}(\mathbf{u})| \geq c$ or, equivalently, if $\max_{\|\mathbf{u}\|=1} t^2_{xy}(\mathbf{u}) \geq c^*$. Under normality, this gives the Hotelling's T^2 statistic

$$\max_{\|\mathbf{u}\|=1} \; t^2_{xy}(\mathbf{u}) = \frac{n_x + n_y}{n_x n_y}(\bar{\mathbf{X}} - \bar{\mathbf{Y}})'\mathbf{S}_p^{-1}(\bar{\mathbf{X}} - \bar{\mathbf{Y}}) \; .$$

If the covariance structure is I_p, then $\mathbf{u}$ that maximizes $t^2(\mathbf{u})$ is the unit vector on the line that connects the two sample means; that is $(\bar{\mathbf{X}} - \bar{\mathbf{Y}})/\|\bar{\mathbf{X}} - \bar{\mathbf{Y}}\|$. As we will see in Subsection 2.2.1, this will play an important role in developing discriminant analysis procedures.

Classical Discriminant Functions

Now let us consider a classification problem where $\mathbf{Z} \in \Pi_x \cup \Pi_y$ is a new observation that we would like to classify in either Π_x or Π_y. Suppose we have a function D such that $\mathbf{Z}$ is classified in Π_x if

$$D(\mathbf{Z}; F, G) > 0 \; .$$

The function D is known as a discriminant function.

The probability of an observation from Π_x being misclassified in Π_y is

$$P^D_{y|x} = P\left\{D(\mathbf{Z}; F, G) < 0 \mid \mathbf{Z} \sim F\right\}$$

and the probability of misclassifying a random variable from population Π_y into population Π_x is

$$P^D_{x|y} = P\left\{D(\mathbf{Z}; F, G) > 0 \mid \mathbf{Z} \sim G\right\} \; .$$

The total cost of misclassification is then $\pi C_x P^D_{y|x} + (1-\pi) C_y P^D_{x|y}$ where π is the prior probability that an observation comes from Π_x and C_x and C_y are the costs of misclassification of an observation from Π_x in Π_y and from Π_y in Π_x, respectively. Under the assumption that the priors are equal and that the costs of misclassification being equal for the two populations ($C_x = C_y = 1$), the total probability of misclassification (TPM) is $P^D = \frac{1}{2}P^D_{y|x} + \frac{1}{2}P^D_{x|y}$. Hereafter, we will assume the costs C_x and C_y to equal 1.

The following two definitions describe the notions of optimality and robustness in the framework of discriminant analysis.

Definition 2.1. *We say a discriminant function D^* is* optimal *if $P^{D^*} \leq P^D$ for any other discriminant function D and we say that D^* is* more optimal *than D^{**} if $P^{D^*} < P^{D^{**}}$.*

Definition 2.2. *We say that discriminant function D^* is* more robust *to a deviation from distributional property $\mathcal{E}$ than discriminant function D^{**}, if D^* is more optimal than D^{**} under the particular deviation from $\mathcal{E}$.*

Fisher (1936) proposed a classifier that looks at a linear combination of the d-covariates that maximizes the separation or minimizes the overlap between the two populations Π_x and Π_y. This is known as Linear Discriminant Function (LDF) and is given by:

$$L(\mathbf{z}; F, G) \equiv (\mu_x - \mu_y)'\Sigma^{-1}\left[\mathbf{z} - \frac{1}{2}(\mu_x + \mu_y)\right]. \tag{2.3}$$

A new observation $\mathbf{Z} = \mathbf{z}$ is now assigned to Π_x if $L(\mathbf{z}; F, G) > C$ and to Π_y otherwise. The cutoff C is usually 0 when the prior probabilities are unknown. This method is built to be optimal in classifying the new observation $\mathbf{Z}$ under the assumption that F and G are both multivariate normal distributions that are different in location but have the same scatter. In particular, if F and G are $N_d(\mu_x, \Sigma_x)$ and $N_d(\mu_y, \Sigma_y)$, respectively and under the assumption that $\Sigma = \Sigma_x = \Sigma_y$.

For the situation where $\Sigma_x \neq \Sigma_y$ is true, the optimal procedure uses a quadratic combination of the d-covariates that maximizes the separation between the populations Π_x and Π_y. This is known as Quadratic Discriminant Function (QDF) which is given by:

$$Q(\mathbf{z}; F, G) \equiv \ln\left(\frac{|\Sigma_y|}{|\Sigma_x|}\right) - (\mathbf{z} - \mu_x)'\Sigma_x^{-1}(\mathbf{z} - \mu_x) + (\mathbf{z} - \mu_y)'\Sigma_y^{-1}(\mathbf{z} - \mu_y). \tag{2.4}$$

This becomes the optimal rule for classification under the conditions where a new observation $\mathbf{Z} = \mathbf{z}$ is assigned to Π_x if $Q(\mathbf{z}; F, G) > 0$ and to Π_y otherwise.

Given the random samples $\mathbb{X}$ and $\mathbb{Y}$, the sample versions of LDF and QDF are given by

$$L(\mathbf{z}; F_{n_x}, G_{n_y}) = (\bar{\mathbf{x}} - \bar{\mathbf{y}})'\hat{\Sigma}^{-1}\left[\mathbf{z} - \frac{1}{2}(\bar{\mathbf{x}} + \bar{\mathbf{y}})\right]$$

and

$$Q(\mathbf{z}; F_{n_x}, G_{n_y}) = \ln\left(\frac{|\hat{\Sigma}_y|}{|\hat{\Sigma}_x|}\right) - (\mathbf{z} - \bar{\mathbf{x}})'\hat{\Sigma}_x^{-1}(\mathbf{z} - \bar{\mathbf{x}}) + (\mathbf{z} - \bar{\mathbf{y}})'\hat{\Sigma}_y^{-1}(\mathbf{z} - \bar{\mathbf{y}})\ ,$$

respectively, where $\hat{\Sigma}_x$, $\hat{\Sigma}_y$, and $\hat{\Sigma}$ are the estimators of Σ_x, Σ_y, and Σ, respectively.

Linear and Quadratic discriminant functions are optimal under certain conditions and assumptions and may be very sensitive to any deviation from the assumptions. Deviations from normality, equal covariance in the case of LDF, existence of outliers in the case of QDF are examples where LDF and QDF are not optimal. In fact the more serious the deviations and the greater the number of deviations, the more sensitive these methods get. Some work that considered the issue of robustness of LDF and QDF is Lachenbruch et al. (1973), Lachenbruch (1975), Hills (1967), McLachlan (1992), Anderson (1984), Dillon (1979), Johnson et al. (1979) and Seber (1984). A number of these authors investigated the robustness of LDF and QDF with respect to their TPM to some non-linear transformations of the normal distribution suggested in Johnson (1949). Hills (1967) looked at discrimination in data that are non-normal, specifically when the data are discrete. Lachenbruch et al. (1973) investigated the performance of LDF under certain non-normality conditions, specifically, log normal, logit normal and the inverse hyperbolic sine normal distributions. Optimal misclassification probabilities in these cases are calculated by taking an appropriate inverse transformation. In such cases, finding the cutoff C theoretically using a minimax rule is a very difficult problem. So Lachenbruch et al. (1973) determined the value of C, approximately, by using 25 different discrete points. They also found that the transformation makes the two populations heteroscedastic, so they proposed to assign the new observation $\mathbf{Z} = \mathbf{z}$ to Π_x if $Q(\mathbf{z}; F, G) > C$ and to Π_y otherwise. They provided some theoretical results in addition to presenting a Monte Carlo study. Their work and the work of others who extended this research, found that both LDF and QDF are greatly affected by these types of non-normality. As a solution to the sensitivity of the linear and quadratic discriminant functions, several authors have proposed robust procedures.

There are various classifiers, both parametric and nonparametric, which are proposed in literature, like kernel-based classification rule (Mojirsheibani, 2000), k-Nearest Neighbor classification rule (Hellman, 1970), decision trees (Ting, 2002), neural networks (Pao, 1989), logistic regression (Brzezinski, 1999), support vector machines (Gunn, 1998), combined classifiers (LeBlanc and Tibshirani, 1996; Mojirsheibani, 1999, 1997) just to name a few. In this dissertation, we will consider various rank-based procedures for performing classification in an optimal and robust manner.

2.2 Rank Based Procedures for Classification

In this section we will introduce two types of rank-based procedures for classification. The first will be classification based on projection pursuit that uses a rank-based projection index and the second will be based on multivariate ranking.

2.2.1 Projection Pursuit Based Classifiers

Fisher's idea of picking a linear or quadratic combination that maximizes the separation between the two samples is in a way finding the projection that maximizes the separation between the groups based on a particular criterion. In fact, Fisher's LDF could be reframed as finding $\mathbf{u}$, say $\mathbf{u}_0$, the projection direction that maximizes

$$t^2(\mathbf{u}) = \frac{\left[\mathbf{u}'(\bar{\mathbf{X}} - \bar{\mathbf{Y}})\right]^2}{\mathbf{u}'\mathbf{S}_p\mathbf{u}\left(\frac{n_x+n_y}{n_x n_y}\right)}$$

where $X \sim N_d(\mu_x, \Sigma)$, $Y \sim N_d(\mu_y, \Sigma)$ and $\mathbf{S}_p^2 = \frac{(n_x-1)s_x^2+(n_y-1)s_y^2}{n_x+n_y-2}$. Here s_x and s_y are estimates of standard deviations of X and Y respectively. One observes that this is the same two sample t test statistic that was given in Equation (2.2).

The data is then reduced to one dimension by projecting it in the direction given by $\mathbf{u}_0$ and one would classify a new observation $\mathbf{Z} = \mathbf{z}$ into Π_x if $|Z_0 - \bar{X}_0| < |Z_0 - \bar{Y}_0|$, where $X_{0i} = \mathbf{u}_0'\mathbf{X}_i$, $Y_{0j} = \mathbf{u}_0'\mathbf{Y}_j$, and $Z_0 = \mathbf{u}_0'\mathbf{Z}$, $i = 1, \ldots, n_x$ and $j = i, \ldots, n_y$. Otherwise, one classifies $\mathbf{Z}$ into Π_y.

When dealing with spherical distributions, the most "interesting direction" happens to be along the line that connects the means. Fisher's LDF amounts to classifying $\mathbf{Z} = \mathbf{z}$ based on its Euclidean distance from the means. If we are dealing with normal distributions with $\Sigma \neq kI_d$, for some $k > 0$, then the most "interesting" direction is given by

$$\mathbf{u}_0 = \frac{\mathbf{S}_p^{-1}(\bar{\mathbf{X}} - \bar{\mathbf{Y}})}{\|\mathbf{S}_p^{-1}(\bar{\mathbf{X}} - \bar{\mathbf{Y}})\|}$$

In other situations, the projection direction is not generally obvious. This search for "interesting" low dimensional projection of high dimensional data, is more popularly known as *projection pursuit* (Friedman and Tukey, 1974; Posse, 1992). "Interestingness" is defined based on a criterion which is measured through a suitable function known as a *projection index.* It turns out that the projection index for LDF is the two-sample t statistic. Friedman and Tukey (1974) in their study used an algorithm that associated with each direction in high dimensional space, a continuous index that measures its "usefulness" as a projection axis. The projection direction is then varied to maximize the index. The hill-climbing algorithms for maximization given by Rosenbrock (1960) and Powell (1964) were then used as the projection index was sufficiently continuous. Posse (1992) used projection pursuit for 2-group discrimination and he did it using kernel estimation. He used fast fourier transform (FFT) to solve the kernel estimation problem of finding the projection direction that is most interesting.

Projection pursuit is a computationally intensive procedure as it is a thorough process of searching all the projection directions to find the most "interesting" projection. This method gets complicated further as the dimension increases, but this technique is gaining popularity with the increased attention given to savvy computer programming techniques and improvements in computer technology.

Any method that effectively helps reduce the dimension from high to low can be treated as a form of projection pursuit. Principal component analysis (PCA) can be treated as a special case of projection pursuit (Huber, 1985). The aim of classical PCA is to find a linear combinations of the original variables that have maximal sample variance (Nason, 1995). The first principal component a^* is the vector that maximizes the variance of the data $\mathbf{X}$ projected along that direction. i.e. $\mathbf{a}^* = \arg\max \mathrm{Var}_x\{\mathbf{a}'\mathbf{X}\}$. Here the projection index happens to be variance of the projected data. The only other constraint in this method is the need for the projection directions to be orthogonal to each other. It happens so that the first principal component has the highest eigenvalue and hence explains the most variance than any other direction. In that sense, eigenvalues are used as projection indices.

Jones (1983) and Huber (1985) gave strong heuristic arguments indicating that a projection is less interesting the more it is normal.

In addition to lowering the dimension, projection pursuit also allows us to overcome problems associated with sparsity of data in high dimensions (Huber, 1985) which is also termed "curse of dimensionality" (Goldstein, 1987). With increasing dimension, the need for more and more data to meet the requirement of sufficient data increases like a curse. Many techniques fail to perform well under the conditions of sparse data. There are also situations where the number of variables or the dimension (d) is much higher than the amount of data or number of observation (n). These are more popularly known as datasets with $d \gg n$ data. As an example (Huber, 1985) assume that a large number of points is distributed uniformly in the 10-dimensional unit ball. Then the radius of a ball containing 5% of the points is $(0.05)^{0.1} = 0.74$. By using projection pursuit techniques, one can reduce the dimension to one and eliminate the problem of sparsity of data. As noted in Huber (1985), "Projection pursuit is the most powerful technique that can lift a one-dimensional technique to higher dimensions" (Chen, 1989). By this he means that a projection pursuit technique can be used to reduce the dimension to one and then any one-dimensional statistical technique can be applied.

Every traditional projection pursuit methodology mainly differs in the choice of projection index. There are certain demands of a good projection index (Posse, 1995):

- Robust to deviations

- Approximately affine invariant

- Consistent

- Simple enough to permit quick computation even for large data sets.

The problem with Fisher's LDF is that it uses the t-statistic as a projection index which is known to be very sensitive to the underlying distributional assumptions. In an attempt to make Fisher's LDF more robust to deviations, many researchers have proposed to use robust version of projection index in place of the t-statistic. Posse (1990) proposed a projection pursuit technique based on a global optimization algorithm and used a chi-squared projection index to find the most interesting plane (two dimensional view). This optimization algorithm was later modified by Posse (1995) by combining it with a structure removal procedure given in Friedman (1987) and used a modified chi-squared index which satisfies all the demands of a good projection index. Chen and Muirhead (1994) and Chen (1989) in his PhD dissertation used various robust estimates in the likes of the median location estimator (Andrews et al., 1972; Huber, 1981), trimmed location estimator, Huber type location M-estimator, the median absolute scale estimator and Huber type scale M-estimator. Montanari (2004) and Chen and Muirhead (1994) used a two-sample Mann-Whitney type statistic as a projection index to measure group separation. They show that their projection pursuit methods are not sensitive to deviations from the homoscedasticity and normality assumptions that are required for the optimality of Fisher's LDF. The method proposed by Montanari was motivated by the idea of what is known as transvariation (Gini, 1916; Montanari, 2004).

Transvariation

Consider two continuous univariate populations Π_x and Π_y with distributions F and G, respectively, defined on $\mathbb{R}$. Suppose we have a random sample $X_1, \ldots, X_{n_x}$ from Π_x and, independent of the first sample, a random sample $Y_1, \ldots, Y_{n_y}$ from Π_y. The two samples are said to *transvariate* with respect to their measures of centers m_x and m_y if there is at least one pair (i, j) such that $(X_i - Y_j)(m_x - m_y) < 0$, $i = 1, \ldots n_x,\ j = 1, \ldots, n_y$. Any difference satisfying this condition is called a *two-group transvariation.* Similarly, the sample $X_1, \ldots, X_{n_x}$ and a given constant $c \in \mathbb{R}$ transvariate with respect to m_x, if there is at least one i such that $(X_i - c)(m_x - c) < 0$, $i = 1, \ldots, n_x$. This is known as a *point-group transvariation.*

The two-group transvariation probability between F and G is defined as

$$\begin{aligned}\tau_{xy} := \tau(F, G) &= P\{(Y - X)(\mu_y - \mu_x) < 0\} \\ &= \int_{\mathbb{R}} \int_{\mathbb{R}} I\{(\mathbf{y} - \mathbf{x})(\mu(G) - \mu(F)) < 0\} dF(x) dG(y)\ , \qquad (2.5)\end{aligned}$$

where $X \ \sim\ F$, $Y \ \sim\ G$ and $\mu_x = \mu(F)$, $\mu_y = \mu(G)$ are the location parameters of F and G, respectively. If F_{n_x} and G_{n_y} be the two empirical distributions of the two random samples. Then an estimate of τ_{xy} is given as

$$\begin{aligned} T_{xy} := \tau(F_{n_x}, G_{n_y}) &= \int_{\mathbb{R}}\int_{\mathbb{R}} I\{(\mathbf{y}-\mathbf{x})(\mu(G_{n_y})-\mu(F_{n_x}))<0\}dF_{n_x}(x)dG_{n_y}(y) \\ &= \frac{1}{n_x n_y}\sum_{i=1}^{n_x}\sum_{j=1}^{n_y} I\{(X_i-Y_j)(m_x-m_y)<0\}\ , \end{aligned}$$

where $I\{A\}=1$ if A is true and is 0 otherwise. Here $m_x=\mu(F_{n_x})$ and $m_y=\mu(G_{n_y})$ are estimators of μ_x and μ_y, respectively. T_{xy} is a nonparametric estimator of the overlap between the distributions F_x and F_y. In particular, $n_x n_y T_{xy}$ gives the number of observations that need to be interchanged so that there will be no overlap between the two samples. If we assume without loss of generality that $\mu_y<\mu_x$, then $\tau_{xy}=P(X<Y)$ which is estimated by

$$T_{xy} = \frac{1}{n_x n_y}\sum_{i=1}^{n_x}\sum_{j=1}^{n_y} I\{X_i<Y_j\} = \frac{u_{xy}}{n_x n_y}\ ,$$

where u_{xy} is the Mann-Whitney statistic given in Equation (2.1). It is easy to see the connection between u_{xy} and ranks as

$$u_{xy} = \sum_{j=1}^{n_y} R(Y_j) + \frac{n_y(n_y+1)}{2}\ ,$$

where $R(Y_j)$ is the rank of Y_j in the joint ranking of $X_1,\ldots,X_{n_x}$ and $Y_1,\ldots,Y_{n_y}$ for $j=1,\ldots,n_y$.

The point-group transvariation probability between F and a constant $c \in \mathbb{R}$ is given by

$$\begin{aligned} \tau_x(c) &= P\{(X-c)(\mu_x - c) < 0\} \\ &= \int_{\mathbb{R}} I\{(\mathbf{x}-c)(\mu(F)-c) < 0\} dF(x) \ , \end{aligned}$$

an estimator of $\tau_x(c)$ is

$$\begin{aligned} T_x(c) &= \int_{\mathbb{R}} I\{(\mathbf{x}-c)(\mu(F_{n_x})-c) < 0\} dF_{n_x}(x) \\ &= \frac{1}{n_x} \sum_{i=1}^{n_x} I\{(X_i - c)(m_x - c) < 0\} \ . \end{aligned} \tag{2.6}$$

$T_x(c)$ measures the centrality of the constant c in the sample $X_1, \ldots, X_{n_x}$. In a way, $T_x(c)$ measures how deep the point c is in the sample $X_1, \ldots, X_{n_x}$. The quantity $n_x T_x(c)$ is the smallest number of observations in the first sample that c needs to *skip* so that all the sample points are to one side of it.

It is difficult to directly generalize the idea of transvariation probability for dimensions higher than one. Projection pursuit offers a way to project high dimensional data into a single dimension where we can compute transvariation probabilities.

Let $F_{\mathbf{u}}$ and $G_{\mathbf{u}}$ be the distributions of $\mathbf{u}'\mathbf{X}$ and $\mathbf{u}'\mathbf{Y}$, respectively, where $\mathbf{X} \sim F$ from population Π_x and $\mathbf{Y} \sim G$ from population Π_y are d-dimensional random variables and $\mathbf{u} \in \mathbb{R}^d$ is a unit vector. The overlap between $F_{\mathbf{u}}$ and $G_{\mathbf{u}}$ with respect to the transvariation probability is

$$\begin{aligned}&P(I\{(\mathbf{u}'\mathbf{X}-\mathbf{u}'\mathbf{Y})(\mu(F_{\mathbf{u}})-\mu(G_{\mathbf{u}}))<0\})\\ &\qquad\qquad=\int_{\mathbb{R}}\int_{\mathbb{R}} I\{(x-y)(\mu(F_{\mathbf{u}})-\mu(G_{\mathbf{u}}))<0\}dF_{\mathbf{u}}(x)dG_{\mathbf{u}}(y)\ ,\end{aligned}$$

where $\mu(F_{\mathbf{u}})$ and $\mu(G_{\mathbf{u}})$ are the location parameters of $F_{\mathbf{u}}$ and $G_{\mathbf{u}}$, respectively. We are interested in finding the projection direction that minimizes this overlap between $F_{\mathbf{u}}$ and $G_{\mathbf{u}}$; that is

$$\mathbf{u}_{opt}=\underset{\|\mathbf{u}\|=1}{\text{Argmin}}\left\{\int_{\mathbb{R}}\int_{\mathbb{R}} I\{(x-y)(\mu(F_{\mathbf{u}})-\mu(G_{\mathbf{u}}))<0\}dF_{\mathbf{u}}(x)dG_{\mathbf{u}}(y)\right\}\ .$$

Given two independent random training samples $\mathbf{X}_1,\ldots,\mathbf{X}_{n_x}$ and $\mathbf{Y}_1,\ldots,\mathbf{Y}_{n_y}$ from Π_x and Π_y, respectively, defined on $\mathbb{R}^d$ $(d\geq 1)$, the estimator of the direction of minimum overlap is given by

$$\hat{\mathbf{u}}_{opt}:=\underset{\|\mathbf{u}\|=1}{\text{Argmin}}\left\{\frac{1}{n_x n_y}\sum_{i=1}^{n_x}\sum_{j=1}^{n_y} I\{(\mathbf{u}'\mathbf{X}_i-\mathbf{u}'\mathbf{Y}_j)(m_x(\mathbf{u})-m_y(\mathbf{u}))<0\}\right\}\ , \qquad (2.7)$$

where $m_x(\mathbf{u})$ and $m_y(\mathbf{u})$ are the locations of the two projected samples $\mathbf{u}'\mathbf{X}$ and $\mathbf{u}'\mathbf{Y}$, respectively. This vector is assumed to be the direction that gives the most interesting view of the data in one dimension as it gives the direction of maximum separation as measured by Gini's transvariation probability (Gini, 1916).

Once the direction that gives the minimum overlap (maximum separation) $\hat{\mathbf{u}}_{opt}$ is found using projection pursuit, the entire data is projected onto that direction and a new observation $\mathbf{Z} = \mathbf{z}$ is allocated into Π_x if $|\mathbf{Z}_0 - m_x(\hat{\mathbf{u}}_{opt})| < |\mathbf{Z}_0 - m_y(\hat{\mathbf{u}}_{opt})|$, where $\mathbf{Z}_0 = \hat{\mathbf{u}}'_{opt}\mathbf{Z}$. Otherwise, we classify $\mathbf{Z}$ into Π_y.

Projection pursuit techniques are very suitable when d is not really high as they sweep the entire space for the best possible view of the data. The biggest problem with projection pursuit though is that it relays its problems to the various techniques that are using projection pursuit as the first step to reduce dimension. The problem is that projection pursuit tries to squish the data into one dimension using a linear combination. This causes problems when certain kinds of data sets like "Banana data" (Rätsch et al., 1998; Rätsch, 1998) and the Hardy data (Hardy, 1991) having certain 'strange' shapes that projection pursuit techniques cannot handle. These are examples of low dimensional data sets which seemingly look and actually are perfectly separated but there exists no projection direction that can reduce the dimension to one with complete separation. Another problem with projection pursuit is the computational complication as the dimension increases; although one can always decrease the number of projections considered as the dimension increases by using advanced computational techniques. If done effectively, this is not going to affect the efficiency of the projection pursuit technique (Filzmoser et al., 2006).

There is still a need for an alternative dimension reduction technique that is nonparametric and robust but does not necessarily require extensive computation. Needless to say, the most important aspect should be to somehow eliminate the drawback with projection pursuit techniques. One such technique is the use of what are called multivariate ranks.

2.2.2 Procedures Based on Multivariate Ranking

In the univariate setting, statistical methods that use rank-based nonparametric techniques do not depend on restrictive distributional assumptions and hence are robust to deviations from these assumptions. For higher dimensions, an alternative to projection pursuit is the idea of *data depth* which is a multivariate version of ranks (see Eddy (1985); Liu (1992)). Data depths are used to measure the "outlyingness" or alternatively "centrality" of a given multivariate sample point with respect to its underlying distribution (Liu et al., 1999; Zuo and Serfling, 2000; Mosler, 2002). In particular, a depth function assigns higher values to points that are more central with respect to a data cloud. This naturally gives a center-outward ranking of the sample points. A number of depth functions are available in the literature. A few popular depth functions are Mahalanobis depth (Mahalanobis, 1936; Liu and Singh, 1993), halfspace depth (Tukey, 1974), simplicial depth (Liu, 1990), majority depth (Singh, 1991), projection depth (Donoho, 1982), and spatial or L_1 depth (Vardi and Zhang, 2000; Jörnsten, 2004). Definition of some of the more popular depth functions are:

- The L_1 or Spatial depth function is given by

$$\mathcal{D}_1(\mathbf{x}; F) = 1 - \left\| E_F \left\{ \frac{\mathbf{x} - \mathbf{X}}{\|\mathbf{x} - \mathbf{X}\|} \right\} \right\| , \tag{2.8}$$

  where $\mathbf{X} \sim F$ and $\| \cdot \|$ is the Euclidean norm.

- The Mahalanobis depth function is given by

  $$MD(\mathbf{x}; F) = [1 + (x - \mu_F)\Sigma_F^{-1}(x - \mu_F)]^{-1},$$

  where μ_F and Σ_F are the mean vector and dispersion matrix of F, respectively. The sample version of M is obtained by replacing μ_F and Σ_F with their sample estimates.

- The half-space depth function is given by

  $$HD(\mathbf{x}; F) = \inf_H \{P(H) : H \text{ is a closed half-space in } \mathbb{R}^d, \mathbf{x} \in H\}$$

  It turns out that $\tau_x(c)$ is the half-space depth of c in one dimension with respect to the population F, that is, $\tau_x(c) = HD(c; F)$. Half-space depth is sometimes also referred to as Tukey depth.

We can easily see that $0 \leq DF \leq 1$, where DF is any depth function, and $\mathbf{x}_1$ is more central to (or deeper in) F than $\mathbf{x}_2$ is central in F if $DF(\mathbf{x}_1; F) > DF(\mathbf{x}_2; F)$. This is true for any depth function DF. Let $\mathcal{F}$ be the class of distributions on the Borel sets of $\mathbb{R}^d$. A statistical depth function is a bounded, nonnegative mapping $D : \mathbb{R}^d \times \mathcal{F} \to \mathbb{R}$ and there are certain properties that are desired of depth functions (Zuo and Serfling, 2000; Hoberg, 2003):

- *Affine Invariance*: The depth of a point $\mathbf{x} \in \mathbb{R}^d$ should not depend on the underlying coordinate system or, in particular, on the scales of the underlying measurements.

$$DF(\mathbf{Ax}+\mathbf{b}; F_{\mathbf{AX}+\mathbf{b}}) = DF(\mathbf{x}; F_{\mathbf{X}})$$

- *Maximality at center*: For a distribution having a uniquely defined "center" (e.g., the point of symmetry with respect to some notion of symmetry), the depth function should attain maximum value at this center. If μ is the center of F, then

$$DF(\mu; F) = \sup_{\mathbf{x}\in\mathbb{R}^d} DF(\mathbf{x}; F)$$

- *Monotonicity relative to deepest point*: As a point $\mathbf{x} \in \mathbb{R}^d$ moves away from the "deepest point" (the point at which the depth function attains maximum value; in particular, for a symmetric distribution, the center) along any fixed ray through the center, the depth at x should decrease monotonically.

$$DF(\mathbf{x}; F) \leq DF(\mu + \alpha(\mathbf{x}-\mu); F) \text{ for } \alpha \in [0,1]$$

- *Vanishing at infinity*: The depth of a point $\mathbf{x}$ should approach zero as $\|\mathbf{x}\|$ approaches infinity.

$$DF(\mathbf{x}; F) \to 0 \text{ as } \|\mathbf{x}\| \to \infty$$

The interested reader may find an extensive list of depth functions along with their definitions in Liu et al. (1999), Zuo and Serfling (2000) or Ghosh and Chaudhuri (2005). Among the numerous depth functions that are in existence, Mahalanobis depth and L_1 depth (spatial depth) are two of the most attractive ones due to their ease of computation. They can be computed exactly for any dimension. The computation of many other depth functions may require algorithms that provide only approximations. This is especially true for higher dimensional data. For example, one usually has to construct very complicated approximate algorithms to compute the halfspace depth of points in three or higher dimensions.

Taking advantage of this notion of ordering multivariate data in a center-outward manner, Jörnsten (2004) proposed the *maximum L_1 depth classifier* that uses the discriminant function

$$\begin{aligned} S(\mathbf{z}; F, G) = \mathcal{D}_1(\mathbf{z}; F) - \mathcal{D}_1(\mathbf{z}; G) &= \left\| E_G \left\{ \frac{\mathbf{z} - \mathbf{Y}}{\|\mathbf{z} - \mathbf{Y}\|} \right\} \right\| - \left\| E_F \left\{ \frac{\mathbf{z} - \mathbf{X}}{\|\mathbf{z} - \mathbf{X}\|} \right\} \right\| \\ &= \left\| \int_{\mathbb{R}^d} \frac{\mathbf{z} - \mathbf{y}}{\|\mathbf{z} - \mathbf{y}\|} dG(\mathbf{y}) \right\| - \left\| \int_{\mathbb{R}^d} \frac{\mathbf{z} - \mathbf{x}}{\|\mathbf{z} - \mathbf{x}\|} dF(\mathbf{x}) \right\| . \end{aligned} \tag{2.9}$$

The new observation $\mathbf{Z} = \mathbf{z}$ is then classified in Π_x if $S(\mathbf{z}; F, G) > 0$ and in Π_y otherwise. Despite its computational ease, a major drawback of this classifier is that it lacks affine invariance because L_1 depth is not affine invariant. It can, however, be made affine invariant by taking $\Sigma_x^{-1/2}(\mathbf{z} - \mathbf{X})$ and $\Sigma_y^{-1/2}(\mathbf{z} - \mathbf{Y})$ in place of $\mathbf{z} - \mathbf{X}$ and $\mathbf{z} - \mathbf{Y}$, respectively, in equation (2.9) (Vardi and Zhang, 2000; Ghosh and Chaudhuri, 2005). Note that one can use any affine equivariant estimators of Σ_x and Σ_y when computing the discriminant function. If the scatter estimator of Tyler (1987) is used, then the resulting maximum L_1 depth classifier is very similar to the classifier given by Crimin et al. (2007). An alternative method of obtaining affine invariance is to scale the data along its principal component directions (PCA-scaling) as given in Hugg et al. (2006). One could use robust principal component analysis (eg: Robust PCA given by Croux et al. (2007)) or scale the data with the robust estimate of covariance structure which will make the L_1 depth function affine invariant in addition to making it robust against deviations.

Once again, for practical purposes, given two independent random training samples $\mathbf{X}_1, \dots, \mathbf{X}_{n_x}$ and $\mathbf{Y}_1, \dots, \mathbf{Y}_{n_y}$ from Π_x and Π_y, respectively, defined on $\mathbb{R}^d$ $(d \geq 1)$, the sample version of $\mathcal{D}_1(\mathbf{x}; F)$ given in (2.8) can be found by replacing the empirical cdf in place of F and G resulting in the sample version of $S(\mathbf{z}; F, G)$ given by

$$\begin{aligned} S(\mathbf{z}; F_{n_x}, G_{n_y}) &= \left\| \int_{\mathbb{R}^d} \frac{\mathbf{z}-\mathbf{y}}{\|\mathbf{z}-\mathbf{y}\|} dG_{n_y}(\mathbf{y}) \right\| - \left\| \int_{\mathbb{R}^d} \frac{\mathbf{z}-\mathbf{x}}{\|\mathbf{z}-\mathbf{x}\|} dF_{n_x}(\mathbf{x}) \right\| \\ &= \left\| \frac{1}{n_y} \sum_{j=1}^{n_y} \frac{\mathbf{z}-\mathbf{Y}_j}{\|\mathbf{z}-\mathbf{Y}_j\|} \right\| - \left\| \frac{1}{n_x} \sum_{i=1}^{n_x} \frac{\mathbf{z}-\mathbf{X}_i}{\|\mathbf{z}-\mathbf{X}_i\|} \right\| . \end{aligned}$$

It must be noted that the maximum L_1 depth classifier is in the class of classifiers known as *maximum depth classifiers* (Ghosh and Chaudhuri, 2005) in that any depth function DF can be used in place of the L_1 depth function. The optimality of the classifier is dependent on the choice of the depth function. The choice of depth functions could be based on various properties like robustness. You obtain QDF if Mahalanobis depth is used in place of the L_1 depth in (2.9). One would assign the new observation $\mathbf{Z} = \mathbf{z}$ in Π_x if $DF(\mathbf{z}; F, G) > 0$ and in Π_y otherwise.

In Billor et al. (2008), it is shown that the robustness of the maximum depth classifier can be improved if one considers the largest order of depth instead of just the maximum depth. They define the transvariation probability between the depth of $\mathbf{Z}$ and the depth of $\mathbf{X}$ relative to the distribution of $\mathbf{X}$ as

$$\begin{aligned}\tau_x(\mathbf{Z}) &= P\left\{(DF(\mathbf{X};F) - DF(\mathbf{Z};F))(DF(\mu_x;F) - DF(\mathbf{Z};F)\right\} < 0) \\ &= P\left\{DF(\mathbf{X};F) < DF(\mathbf{Z};F)\right\}\ ,\end{aligned}$$

where the second equality is due to the fact that μ_x is the point of maximum depth. Given a random sample $\mathbf{X}_1, \ldots, \mathbf{X}_{n_x}$, an estimator of $\tau_x(\mathbf{Z})$ is

$$\hat{\tau}_x(\mathbf{Z}) = P\left\{DF(\mathbf{X};F_{n_x}) < DF(\mathbf{Z};F_{n_x})\right\} = \frac{1}{n_x}\sum_{i=1}^{n_x} I\{DF(\mathbf{X}_i;F_{n_x}) < DF(\mathbf{Z};F_{n_x})\}$$

Based on these, they define the maximum depth rank discriminant function as

$$\begin{aligned}DT(\mathbf{z};F,G) &= P\left\{DF(\mathbf{X};F) < DF(\mathbf{z};F)\right\} - P\left\{DF(\mathbf{Y};G) < DF(\mathbf{z};G)\right\} \\ &= \int_{\mathbb{R}^d} I\{DF(\mathbf{x};F) < DF(\mathbf{z};F)\}dF(\mathbf{x}) - \\ &\qquad \int_{\mathbb{R}^d} I\{DF(\mathbf{y};G) < DF(\mathbf{z};G)\}dG(\mathbf{y})\end{aligned}$$

which is estimated by

$$\begin{aligned}DT(\mathbf{z};F_{n_x},G_{n_y}) =& \frac{1}{n_x}\sum_{i=1}^{n_x} I\{DF(\mathbf{X}_i;F_{n_x}) < DF(\mathbf{z};F_{n_x})\} - \\ &\qquad \frac{1}{n_y}\sum_{j=1}^{n_y} I\{DF(\mathbf{Y}_j;G_{n_y}) < DF(\mathbf{z};G_{n_y})\}\ .\end{aligned}$$

The maximum depth rank classifier classifies $\mathbf{z}$ into Π_x if $DT(\mathbf{z}; F_{n_x}, G_{n_y}) > 0$. This classifier is robust just like most rank-based procedures and can be directly extended to more than two groups unlike the transvariation-based method.

The use of data depth for classification purposes has also been considered by Mosler and Hoberg (2003). They define a combination of two existing depth functions zonoid and Mahalanobis to come up with a new depth function and they call it zonoid-Mahalanobis depth. Mahalanobis depth is parametric depth function that is sensitive to deviations, while zonoid depth function is nonparametric and based on convex hulls. In this regard, halfspace, simplicial and convex-hull peeling depths are in the same class of depths as zonoid depth. A drawback of such depth functions is that they vanish outside the convex hull, so points lying outside the convex hulls of all classes cannot be classified using such depth functions. The zonoid-Mahalanobis depth function uses a combination as in one would use the zonoid depth as long as the point lies inside at least one of the convex hulls, else the Mahalanobis depth is used for classification. Halfspace depth based classification for two populations has been suggested by Christmann and Rousseeuw (2001) and Christmann et al. (2002). Donoho and Gasko (1992) showed that the computation of HD in higher dimensions can be performed using the projection pursuit method. In particular, the sample half-space depth in one-dimension is defined as

$$HD_1(x; X) = \min(\#\{i : X_i \leq x\}, \#\{i : X_i \geq x\})$$

and for $\mathbf{x} \in \mathbb{R}^d$ the sample half-space depth is defined as,

$$HD_d(\mathbf{x}; \mathbf{X}) = \min_{\|\mathbf{u}\|=1} HD_1(\mathbf{u}'\mathbf{x}; \mathbf{u}'\mathbf{X}) \tag{2.10}$$

where the dataset $\mathbf{u}'\mathbf{X}$ is the one-dimensional projection of the d-dimensional dataset $\mathbf{X}$.

This means that the maximum depth classifier based on the half-sapce depth (Ghosh and Chaudhuri, 2005) is equivalent to using projection pursuit with point-group transvariation (Montanari, 2004) and allocating using a minmax rule. To that end, let F and G be the data generating distributions and $F_{\mathbf{u}}$ and $G_{\mathbf{u}}$ be the distributions of the projected data as defined above. Then using the definition given in Equation (2.6) we can define the discriminant function

$$\begin{aligned} D_{\text{pgt}}(\mathbf{z}; F, G) = \min_{\|\mathbf{u}\|=1} \int_{\mathbb{R}} I\{(x - \mathbf{u}'\mathbf{z})(\mu(F_{\mathbf{u}}) - \mathbf{u}'\mathbf{z}) < 0\} dF_{\mathbf{u}}(x) \, - \\ \min_{\|\mathbf{u}\|=1} \int_{\mathbb{R}} I\{(y - \mathbf{u}'\mathbf{z})(\mu(G_{\mathbf{u}}) - \mathbf{u}'\mathbf{z}) < 0\} dG_{\mathbf{u}}(y) \end{aligned}$$

and allocate $\mathbf{z}$ in Π_x if $D_{\text{pgt}}(\mathbf{z}; F_{n_x}, G_{n_y}) > 0$.

2.3 Classifiers Based on Robust Estimators of Location and Scale

The reason LDF and QDF are sensitive to deviations from underlying assumptions is due to the fact that the estimators of mean and covariance that they use are sensitive to such deviations. So a natural idea is to replace these quantities with robust quantities to attain robustness in the classifier. Perhaps the earliest such procedure was proposed by Randles et al. (1978b) where Huber's M estimates of the mean vector and covariance matrix (Huber, 1977) are used in place of the mean and covariance used in LDF and QDF.

2.3.1 M Estimators

We will give a brief discussion of Huber's M estimates in the following. M estimates are solutions, $\hat{\theta}$, that maximize

$$\sum_{i=1}^{n} \rho(x_i, \theta) \ ,$$

where ρ is a function that depends on the sample and a vector of parameters θ. The properties of this function are discussed below. Often it is simpler to differentiate with respect to θ and solve for the root of the derivative. When this differentiation is possible, the M-estimator is said to be of ψ-type. Otherwise, the M-estimator is said to be of ρ-type.

ρ type

For positive integer d, let $(\mathcal{X}, \Sigma)$ and $(\Theta \subset \mathbb{R}^d, S)$ be measure spaces. $\theta \in \Theta$ is a vector of parameters. An M-estimator of ρ-type T is defined through a measurable function $\rho : \mathcal{X} \times \Theta \rightarrow \mathbb{R}$. It maps a probability distribution F on $\mathcal{X}$ to the value

$T(F) \in \Theta$ (if it exists) that minimizes $\int_{\mathcal{X}} \rho(x,\theta)dF(x)$:

$$\hat{\theta} = T(F) := \arg\min_{\theta\in\Theta} \int_{\mathcal{X}} \rho(x,\theta)dF(x) \ .$$

ψ type

If ρ is differentiable, then the computation of $\hat{\theta}$ is usually much easier. An M-estimator of ψ-type T is defined through a measurable function $\psi : \mathcal{X} \times \Theta \rightarrow \mathbb{R}^d$. It is assumed that the true parameter θ satisfies $\int_{\mathcal{X}} \psi(x,\theta)dF(x) = 0$. Then a ψ-type M estimator $T(F)$ is defined implicitly as the solution of the vector equation

$$\int_{\mathcal{X}} \psi(x,T(F))dF(x) = 0 \ .$$

For many choices of ρ or ψ, no closed form solution exists and an iterative approach to computation is required. It is possible to use standard function optimization algorithms or an iteratively re-weighted least squares fitting algorithm typically happens to be the preferred method.

2.3.2 S Estimators

S-estimator based classifiers are given by He and Fung (2000) and Croux and Dehon (2001). S estimators were first defined in the context of regression by Rousseeuw and Yohai (1984). Let $\Delta(\mathbf{x};\mathbf{a},\mathbf{C}) = \{(\mathbf{x}-\mathbf{a})'\mathbf{C}^{-1}(\mathbf{x}-\mathbf{a})\}^{\frac{1}{2}}$ for any $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{a} \in \mathbb{R}^d$ and $\mathbf{C} \in S(d)$, where $S(d)$ is the set of all symmetric positive definite matrices in

$\mathbb{R}^d \times \mathbb{R}^d$. Let $s(\mathbf{a}, \mathbf{C})$ be the solution of

$$\frac{1}{m}\sum_{i=1}^{m} \rho \left\{ \frac{\Delta(\mathbf{x}_i; \mathbf{a}, \mathbf{C})}{s(\mathbf{a}, \mathbf{C})} \right\} = E\left\{ \rho(||x||^{\frac{1}{2}}) \right\} \ ,$$

where ρ is such that $\rho(0) = 0$, where ρ is symmetric about 0 and ρ is nondecreasing on $(0, c)$ and constant on (c, ∞) for some constant $c > 0$. Here the expectation on the right hand side is taken at the standard d-variate normal distribution. Now let $(\mathbf{a}^*, \mathbf{C}^*)$ be the minimizers of $s(\mathbf{a}, \mathbf{C})$ subject to $det(\mathbf{C}) = 1$, then

$$\tilde{\mu}^S = \mathbf{a}^* \text{ and}$$

$$\tilde{\Sigma}^S = s(\mathbf{a}^*, \mathbf{C}^*)\mathbf{C}^*$$

are the S-estimators of μ and Σ.

2.3.3 MCD Estimators

The MCD (Maximum Covariance Determinant) based LDF and QDF given in Hubert and Van Driessen (2004), the classifier based on a robust version of the Lawley-Hotelling test uses the spatial median estimator of Hettmansperger and Randles (2002) and the related scatter estimator of Tyler (1987) given by Crimin et al. (2007).

Hubert and Van Driessen (2004) used the re-weighted MCD estimate of multivariate location and scale (Rousseeuw, 1984, 1985), because of its good statistical properties and the FAST-MCD algorithm (Rousseeuw and Van Driessen, 1999) which provides an efficient algorithm of computing the estimates for large data sets.

For the X sample, the MCD estimator is defined as the mean $\hat{\mu}_{x,0}$ and the covariance matrix $S_{x,0}$ of the h_x observations (out of n_x) whose covariance matrix has the lowest determinant. The quantity h_x should be larger than $\lfloor (n_x - p + 1) = 2 \rfloor$ and $n_x - h_x$ should be smaller than the number of outliers in the X population. With this choice the MCD attains its maximal breakdown value of $\lfloor (n_x - p + 1) = 2 \rfloor \approx 50\%$. The breakdown value of an estimator is defined as the largest percentage of contamination it can withstand (Rousseeuw and Leroy, 1987). If one suspects, less than 25% contamination in the X sample, it is advised to take $h_x \approx 0.75 n_x$ as this yields a higher finite-sample efficiency (Croux and Haesbroeck, 2000). Based on the initial estimates $\hat{\mu}_{x,0}$ and $S_{x,0}$ one computes, for each observation x_i, its (preliminary) robust distance (Rousseeuw and Vanzomeren, 1990)

$$RD^0_{x_i} = \sqrt{(x_i - \hat{\mu}_{x,0})' S^{-1}_{x,0} (x_i - \hat{\mu}_{x,0})}\ , \quad i = 1, \ldots, n_x$$

They assign weight 1 to x_i if $RD^0_{x_i} \leq \sqrt{\chi^2_{p,0:975}}$ and weight 0 otherwise. The reweighted MCD estimator is then obtained as the mean $\hat{\mu}_{x,MCD}$ and the covariance matrix $\hat{\Sigma}_{x,MCD}$ of those observations with weight 1. It is shown by Croux and Haesbroeck (2000) that this reweighting step increases the finite-sample efficiency of the MCD estimator considerably, whereas the breakdown value remains the same. This can also be used to flag outliers and so can be used to detect outliers.

The Robust Quadratic Discriminant Rule (M) is: $\mathbf{Z} = \mathbf{z} \in \Pi_y$ if $d_2^M(\mathbf{z}) > d_1^M(\mathbf{z})$ where

$$d_x^M(\mathbf{z}) = -\frac{1}{2} ln|\hat{\Sigma}_{x,MCD}| - \frac{1}{2}(\mathbf{z} - \hat{\mu}_{x,MCD})'\hat{\Sigma}_{x,MCD}^{-1}(\mathbf{z} - \hat{\mu}_{x,MCD}) \qquad (2.11)$$

and $\mathbf{z} \in \Pi_x$ otherwise. The quantity $d_y^M(\mathbf{z})$ is defined analogously as $d_x^M(\mathbf{z})$.

Robustified Fisher's linear discriminant rule (RLDR), which can be described as $\mathbf{z} \in \pi_x$ if

$$(\hat{\mu}_x - \hat{\mu}_y)'\hat{\Sigma}^{-1}(\mathbf{z} - (\hat{\mu}_x - \hat{\mu}_y)/2) > 0;$$

and $\mathbf{z} \in \pi_y$ otherwise.

To construct RLDR they first look for initial estimates of the group means and the common covariance matrix, denoted by $\hat{\mu}_{x,0}$, $\hat{\mu}_{y,0}$ and $\hat{\Sigma}_0$. This will already yield a discriminant rule based on $\hat{d}_x^{RL}(\mathbf{z}, \hat{\mu}_{x,0}, \hat{\Sigma}_0)$ and $\hat{d}_y^{RL}(\mathbf{z}, \hat{\mu}_{y,0}, \hat{\Sigma}_0)$. We will then also consider the reweighting procedure based on the robust distances

$$RD_{x_i}^0 = \sqrt{(x_i - \hat{\mu}_{x,0})'\hat{\Sigma}_0^{-1}(x_i - \hat{\mu}_{x,0})}\,, \quad i = 1, \ldots, n_x$$

and

$$RD_{y_j}^0 = \sqrt{(y_j - \hat{\mu}_{y,0})'\hat{\Sigma}_0^{-1}(y_j - \hat{\mu}_{y,0})}\,, \quad j = 1, \ldots, n_y$$

For each observation in X sample, let $w_{x_i} = 1$ if $RD_{x_i}^0 \leq \sqrt{\chi^2_{p,0:975}}$ and $w_{x_i} = 0$ otherwise. w_{y_j} are defined similarly for the Y sample. The final estimates are then obtained as the mean and the pooled covariance matrix of the observations with weight 1, i.e.

$$\hat{\mu}_x = \frac{\sum_{i=1}^{n_x} w_{x_i} x_i}{\sum_{i=1}^{n_x} w_{x_i}} \ , \quad \hat{\mu}_y = \frac{\sum_{j=1}^{n_y} w_{y_j} y_j}{\sum_{j=1}^{n_y} w_{y_i}} \ ,$$

$$\hat{\Sigma} = \frac{\sum_{i=1}^{n_x} w_{x_i}(x_i - \hat{\mu}_x)(x_i - \hat{\mu}_x)' + \sum_{j=1}^{n_y} w_{y_j}(y_j - \hat{\mu}_y)(y_j - \hat{\mu}_y)'}{\sum_{i=1}^{n_x} w_{x_i} + \sum_{j=1}^{n_y} w_{y_j}} \tag{2.12}$$

and the resulting linear discriminant rule is then based on $\hat{d}_x^{RL}(\mathbf{z}, \hat{\mu}_x, \hat{\Sigma})$ and $\hat{d}_y^{RL}(\mathbf{z}, \hat{\mu}_y, \hat{\Sigma})$.

To obtain the initial covariance estimate $\hat{\Sigma}_0$, they consider three different methods. The first approach is straightforward, and has been applied by Chork and Rousseeuw (1992) using the Minimum Volume Ellipsoid estimator (Rousseeuw, 1984), and Croux and Dehon (2001) using S-estimators (Rousseeuw and Yohai, 1984). The MCD estimates $\hat{\mu}_{x,MCD}$, $\hat{\mu}_{y,MCD}$, $\hat{\Sigma}_{x,MCD}$ and $\hat{\Sigma}_{y,MCD}$ are obtained, and then the individual covariance matrices are pooled, yielding

$$\hat{\Sigma}_{PCOV} = \frac{n_x \hat{\Sigma}_{x,MCD} + n_y \hat{\Sigma}_{y,MCD}}{n_x + n_y}$$

For the second approach, they adapt one of the proposals of He and Fung (2000) who use S-estimators to robustify Fisher's linear discriminant function. The idea is based on pooling the observations instead of the group covariance matrices.

The third estimator combines the two previous approaches and is aimed to find a fast approximation to the Minimum Within-group Covariance Determinant criterion of Hawkins and McLachlan (1997). Instead of applying the same trimming proportion to each group, they proposed to find the h observations out of the whole data set of size n, such that the pooled within-group sample covariance matrix $\hat{\Sigma}_H$ has minimal determinant. The algorithm described in Hawkins and McLachlan (1997) is very time-consuming because it is based on pairwise swaps. Hubert and Van Driessen (2004) proposed the fast approximation for two groups which is much faster than the algorithm given in Hawkins and McLachlan (1997). It has to be noted that this algorithm can fail if some of the groups are very small where there is a possibility that the final subset H does not contain $p + 1$ observations from each group, making those group covariance matrices singular.

Similar to Hubert and Van Driessen (2004), Joossens and Croux (2004) performed simulation studies to compare LDF and QDF with the MCD based LDF and QDF as well as the S-estimator based classifiers.

2.3.4 Other

There are other estimates like the R-estimates and the L-estimates (Serfling, 1980) which could also be used in classification in a similar fashion.

An issue that was discussed in Lachenbruch et al. (1973) is the problem of imbalance between the two misclassification error rates $P^D_{y|x}$ and $P^D_{x|y}$ resulting from the use of $D =$ QDF and $D =$ LDF when the parent populations are non-normal. Ideally one would want $P^D_{y|x} = P^D_{x|y}$ or a situation where a classifier can definitely confirm to provide lower misclassification rates for the samples of choice. Since this is not a possibility, the only logical option is to maintain the balance in tact. A remedy for this undesirable property of classifiers was given in Randles et al. (1978a) and Ng and

Randles (1983). In these papers, it is shown that balance is attained (asymptotically) through ranking of discriminant functions such as LDF and QDF while keeping the TPM relatively low. Randles et al. (1978b) use this ranking technique on robust linear and quadratic discriminant functions that use M estimates and they showed that the resulting classifier is robust compared to LDF and QDF and balance.

Chapter 3

Proposed Robust Projection Pursuit Based Allocation Schemes

3.1 Introduction

There are two main aspects of a projection pursuit technique - a projection index and the most efficient way to search for the best projection direction. Works in two-dimensional projection pursuit are discussed in Friedman and Tukey (1974), Jones (1983), Jones and Sibson (1987) and Friedman (1987), and suggestions are made by Yenyukov (1988), Yenyukov (1989). An effective algorithm for the same was provide by Posse (1990). He uses the chi-square measure as his projection index. He uses random search for projection directions to find local optima and restart the search process repeatedly to make sure they find the optimum. The two-dimensional problem might not be as complicated for the modern computational power, but as the dimension increases there is a need for more efficient projection pursuit algorithms. Posse (1992) used the algorithm given in Huber (1989) which is similar to that given by Posse (1990). A quickly generated random walk on the d-dimensional hypersphere S^d sweeps the entire space where areas of interest are identified to find the local optima. This happens to be very efficient as it saves computational time while still maintaining the efficiency of the algorithm.

For our projection pursuit based classifier, we use two-group transvariation (Gini, 1916) as projection index. We use similar principles of Posse (1992) where we sweep the high dimensional space and then find the projection direction that maximizes separation by minimizing two-group transvariation. In order for the sweep to be successful, the projection directions need to be uniformly distributed on the hypersphere. One such method that allows us to do this is NT-net given by Fang and Wang (1994).

3.2 NT-Net

To perform projection pursuit in $\mathbb{R}^d$ we need points that are "uniformly scattered" on the surface of the unit hypersphere $\mathbb{S}^{d-1}$ as given in Fang and Wang (1994).

- If $\mathbf{W} \sim NID_d(\mathbf{0}, I_d)$, then

  $$(\mathbf{W}'\mathbf{W})^{-1/2}\mathbf{W}$$

  is uniformly distributed on $\mathbb{S}^{d-1}$.

- If $\mathbf{V}_1, \ldots, \mathbf{V}_k$ is a number theoretic net (NT-net) on the $d-1$ dimensional unit cube $\mathbb{C}^{d-1}$, then

  $$\{\mathbf{W}_i = \mathcal{S}(\mathbf{V}_1, \ldots, \mathbf{V}_k)\ ,\ i = 1, \ldots, k\}\ ,$$

  where $\mathcal{S}$ is the spherical coordinate transformation operator, is an NT-net on $\mathbb{S}^{d-1}$.

Let $\Phi = (\phi_1, \ldots, \phi_{d-1})$ be a point on $\mathbb{C}^{d-1}$. We use the spherical coordinate transformation

$$X_j = \prod_{i=1}^{j-1} S_i C_j, \quad j = 1, ..., d-1,$$

$$X_d = \prod_{i=1}^{d-1} S_i,$$

where

$$S_i = sin(\pi\phi_i), \quad C_i = cos(\pi\phi_i) \quad i = 1, ..., d-2,$$

$$S_{d-1} = sin(2\pi\phi_{d-1}), \quad C_{d-1} = cos(2\pi\phi_{d-1}.)$$

Then $\mathcal{X} = (X_1, \ldots, X_d)$ is a point on $\mathbb{S}^{d-1}$. If $\Phi_1, \ldots, \Phi_k$ forms an NT-net on $\mathbb{C}^{d-1}$, then $\mathcal{X}_1, \ldots, \mathcal{X}_k$ forms an NT-net on $\mathbb{S}^{d-1}$. For example, to obtain an NT-net for $d = 3$, if $\{(v_{i1}, v_{i2})\ ,\ i = 1, \ldots, k\}$ is a NT-net on $[0, 1]^2$, then $\{(w_{i1}, w_{i2}, w_{i3})\ ,\ i = 1, \ldots k\}$ is a NT-net on $\mathbb{S}^2$ as given in Figure 3.1, where

$$\begin{aligned} w_{i1} &= 1 - 2v_{i1} \\ w_{i2} &= 2\sqrt{v_{i1}(1 - v_{i1})}\cos(2\pi v_{i2}) \\ w_{i3} &= 2\sqrt{v_{i1}(1 - v_{i1})}\sin(2\pi v_{i2}) \end{aligned}$$

To measure whether the points on the hypersphere are uniformly distributed, measures such as F-Discrepancy are used (Fang and Wang, 1994). Let $F(x)$ be a cdf in $\mathbb{R}^d$ and $\wp = x_k, k = 1, ..., n$ be a set of points on $\mathbb{R}^d$, then

$$\Delta^*_F(n, \wp) = \sup_{x \in \mathbb{R}^d} |F_n(x) - F(x)|$$

is called the F-Discrepancy of $\wp$ with respect to $F(x)$, where $F_n(x)$ is the empirical distribution of $x_1, ..., x_n$. F-Discrepancy is a measure of the representation of $\wp$ with

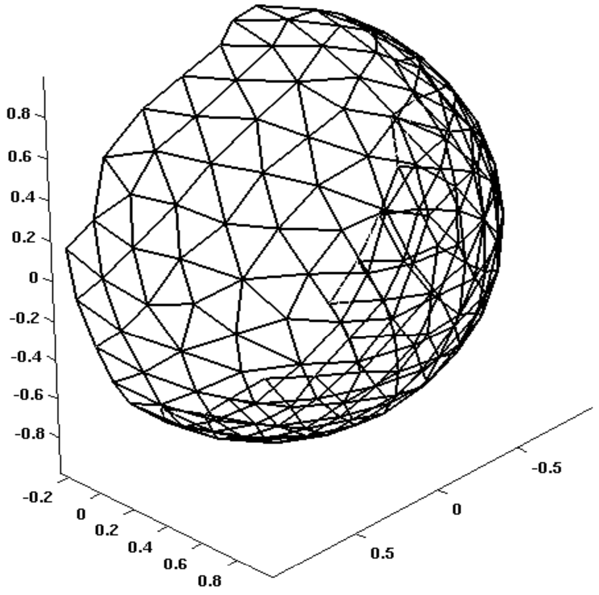

Figure 3.1: An NT-net on $\mathbb{S}^2$

respect to $F(x)$. We chose to use NT-net to provide us with our projection directions for our projection pursuit method because of this property.

The interested reader may find alternative ways of generating projection directions in Marsaglia (1972), where three different Monte Carlo point picking methods are given.

It is well known that projection pursuit based methods are computationally expensive. Although, advanced computational techniques can be effectively used to minimize the number of projections considered without affecting the efficiency of the technique (Filzmoser et al., 2006). Inspired by Posse (1992), we provide one such algorithm that would briefly sweep the entire space using a small number of projection directions and then localize to find the best projection direction that optimizes the projection index. We use two-group transvariation as a projection index as given in Montanari (2004).

Algorithm:

Step 1: Uniformly spread points that cover the entire unit hyper cube $\mathbb{C}^{d-1}$ are generated.

Step 2: These points on the unit hyper cube are spherically transformed onto a unit hyper sphere $\mathbb{S}^{d-1}$ as described above. These points are uniformly distributed as given in Fang and Wang (1994) using the F-discrepancy criteria.

Step 3: The projection index for each of the projections on the hyper sphere is calculated and the projection that provides the optimum projection index is chosen.

The optimum projection is the direction that produces maximum separation or minimum overlap between the training samples.

Step 4: In the case of more than one projection direction providing the optimum projection index, the projection direction that provides the most spread for the data is considered. One can choose a robust version of spread to make the algorithm more robust.

Step 5: The point on the unit hyper sphere that provides the projection direction in 'Step 4' is identified and easily traced back to the point on the unit hyper cube. In the unit hyper cube, the points immediately surrounding this point are used to create more points. The number of points generated here could be the same as in 'Step 1'.

Step 6: A grid of uniformly spread points is now generated on this localized region of the unit hyper cube.

Step 7: These points on the localized region of the unit hyper sphere are spherically transformed, which now form a unit hyper arc instead of a unit hyper sphere. This gives a localized view of the region that gave the optimum projection direction in the previous iteration.

Step 8: 'Step 3' through 'Step 7' are then repeated until a convergence criterion is reached. If transvariation probability is used as a projection index, the convergence criterion could be to continue until the projection index stops decreasing.

It has to be noted that the convergence criterion will take you to the projection direction that gives a local optimum projection index. For data that follows a certain unimodal shape, there will only be a global optimum for the projection index and the algorithm takes you to the projection direction that achieves this global optimum. Since transvariation is discrete, the algorithm ensures that there exists only one projection direction that provides the least two-group transvariation.

The number of points needed per dimension for this algorithm depends on the dimension under consideration and the computational power. While a thorough search would confirm an optimum projection index, Monte Carlo simulations done at the end of this chapter have shown us that for certain unimodal distributions, 5 projection directions per dimension seem to be more than sufficient. We tried the process with 7 and 9 projection directions per dimension but did not see any noticeable improvement. If 5 projection directions per dimension are used, the total number of projection directions would be 5^{d-1}, which means that as the dimension increases, the number of projection directions required also increases. The beauty of using NT-net for projection directions is that you do not have to look at the entire region of space at one time. You could break the entire space that needs to be explored into regions, which can be controlled easily by controlling the points on the unit hyper cube.

Once the vector $\hat{\mathbf{u}}_{opt}$ that gives us the most interesting view of the data in a single dimension is found using (2.7), we project the test samples in this direction and allocate the new observation into one of the two groups based on a particular allocation scheme.

3.3 Allocation Schemes

We will consider four different schemes of allocation in this study. The first method was proposed by Montanari (2004) and is based on projected distances of the new observation from the centers of the training samples. The second is a suggestion given in Montanari (2004), wherein a nonparametric allocation scheme was proposed and immediately abandoned. We will discuss the reason why she had to abandon her suggested scheme and we will provide two alternative allocation schemes which are completely nonparametric. These are based on the positions of the projected points relative to the projected centers of the training samples.

It must be noted that all the procedures given in this chapter are affine invariant. Invariance to translations is immediate. Moreover, any rotation of the data will result in the same rotation of the projection direction that gives the "most interesting" view and the resulting projected data do not depend on the coordinate system used. A formal argument showing the affine invariance of these procedures is found in Lemma 2.1 of Donoho and Gasko (1992) and Theorem 2.1 of Zuo and Serfling (2000).

3.3.1 Allocation Based on Distance

Given two independent random training samples $\mathbf{X}_1, \ldots, \mathbf{X}_{n_x}$ and $\mathbf{Y}_1, \ldots, \mathbf{Y}_{n_y}$ from Π_x and Π_y, respectively, defined on $\mathbb{R}^d$ ($d \geq 1$), a new observation $\mathbf{Z} = \mathbf{z}$ is classified in Π_x if $|\hat{\mathbf{u}}'_{opt}\mathbf{Z} - m_x(\hat{\mathbf{u}}_{opt})| < |\hat{\mathbf{u}}'_{opt}\mathbf{Z} - m_y(\hat{\mathbf{u}}_{opt})|$, otherwise classify it in Π_y. Here $m_x(\hat{\mathbf{u}}_{opt})$ and $m_y(\hat{\mathbf{u}}_{opt})$ are centers of the two projected groups. One may take either the mean or the median as a measure of location. We considered the median as given in Montanari (2004). Hereafter the classifier obtained using this allocation method will be referred to as Transvariation-Distance (TD) classifier.

This allocation scheme is based on distances from the projected center of the data from the projected new observation, which makes this allocation scheme parametric. Without regard for the shape of the distribution the data follows, the center is used as a reference for allocation. There seems to be no problem in allocation when the distributions under consideration are symmetric but as the distributions get away from this assumption of symmetry, the dependence of the method on parameters makes the scheme sensitive to this assumption. Montanari (2004) was aware of this problem and tried to fix this with a non-parametric version that she suggested in her research.

3.3.2 Allocation Based on Point-Group Transvariation

A nonparametric allocation option suggested by Montanari (2004) is based on the ranking of the new observation among the two samples. This utilizes the point group transvariation defined by Gini (1916) between the projected new observation and projected $\mathbf{X}$ and the projected new observation and projected $\mathbf{Y}$. Allocate a new observation $\mathbf{Z} = \mathbf{z}$ into Π_x if $T_x(\mathbf{Z}) > T_y(\mathbf{Z})$; otherwise, it is assigned to Π_y where

$$T_x(\mathbf{Z}) = \frac{1}{n_x} \sum_{i=1}^{n_x} I\{(\hat{\mathbf{u}}'_{opt}\mathbf{X}_i - \hat{\mathbf{u}}'_{opt}\mathbf{Z})(m_x(\hat{\mathbf{u}}_{opt}) - \hat{\mathbf{u}}'_{opt}\mathbf{Z}) < 0\} \qquad \text{and}$$

$$T_y(\mathbf{Z}) = \frac{1}{n_y} \sum_{i=1}^{n_y} I\{(\hat{\mathbf{u}}'_{opt}\mathbf{Y}_i - \hat{\mathbf{u}}'_{opt}\mathbf{Z})(m_y(\hat{\mathbf{u}}_{opt}) - \hat{\mathbf{u}}'_{opt}\mathbf{Z}) < 0\} \qquad (3.1)$$

As argued earlier, this allocation scheme is based on ranks. This gets rid of the non optimality problem that TD has when skewed distributions are considered. Although this allocation scheme makes this method completely nonparametric and works better than TD for skewed distributions, it does not perform as well for data with unequal sample sizes. This is due to the fact that an equal prior restriction is imposed by counting and we neglect group two(one) when we find the ranking of the new point in group one(two). So the priors are not necessarily taken into account and the effect shows in the misclassification error rate especially when the sample sizes are unequal. Montanari (2004) abandoned this scheme for this very reason. Adding to the problem of priors that this allocation scheme has is another problem of rather high likelihood of ties between T_x and T_y given in (3.1). The likelihood of ties is the most noticeable in equal sample sizes cases and these cases happen to be the only cases that this allocation scheme works efficiently. This problem of ties requires some kind of a tie breaking strategy. The one employed by us is to randomly assign the observation into one of the groups using a coin flip. The classifier obtained using this allocation scheme will be referred to as Point-Group Transvariation (PGT) classifier.

3.4 Proposed Symmetrized Allocation Scheme

The schemes, TD and PGT, one being parametric and the other being nonparametric, still have problems with skewed cases and problems with unequal sample size cases, respectively. In spite of being nonparametric, the PGT scheme did not include any information about the second group when finding point-group transvariation between the new observation and the first group. We needed to come up with a new allocation scheme that considers both the groups while finding a nonparametric alternative to the existing schemes.

We had a nonparametric way of looking at two groups at the same time: the two-group or group-group transvariation defined by Gini (1916). The next issue we faced was to somehow include the new observation in the calculation of the group-group transvariation and be able to use it for allocation. So we propose a new nonparametric alternative that is based on ranking of the new observations in the groups while considering both the groups.

To allocate a new observation $\mathbf{Z}$, we define $\mathbf{X}^* = \{\mathbf{X}_1, \ldots, \mathbf{X}_{n_x}, \mathbf{Z}\}$ and similarly define $\mathbf{Y}^* = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_{n_y}, \mathbf{Z}\}$. The idea is to find T_{x^*y}, the transvariation probability between $\mathbf{X}^*$ and $\mathbf{Y}$ given by

$$T_{x^*y} = \frac{1}{(n_x+1)n_y} \sum_{\mathbf{x}^* \in \mathbf{X}^*} \sum_{\mathbf{y} \in \mathbf{Y}} I\{(\hat{\mathbf{u}}'_{opt}\mathbf{x}^* - \hat{\mathbf{u}}'_{opt}\mathbf{y})(m_x(\hat{\mathbf{u}}_{opt}) - m_y(\hat{\mathbf{u}}_{opt})) < 0\} \qquad (3.2)$$

and T_{xy^*}, the transvariation probability between $\mathbf{X}$ and $\mathbf{Y}^*$ given by

$$T_{xy^*} = \frac{1}{n_x(n_y+1)} \sum_{\mathbf{x}\in\mathbf{X}} \sum_{\mathbf{y}^*\in\mathbf{Y}^*} I\{(\hat{\mathbf{u}}'_{opt}\mathbf{x} - \hat{\mathbf{u}}'_{opt}\mathbf{y}^*)(m_x(\hat{\mathbf{u}}_{opt}) - m_y(\hat{\mathbf{u}}_{opt})) < 0\} \tag{3.3}$$

and see the effect of the new observation on the quantities T_{x^*y} and T_{xy^*}. These two quantities have T_{xy} in common, where

$$T_{xy} = \frac{1}{n_x n_y} \sum_{\mathbf{x}\in\mathbf{X}} \sum_{\mathbf{y}\in\mathbf{Y}} I\{(\hat{\mathbf{u}}'_{opt}\mathbf{x} - \hat{\mathbf{u}}'_{opt}\mathbf{y})(m_x(\hat{\mathbf{u}}_{opt}) - m_y(\hat{\mathbf{u}}_{opt})) < 0\} \ , \tag{3.4}$$

is the transvariation probability between X and Y as given in (2.5). The quantity $(n_x+1)n_yT_{x^*y} - n_xn_yT_{xy}$ is the number of observations in group $\mathbf{X}$ with which the new observation transvariates. Similarly, the quantity $n_x(n_y+1)T_{xy^*} - n_xn_yT_{xy}$ is the number of observations in group $\mathbf{Y}$ with which the new observation transvariates. So, the difference $T_{x^*y} - T_{xy}$ is due to the addition of the new observation $\mathbf{Z}$ in $\mathbf{X}$ and the difference $T_{xy^*} - T_{xy}$ is due to the addition of the new observation $\mathbf{Z}$ to $\mathbf{Y}$. These differences can be thought of as the contributions that $\mathbf{Z}$ makes towards the transvariations if it were to belong in the groups in which it is placed. These differences could be positive or negative based on the location of the new observation but cannot both be positive or both negative for a particular new observation. If $\mathbf{Z}$ belongs to the group it was placed in, it will not contribute to the transvariation with the other group and thus creating a negative difference, else it would create a positive difference.

A negative difference indicates that the new observation **Z** is placed in the correct group. So a new observation $\mathbf{Z} = \mathbf{z}$ is allocated to Π_x if $T_{x^*y} - T_{xy} < T_{xy^*} - T_{xy}$, else it is allocated to Π_y. Since both the differences contain T_{xy}, regardless of the differences, we allocate the new observation to Π_x if $T_{x^*y} < T_{xy^*}$, else we classify it in Π_y.

This classifier takes care of the problems that the previous two allocations schemes could not take care of: the unequal sample sizes problem of PGT (this is done by considering both the groups) and the deviations from symmetric distributions problem of TD (this is done in the same way PGT did it). We will call this classifier Group-Group Transvariation (GGT) classifier.

3.5 Proposed Smoothed Allocation Scheme

As discussed above, PGT is a definite improvement in terms of dealing with the skewed distribution cases, but the problem of unequal sample sizes remains. One way of looking at it is to say that PGT does not consider both the groups at the same time and we came up with GGT. Another way of looking at the issue with PGT is to say that PGT looks at transvariations, which lacks smoothness in the sense that the individual votes are either 0 or 1 (Mojirsheibani, 2000). For instance, in Equation (3.1), the vote of each X in $T_x(z)$ is either 0 or $1/n_x$ while the vote of each Y in $T_y(z)$ is either 0 or $1/n_y$. Obviously, this presents a problem when $n_x \neq n_y$. A flexible procedure where the votes are allowed to take values between 0 and 1 in a way that is optimal is more appropriate.

Once the data is projected onto the optimum projection using projection pursuit, let us consider the point-group transvariation between a point c and a group $X_1, \ldots, X_{n_x}$ and without loss of generality let us say that $m_x < c$. Then the sample version of the point-group transvariation between c and $\mathbf{X}$ is given by

$$T_x(c) = \frac{1}{n_x} \sum_{i=1}^{n_x} I\{X_i - c < 0\}$$

which is a sign statistic (Hollander and Wolfe, 1999), a one-sample nonparametric test statistic. The problem is that an observation is treated as a transvariation regardless of whether X is barely greater than c or much higher than c. There seems to be no weight associated to the location. Similar argument can be made for non-transvariations where an observation is treated as a non-transvariation regardless of whether X is more or barely smaller than c.

There is a need for a function that weighs in the distance and the location as well to assign a number in the interval (0, 1). That is, a function that gives each transvariation a value between 0 and 1 based on the distance. So the indicator function needs to be replaced by a function that is always between 0 and 1 and continuous. The difference between a transvariation which is 0 or 1, and a smooth function that provides values between 0 and 1 is shown in Figure 3.2. A **CDF** seems to be a natural choice as a function here, although other functions can be used.

Classify $\mathbf{Z} = \mathbf{z}$ in Π_x if $T_x(\mathbf{z}) > T_y(\mathbf{z})$ where

$$T_x(c) = \frac{1}{n_x} \sum_{i=1}^{n_x} \left[1 - K_x\{(\hat{\mathbf{u}}'_{opt}\mathbf{X}_i - \hat{\mathbf{u}}'_{opt}\mathbf{z})(m_x(\hat{\mathbf{u}}_{opt}) - \hat{\mathbf{u}}'_{opt}\mathbf{z})\} \right] \qquad \text{and}$$

$$T_y(c) = \frac{1}{n_y} \sum_{j=1}^{n_y} \left[1 - K_y\{(\hat{\mathbf{u}}'_{opt}\mathbf{Y}_j - \hat{\mathbf{u}}'_{opt}\mathbf{z})(m_y(\hat{\mathbf{u}}_{opt}) - \hat{\mathbf{u}}'_{opt}\mathbf{z})\} \right], \tag{3.5}$$

else classify it in Π_y.

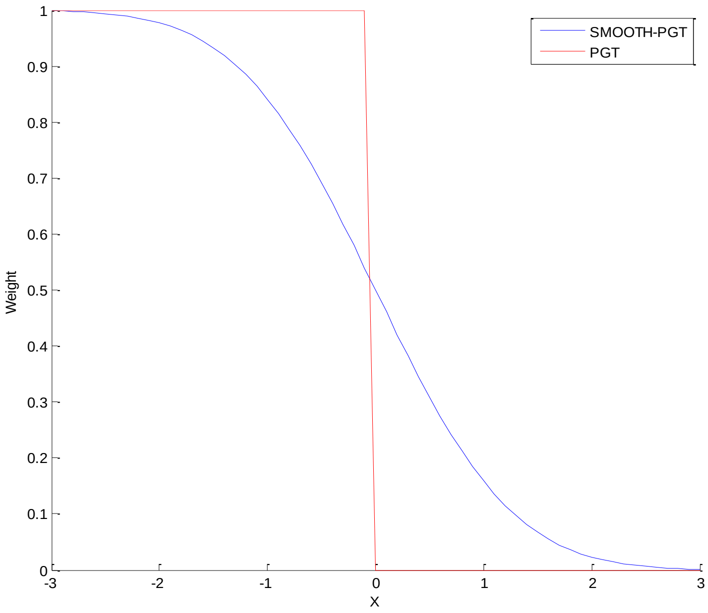

Figure 3.2: Graphical Depiction of Transvariation and Smoothed Transvariation

K_x and K_y need not necessarily be CDFs. They can be taken to be functions that satisfy certain requirements or any CDF defined on $\mathbb{R}$ with pdf k_x and k_y, respectively, that is continuous on $(-\delta, \delta)$ for some $\delta > 0$ and $k_x(0) > 0$ and $k_y(0) > 0$. It is not a requirement that both the samples need to follow a certain shape. Note that PGT uses the CDF of the bernoulli random variable $I(X \geq 0)$.

The conversion of a 0 or 1 transvariation into a number in the interval (0, 1) is based on a smoothing parameter. Smoothing works well in avoiding the problem that PGT had with allocation in the case of unequal sample sizes. Kernel smoothing was used for classification by Mojirsheibani (2000). In our study, we use the CDF of the t-distribution for K with degrees of freedom (df) as the smoothing constant. This gives us a wide range of CDFs with varying scale that could be used to fit various distributions.

For a data set with training and testing samples, the process of finding the optimum smoothing constant (df) is as follows:

Algorithm

Step 1: After finding the optimum projection direction using NT-net, the data are projected onto this direction. You would then allocate the testing sample using equation (3.5).

Step 2: To allocate, you need the df that defines the t-CDF, which provides the best smoother for that particular shape of data. It makes a lot of sense to use two different smoothing df, one for each group of data. We use the training sample to train and find the best smoother for each group.

Step 3: Finding the smoother theoretically is a very difficult problem. In fact, there exists no closed form solution when the t-CDF is used in place of the indicator function in transvariation. Instead we use a bivariate grid, containing $df(df_1, df_2)$, one for each group.

Step 4: We then use the training data and apply leave-one-out cross validation to find the probability of misclassification for each possible pair of df. The combination that provided the least probability of misclassification is then picked as the pair of smoothing constants.

Step 5: We finally use equation (3.5) with the smoothing constants found in Step 4, to find the total probability of misclassification for this data.

When the data are presented as a single sample, as is the case for real data sets, we remove one observation from the data and try to use the rest of the data as a training sample to allocate this observation. This is like leave-one-out cross validation. So, we end up using a double one-at-a-time cross validation, once for finding the misclassification for the data and again to find the best set of smoothing df for the training sample.

We tried the normal cdf which worked really well for distributions that are normal or near normal. As the distributions deviated from normality, the efficiency of the normal cdf with the spread as the smoother went down. Based on our simulations, this was partly because normal distribution with sigma as a smoother, lacks the range to cover a variety of distributions. This was evident when most of the smoothing constants ended up being very close to zero, which makes the normal distribution almost like a point mass and hence a transvariation. In such cases, smooth becomes PGT. We chose to use t-cdf with *df* as a smoother for its wide range. We are convinced that a better smoother exists out there and we leave the solution of finding it as an open problem.

We refer to the classifier that uses this allocation scheme as smooth-PGT (SPGT). This classifier has all the positives of PGT but gets rid of the problem with PGT, that is, with unequal sample sizes. By counting transvariations, PGT puts a restriction of equal priors. As discussed above, transvariation with one group is scaled by $\frac{1}{n_x}$, while the transvariation with the other group is scaled by $\frac{1}{n_y}$. This creates a problem when the sample sizes are not equal, especially when the sample sizes are highly unequal. In the case of the SPGT allocation scheme, each transvariation is smoothed, in our case using a t-CDF. This takes care of the shape of the distributions by weighing each point based on their relative position and the distance from the new observation. The two smoother $df's$ defining the t-CDf's, which provide the least misclassification are found using the training samples. This process of finding the best smoother that smoothes the data, closes the gap between the difference in sample sizes.

3.6 Application on Real Data and Monte Carlo Simulation

We would like to show the optimality of the proposed classifiers by applying the methods on some real data sets and a variety of simulation settings.

3.6.1 Application on Real Data Sets

The data sets that we chose to consider are

Leukemia : This dataset is given by Golub et al. (1999) and comes from a study of gene expression in two types of acute leukemias: acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). The dataset is available at

`http://www.genome.wi.mit.edu/MPR`

Gene expression levels were measured using Affymetrix high-density oligonucleotide arrays containing 7129 human genes. The data are comprised of 47 cases of ALL (38 B-cell ALL and 9 T-cell ALL) and 25 cases of AML. We used the method of gene selection given by Nguyen and Rocke (2002) that uses a *t*-statistic to select expressed genes. After preprocessing, the data are summarized by a 72x1751 matrix. To reduce the dimension of the data set, we considered the first 10 principal components.

Colon : This data set contains gene expression in 40 tumor and 22 normal colon tissue samples analyzed with an Affymetrix oligonucleotide array (Alon et al., 1999). The expression level of 6500 genes were measured and the 2000 genes with the highest minimal intensity were retained. We again applied the gene selection method given by Nguyen and Rocke (2002) to select the genes that exhibit maximum variation among the 62 tissues. The resulting data matrix contained 250 genes. We then kept the first 9 principal components for the purpose of classification.

Fisher's Iris : This is the classic Iris dataset given by Fisher (1936). It consists of three species (Setosa, Versicolor, and Virginica) of Iris flowers with 50 observations of each species. The dataset contains four variables: sepal length, sepal width, petal length, and petal width. We wish to identify a new Iris flower as Versicolor or not based on measurements on these four variables. We considered the original data as well as a contaminated version where the contamination is done by placing a single outlier in the Versicolor group in a similar manner as Crimin et al. (2007).

We will consider the proposed methods, GGT and SPGT and as comparison we will consider the allocation scheme, TD, proposed by Montanari (2004) and her suggested allocation, PGT. We will also consider maximum depth classifier, MaxD, based on L_1 depth given by Ghosh and Chaudhuri (2005) and Jörnsten (2004) and the classical LDF and QDF. The results are presented in Table 1 given in the appendix.

- For the original Iris data, QDF and MaxD gave the lowest TPM at 3.33%. The least optimal performer was TD with a misclassification error rate of 31.33%. When a cluster of five outliers was added to the Iris data, GGT gave the lowest TPM at 3.87% with PGT really close at 3.97%. The misclassification error rate for SPGT went down slightly, while the error rates for MaxD, LDF and QDF exploded. For LDF and MaxD, the error rate was doubled and for QDF, the error rates increased almost 4 times.

- For the Leukemia data, GGT happened to be the method that provided the least TPM of 1.39%; The SPGT method had the next best misclassification error rate at 2.78%. PGT is close with 2.9% while TD happened to be the least optimal among the methods considered with an error rate of 11.11%. MaxD, LDF and QDF shared the same error rate at 4.17%.

- Smooth-PFT, GGT, TD and LDF shared the lowest TPM at 12.9%. The least optimal method being MaxD with a misclassification error rate of 17.74%.

3.6.2 Monte Carlo Simulation

The common simulation settings used by authors in literature looks at normal distributions and as a deviation they include outliers, look at distributions with heavy tail and log-normal distribution as a case of skewed distribution. In all those cases, most often than not, authors choose to use the same distributions for both the groups and/or keep most of the other aspects simple. We perform a very extensive Monte Carlo simulation to study the optimality (in terms of misclassification error rates) of the proposed classification procedures under a variety of distributional settings:

- Homoscedastic (vs) Heteroscedastic

- Equal sample sizes (vs) Unequal sample sizes

- Same distributions (vs) Different distributions for the two groups

- Symmetric (vs) Skewed distributions

- Normal tail (vs) Long tailed symetric distributions

- Combinations therein

The simulation study follows the same setup given in Montanari (2004). We start off by generating training samples of the given sizes which are used to formulate the classification rules. Testing samples of size 1000 from each group are then generated and the misclassification error rates are calculated by computing the proportion of misclassified testing sample observations in each group. This process is then repeated 50 times and the mean and standard error of the misclassification error rates are computed.

We consider Cauchy (C, which is t with 1 degree of freedom), t with 2 degrees of freedom (t_2), Normal (N) distributions, and log-normal (LN) distributions. Training samples of equal $(150, 150)$ and unequal $(50, 250)$ sizes were generated from the distributions.

We consider four-dimensional distribution to generate the data. We consider centers $(0, 0, 0, 0)'$ and $(2, 0, 0, 0)'$ and covariance matrices $I = I_4$ and

$$W = \begin{pmatrix} 1 & -1 & -1 & 1 \\ -1 & 2 & 2 & .25 \\ -1 & 2 & 3 & .5 \\ 1 & .25 & .5 & 4 \end{pmatrix}.$$

In the reporting of the results we use the notation $K(A), H(B)$ to represent distributions K and H (K and H are C, t_2, N or LN) with covariance matrices A and B (A and B are $I = I_4$, W), respectively. We consider all the methods as mentioned in the real data sets. A summary of the results is given in Table 2 given in the appendix. The following observations are noted:

- A look at the results for the PGT classifier indicates that it is not optimal for almost all of the unequal sample size cases confirming Montanari's suspicion that led her to abandon the point-group transvariation allocation scheme. GGT and SPGT classifiers have error rates comparable for equal sample size cases compared to the PGT classifier, lower error rates in almost all cases for the unequal sample size cases and much lower error rates in most cases for the unequal sample size cases. For these reasons, we will not be considering PGT as a contender for misclassification error rate.

- Considering the standard errors of the unequal sample size cases, MaxD and PGT classifiers are less precise than the other classifiers, MaxD being the least precise classifier. Except in a few cases, PGT is less precise than GGT and SPGT.

- Error rates for GGT and SPGT classifiers are comparable to TD for symmetric and same distributions. GGT, SPGT and TD have better error rates compared to LDF, QDF and MaxD for heavy tailed distributions (Cauchy and t(2)) while for the normal case, GGT and SPGT have error rates comparable with LDF, QDF and TD classifiers but superior to MaxD classifier.

- For skewed but same distribution cases, SPGT and GGT are the best methods among the methods considered in terms of misclassification error rate and standard error. The error rates and standard errors of SPGT and GGT are comparable.

- For different but homoscedastic distributions, there are cases where SPGT is the best method, where GGT is the best method, where TD is the best method. One of the three methods SPGT, GGT or TD is the best method in these cases. Best in terms of misclassification error rate and standard error. For these settings, it can also be noted that GGT acts up for the unequal sample size cases. The error rates for GGT decreases for some cases with unequal sample sizes and increases substantially for some other cases.

- For different and heteroscedastic distributions, it can be noted that QDF does fairly better than it usually does. As mentioned in Chapter 2, QDF works better for heteroscedastic cases, but the different distributions inflates the error rates in some cases. Except for QDF, SPGT happens to be the best method among the methods considered. The only exception is GGT for some unequal sample size cases and where log-normal distribution is considered with equal sample sizes.

Chapter 4

Robust and Balanced Classification

4.1 Introduction

Extensive research has been done in the area of classification. To name a few of the main categories that authors looked at are parametric methods, non-parametric alternatives to the parametric methods, using robust estimates in place of the sensitive ones. These methods try to create a method that provides the lowest possible TPM. Not much has been done in regard to the issue of balance of the misclassification error rates within each group. This issue was briefly talked about in Lachenbruch et al. (1973) and then a solution was provided by Ng and Randles (1983) where they rank LDF and QDF to create balance in the misclassification error rates, that is, $P^D_{y|x} = P^D_{x|y}$.

The issue of balance needs more importance than it has been given in literature in terms of research done on this topic. The issue of balance becomes very pertinent in situations where one of misclassifications ($P^D_{y|x}$, $P^D_{x|y}$) is costlier than the other. For example, in a two group classification problem with a group of patients having cancer and the another group not having cancer, a misclassification of a cancer patient into a noncancerous group could prove costly in terms of life. Ideally there is a need for a method that can control the ratio of the misclassification error rates at a required level. When the investigator does not have any information on the costs of misclassification and prior probabilities, then it is best to maintain the balance between the two misclassification error rates. Some existing methods can provide a low TPM but there are not many methods, especially robust methods that can create this balance while minimizing the overall misclassification error rate. We propose one such method that can maintain the balance in the two misclassification error rates while controlling the overall TPM. We show that balance can be achieved as long as ranking is incorporated and both groups are considered while allocating the new observation (Randles et al., 1978a).

4.2 Ranking Discriminant Function

Given two independent random training samples $\mathbb{X}$ of size n_x given by $\mathbf{X}_1, \ldots, \mathbf{X}_{n_x}$ from population Π_x with distribution F and $\mathbb{Y}$ of size n_y given by $\mathbf{Y}_1, \ldots, \mathbf{Y}_{n_y}$ from Π_y with distribution G, defined on $\mathbb{R}^d$ $(d \geq 1)$. Let F_{n_x} and G_{n_y} be the empirical distribution functions of $\mathbb{X}$ and $\mathbb{Y}$, respectively. Suppose, as before, that we have an absolutely continuous and real valued discriminant function D such that $\mathbf{Z} = \mathbf{z}$ is classified in Π_x if $D(\mathbf{z}; F, G) > 0$ and in Π_y otherwise. Now let us define a rank discriminant function RD as

$$\begin{aligned} RD(\mathbf{z}; F, G) = P\left\{D(\mathbf{z}; F, G) \geq D(\mathbf{X}; F, G) \mid \mathbf{X} \sim F\right\} \\ - P\left\{D(\mathbf{z}; F, G) \leq D(\mathbf{Y}; F, G) \mid \mathbf{Y} \sim G\right\} . \end{aligned} \tag{4.1}$$

Naturally, one classifies $\mathbf{Z} = \mathbf{z}$ in Π_x if $RD(\mathbf{z}; F, G) > 0$ and in Π_y otherwise. This classifier is looking at whether the new observation belongs more to $\mathbb{X}$ or to $\mathbb{Y}$ and the calculation depends on how $D(.)$ is chosen.

To define the sample version, we will form two augmented samples by placing the new observation $\mathbf{Z} = \mathbf{z}$ in $\mathbb{X}$ forming $\mathbb{X}^* = \{\mathbf{X}_1, \ldots, \mathbf{X}_{n_1}, \mathbf{Z}\}$ and in $\mathbb{Y}$ forming $\mathbb{Y}^* = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_{n_2}, \mathbf{Z}\}$. Let $F^*_{n_x+1}$ and $G^*_{n_y+1}$ be the empirical distribution functions of $\mathbb{X}^*$ and $\mathbb{Y}^*$, respectively. We then have

$$\begin{aligned} RD(\mathbf{z}; F_{n_x}, G_{n_y}) = & \frac{1}{n_x+1} \sum_{\mathbf{x} \in \mathbb{X}^*} I\left\{ D(\mathbf{z}; F^*_{n_x+1}, G_{n_y}) \geq D(\mathbf{x}; F^*_{n_x+1}, G_{n_y}) \right\} \\ & - \frac{1}{n_y+1} \sum_{\mathbf{y} \in \mathbb{Y}^*} I\left\{ D(\mathbf{z}; F_{n_x}, G^*_{n_y+1}) \leq D(\mathbf{y}; F_{n_x}, G^*_{n_y+1}) \right\}, \end{aligned} \tag{4.2}$$

where $I\{A\}$ is the indicator function of the event A. We would then allocate $\mathbf{Z} = \mathbf{z}$ to Π_x if $RD(\mathbf{z}; F_{n_x}, G_{n_y}) > 0$ and to Π_y otherwise.

Under certain mild regularity conditions given in Section 4 of Ng and Randles (1983), the two misclassification error rates of $RD(\mathbf{z}; F_{n_x}, G_{n_y})$ are asymptotically equal or controlled to be a specific constant r. In particular, assume the regularity conditions given below are true:

1. The sample versions of the discriminant functions must be symmetric in their argument. That is,

$$D(\cdot|\mathbf{X}_1, \ldots, \mathbf{X}_{n_x}; \mathbf{Y}_1, \ldots, \mathbf{Y}_{n_y}) = D(\cdot|\mathbf{X}_{s_1}, \ldots, \mathbf{X}_{s_{n_x}}; \mathbf{Y}_{t_1}, \ldots, \mathbf{Y}_{t_{n_y}})$$

for any permutation $(s_1, \ldots, s_{n_x})$ of $(1, \ldots, n_x)$ and $(t_1, \ldots, t_{n_y})$ of $(1, \ldots, n_y)$

2. For each z $\in \mathbb{R}^d$,

$$D(\mathbf{z}; F_{n_x}, G) \xrightarrow{\mathcal{P}} D(\mathbf{z}; F, G) \quad \text{and} \quad D(\mathbf{z}; F, G_{n_y}) \xrightarrow{\mathcal{P}} D(\mathbf{z}; F, G) \; ,$$

   where $D(\cdot)$ is a real valued function defined on $\mathbb{R}^d$.

3. For all $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^d$, we have

$$D(\mathbf{z}_1; F, G) < D(\mathbf{z}_2; F, G) \quad \text{iff} \quad D(\mathbf{z}_1; G, F) > D(\mathbf{z}_2; G, F) \; .$$

4. $\mathbf{Z}$, $\mathbf{X}_i$'s and $\mathbf{Y}_j$'s are all mutually independent

5. The random variables $D(\mathbf{X}; F, G)$ and $D(\mathbf{Y}; F, G)$ are absolutely continuous

6. The equalities

$$\begin{aligned} &P\left\{F_{n_x}(D(\mathbf{X}; F, G)) = rG_{n_y}(D(\mathbf{X}; G, F))\right\} = 0 \; , \quad \text{and} \\ &P\left\{F_{n_x}(D(\mathbf{Y}; F, G)) = rG_{n_y}(D(\mathbf{Y}; G, F))\right\} = 0 \; , \end{aligned}$$

   hold for some $r \in \mathbb{R}^+$.

Then we have

$$P\{RD(\mathbf{Z}; F, G) < 0 \mid \mathbf{Z} \sim F\} = rP\{RD(\mathbf{Z}; F, G) > 0 \mid \mathbf{Z} \sim G\} \; .$$

The rank discriminant functions that we will study are:

RL: This uses the LDF $L(\mathbf{z}; F, G)$ given in (2.3) in place of $D(\mathbf{z}; F, G)$ in (4.1). This would be the method proposed by Randles et al. (1978a) where he ranks LDF.

RQ: This uses the QDF $Q(\mathbf{z}; F, G)$ given in (2.4) in place of $D(\mathbf{z}; F, G)$ in (4.1).This would be the method proposed by Randles et al. (1978a) where he ranks QDF.

RS: This uses the ranking of the L_1 depth function defined in (2.8). However, to comply with the regularity conditions of Ng and Randles (1983) regarding effective sample separating ability of the discriminant function, we will use

$$\tilde{S}(\mathbf{z}; F, G) = \frac{1}{\mathcal{D}_1(\mathbf{z}; G)} - \frac{1}{\mathcal{D}_1(\mathbf{z}; F)}$$

in place of $D(\mathbf{z}; F, G)$ in (4.1). This is compatible with all the other discriminant functions we use since $1/\mathcal{D}_1$ is some kind of a measure of distance from the center of the distribution. If Mahalanobis depth is used the distance measure would become Mahalanobis distance and the classifier using this distance would exactly be RQ. In fact any robust affine invariant measure of distance can be used in place of this distance and the method still works. The more robust the distance, the more robust the classifier.

RM: Here we use the QDF given in (2.4) in place of $D(\mathbf{z}; F, G)$ in (4.1). However, in the sample version, we will use the minimum covariance determinant (MCD) estimators of location and covariance matrix (Rousseeuw, 1984, 1985) as done in

Hubert and Van Driessen (2004). The quadratic discriminant function of Hubert and Van Driessen (2004) will be denoted by $M(\mathbf{z}; F, G)$ as given in Equation (2.11). We use the FAST-MCD algorithm of Rousseeuw and Van Driessen (1999) with the default number of observations $0.75n_x$ and $0.75n_y$ for MCD computations. This assumption is safe to assume as long as you do not suspect more than 25% of the data to be outliers. The MCD estimates are essentially trimmed mean and covariance structure as they are looking at the central 75% of the data only. A very good property of MCD estimates is that a method using these estimates will remain affine invariant as the MCD estimates are affine invariant.

RL and RQ are ranking LDF and QDF respectively as given in Randles et al. (1978a). Although ranking is nonparametric, the base, as in the quantities being ranked are completely parametric. That makes this method semi-parametric. We prove via simulation in Section 4.3 that although ranking makes the method robust, the quantities being ranked matters and controls the overall TPM. Inspired by Randles et al. (1978a), we prove via simulation that the ranked methods are more balanced than the unranked counterparts. We also show that RS and RM are more robust than RL and RQ due to the fact that RL and RQ have a parametric base that is sensitive to deviations and can only be so much better under deviations.

4.3 Application on Real Data and Monte Carlo Simulation

We would like to show the optimality of the proposed classifiers by applying the methods on some real data sets and a variety of simulation settings.

4.3.1 Application on Real Data Sets

We will use the same three datasets as described in Subsection 3.6.1. We will consider the proposed classifiers, RM and RS, which are robust ranked versions of the unranked classifiers, M and S. M is the classifier created by replacing the parametric quantities in QDF with MCD estimates. S is the maximum depth classifier based on L_1 depth given by Ghosh and Chaudhuri (2005) and Jörnsten (2004). We consider the classical L (LDF) and Q (QDF) and their ranked versions RL and RQ given by Randles et al. (1978a). The results are shown in Table 3 given in the appendix.

- For the original Iris data, Q gave the lowest TPM at 4% but the two misclassification error rates remain unbalanced at 1% and 10%. The method that provided the most balance was RM where each misclassification error rate is equal to 6%. The worst performer was L with misclassification error rates of 28% and 26%. When an outlier was added to the Iris data, M and RM gave the lowest TPM at 8% with RM giving greater balance (8%, 7.8%) than M (6%, 11.8%).

- For the Leukemia data, RL and L gave the lowest TPM of 4.2%; however, the error rates of L were unbalanced (2.1%, 8%) compared to those of RL (4.3%, 4%). All the remaining ranked discriminant functions gave a TPM of 8.3%. The method that performed poorly for this data was M with TPM of 12.5%.

- Considering the Colon data, both RM and M yielded TPM equal to 9.7% outperforming all other classifiers. Once again, RM had balanced error rates (9.1%, 10%) compared to M (13.6%, 7.5%). The least optimal method for this data was S with unbalanced error rates (40.9%, 10%) giving a TPM of 21%.

We also performed leave-one-out cross validation for the Leukemia and Colon data after using the gene selection scheme given by Dudoit et al. (2002) as well as one that uses a Wilcoxon statistic in place of the t-statistic. The results were very similar and hence not reported here.

4.3.2 Monte Carlo Simulation

For the same reasons mentioned in Subsection 3.6.2, we perform a very extensive Monte Carlo simulation to study the optimality (in terms of misclassification error rates) of the proposed classification procedures under a variety of distributional settings:

- Homoscedastic (vs) Heteroscedastic
- Equal sample sizes (vs) Unequal sample sizes
- Same distributions (vs) Different distributions for the two groups
- Normal tail (vs) heavy tailed symmetric distributions

- Sparse data (vs) Dense data

- Combinations therein

The simulation study follows the same setup given in Montanari (2004). We start off by generating training samples of the given sizes which are used to formulate the classification rules. Testing samples of size 1000 from each group are then generated and the misclassification error rates are calculated by computing the proportion of misclassified testing sample observations in each group. This process is then repeated 50 times and the mean and standard error of the misclassification error rates are computed.

We considered a two-dimensional case in addition to the four-dimensional case considered for the projection pursuit methodology to study the effect of data abundance with the misclassification error rate.

We consider Cauchy (C, which is t with 1 degree of freedom), t with 2 degrees of freedom (t_2), and Normal (N) distributions. Training samples of equal $(50, 50)$ and unequal $(25, 75)$ sizes were generated from the distributions.

In the two-dimensional setting, we used centers $(0, 0)'$ and $(2, 0)'$ for the two distributions whereas the covariance matrices considered are the independence case I_2 and a correlated data case

$$V = \begin{pmatrix} 1 & 1 \\ 1 & 4 \end{pmatrix}.$$

For the four-dimensional study we considered centers $(0,0,0,0)'$ and $(2,0,0,0)'$ and covariance matrices I_4 and

$$W = \begin{pmatrix} 1 & -1 & -1 & 1 \\ -1 & 2 & 2 & .25 \\ -1 & 2 & 3 & .5 \\ 1 & .25 & .5 & 4 \end{pmatrix}.$$

In the reporting of the results we use the notation $K(A), H(B)$ to represent distributions K and H (K and H are C, t_2, or N) with covariance matrices A and B (A and B are I_2, I_4, W, or V), respectively. Thus, in our study, the two groups could come from the same distribution with different means and the same covariance matrix (homoscedastic), the same distribution with different means and different covariance matrices (heteroscedastic), and/or two different distributions.

A summary of the results for the two-dimensional case is given in Table 4 whereas Table 5 (given in the appendix) contains the results from the four-dimensional study. We infer the following from the results:

- S *versus* RS: S is heavily unbalanced, especially when the sample sizes are not the same, the tail thickness of one or more of the distributions increases, the distributions are heteroscedastic, or the two groups do not share the same distribution. The balance of RS is not affected by any of these situations. RS generally has lower TPM than S, and the difference is pronounced when at least one of the distributions is t_2 or Cauchy. Moreover, RS has better consistency of correct classification in that the standard errors of its misclassification error rates are generally much lower in comparison to S.

- *L versus RL*: As expected, L gives poor performance in the case of heteroscedastic populations. Outside of balancing the misclassification error rates, RL does not appear to improve the TPM of L with the exception of the Cauchy distribution cases. In fact, RL gives worse TPM than L when two different distributions are considered and one of the distributions is normal.

- *Q versus RQ*: The misclassification error rates of Q are severely unbalanced especially when one of the distributions is heavy tailed. In addition to providing balance, RQ gives lower error rates than Q in heavy tailed (C or t_2) situations. The standard errors of the misclassification error rates of Q are three to four times higher than those of RQ when both distributions are heavy tailed.

- *M versus RM*: The TPM of RM is generally lower than that of M in the case of heavy tailed distributions. This is especially visible in the more sparse four-dimensional case. Otherwise, the TPMs of RM and R are comparable, with RM doing a better job of maintaining the balance between the misclassification error rates. The imbalance between the error rates of M increases when the two distributions are different and at least one of them is heavy tailed. Moreover, the standard errors of the misclassification error rates of M are 1.5 to 2 times larger than those of RM in the cases where a Cauchy distribution is involved.

- In general, *RM* and *M* provide superior performance in terms of TPM when a heavy tailed distribution is involved. *RM* gives the best performance of all the methods studied in terms of TPM, balance, and standard errors when both the distributions are Cauchy. This is true in both homoscedastic and heteroscedastic cases.

- Not surprisingly, as shown in Randles et al. (1978a), ranking provides balance between the group misclassification error rates. Moreover, in the cases where heavy tailed or two different distributions are considered, ranking itself appears to provide added optimality (smaller TPM and standard errors) even when the original method is robust.

Chapter 5

Influence Function

5.1 Introduction

The focus of this chapter is the analysis of the influence functions of some of the discriminant functions that were introduced in earlier chapters. As discussed in Hubert and Van Driessen (2004), the robustness of a classifier directly depends on the robustness of the discriminant function used. The influence on the LDF of a single point of perturbation added to one of the two populations was studied by Campbell (1978). Croux and Dehon (2001) studied the influence function of the LDF where the underlying distributions are assumed to be multivariate normal and where both populations are contaminated. Croux et al. (2008) followed a similar approach but with a penalty term included to make the classifier optimal. Recently Huang et al. (2007) studied the pair-perturbation influence function of the LDF where a pair of points of perturbation are included in one of the two populations.

On the other hand, the influence function of quadratic discrimination appears to be slightly more complicated. Croux and Joossens (2005) studied the influence of perturbing a point in one of the populations on the misclassification error rate of quadratic discriminant analysis. Although more complex, this is QDF version of the work of Croux and Dehon (2001) where underlying normality is assumed.

The purpose of this chapter is two-fold: to provide the influence function for the QDF thus filling this gap in the literature and to provide the influence function for the rank based discriminant function introduced in Chapter 4. Both of these are given without making any underlying distributional assumptions.

We will suppose that both distributions are ϵ-contaminated (Van Ness and Yang, 1998) and define

$$F_{\mathbf{y},\epsilon} = (1-\epsilon)F + \epsilon\Delta_{\mathbf{y}} \qquad \text{and} \qquad G_{\mathbf{y},\epsilon} = (1-\epsilon)G + \epsilon\Delta_{\mathbf{y}} \ ,$$

where $\Delta_{\mathbf{y}}$ is the distribution function of the point mass at $\mathbf{y}$. Now if $T(F,G)$ is a functional, then the influence function is defined as

$$IF(\mathbf{y}; T(F,G)) = \lim_{\epsilon \downarrow 0} \frac{T(F_{\mathbf{y},\epsilon}, G_{\mathbf{y},\epsilon}) - T(F,G)}{\epsilon} \ . \tag{5.1}$$

5.2 IF for QDF

The quadratic discriminant function is given in Equation (2.4) that we will reproduce here using a functional notation for convenience

$$\begin{aligned} Q(\mathbf{z}; F, G) = \ln\left(\frac{|\Sigma(G)|}{|\Sigma(F)|}\right) &-(\mathbf{z}-\mu(F))'[\Sigma(F)]^{-1}(\mathbf{z}-\mu(F))+ \\ &(\mathbf{z}-\mu(G))'[\Sigma(G)]^{-1}(\mathbf{z}-\mu(G)) \ . \end{aligned}$$

Define the function

$$\phi_F(\mathbf{x},\mathbf{v}) = (\mathbf{x}-\mu(F))'[\Sigma(F)]^{-1}(\mathbf{v}-\mu(F)) \; .$$

The following theorem gives the influence function of the QDF.

Theorem 5.1. *For a fixed value of* $\mathbf{z}$*, the influence function of* $Q(\mathbf{z};F,G)$ *is given by*

$$\begin{aligned} IF(\mathbf{y};Q(\mathbf{z};F,G)) = \phi_G(\mathbf{z},\mathbf{z}) - 2\phi_G(\mathbf{z},\mathbf{y}) - \phi_G^2(\mathbf{z},\mathbf{y}) - \phi_F(\mathbf{z},\mathbf{z}) + 2\phi_F(\mathbf{z},\mathbf{y}) + \\ \phi_F^2(\mathbf{z},\mathbf{y}) + \phi_G(\mathbf{y},\mathbf{y}) - \phi_F(\mathbf{y},\mathbf{y}) \; . \end{aligned}$$

Proof. Write

$$Q(\mathbf{z};F_{\mathbf{y},\epsilon},G_{\mathbf{y},\epsilon}) = \ln\left(\frac{|\Sigma(G_{\mathbf{y},\epsilon})|}{|\Sigma(F_{\mathbf{y},\epsilon})|}\right) - \phi_{F_{\mathbf{y},\epsilon}}(\mathbf{z},\mathbf{z}) + \phi_{G_{\mathbf{y},\epsilon}}(\mathbf{z},\mathbf{z}) \; .$$

Straight forward algebra shows that $\mu(F_{\mathbf{y},\epsilon}) = \mu(F) + \epsilon(\mathbf{y}-\mu(F))$ and $\Sigma(F_{\mathbf{y},\epsilon}) = (1-\epsilon)\Sigma(F) + \epsilon(\mathbf{y}-\mu(F))(\mathbf{y}-\mu(F))'$. As shown in Campbell (1978), writing $\mu = \mu(F), \Sigma = \Sigma(F)$, and $\mathbf{w} = \mathbf{y}-\mu(F)$, we have

$$\begin{aligned} [\Sigma(F_{\mathbf{y},\epsilon})]^{-1} &= (1-\epsilon)^{-1}\left[\Sigma^{-1} - \frac{\epsilon\Sigma^{-1}\mathbf{w}\mathbf{w}'\Sigma^{-1}}{(1-\epsilon)+\epsilon\mathbf{w}'\Sigma^{-1}\mathbf{w}}\right] \\ &= (1+\epsilon)\Sigma^{-1} - \epsilon\Sigma^{-1}\mathbf{w}\mathbf{w}'\Sigma^{-1} \end{aligned}$$

up to order ϵ. Thus

$$\begin{aligned}\phi_{F_{\mathbf{y},\epsilon}}(\mathbf{z},\mathbf{z}) &= \{(\mathbf{z}-\mu)-\epsilon\mathbf{w}\}' \left[(1+\epsilon)\Sigma^{-1}-\epsilon\Sigma^{-1}\mathbf{w}\mathbf{w}'\Sigma^{-1}\right]\{(\mathbf{z}-\mu)-\epsilon\mathbf{w}\} \\ &= (1+\epsilon)\phi_F(\mathbf{z},\mathbf{z})-2\epsilon(1+\epsilon)\phi_F(\mathbf{z},\mathbf{y})+\epsilon^2(1+\epsilon)\phi_F(\mathbf{y},\mathbf{y})-\epsilon\phi_F^2(\mathbf{z},\mathbf{y})+ \\ &\qquad 2\epsilon^2\phi_F(\mathbf{z},\mathbf{y})\phi_F(\mathbf{y},\mathbf{y})-\epsilon^3\phi_F^2(\mathbf{y},\mathbf{y})\ .\end{aligned}$$

This gives

$$\left[\frac{\partial}{\partial\epsilon}\phi_{F_{\mathbf{y},\epsilon}}(\mathbf{z},\mathbf{z})\right]_{\epsilon=0} = \phi_F(\mathbf{z},\mathbf{z})-2\phi_F(\mathbf{z},\mathbf{y})-\phi_F^2(\mathbf{z},\mathbf{y})\ . \tag{5.2}$$

Using similar steps

$$\left[\frac{\partial}{\partial\epsilon}\phi_{G_{\mathbf{y},\epsilon}}(\mathbf{z},\mathbf{z})\right]_{\epsilon=0} = \phi_G(\mathbf{z},\mathbf{z})-2\phi_G(\mathbf{z},\mathbf{y})-\phi_G^2(\mathbf{z},\mathbf{y})\ . \tag{5.3}$$

Now using a result of Golberg (1972) [p. 1125, Equation (9)] we find

$$\frac{\partial}{\partial\epsilon}|\Sigma(F_{\mathbf{y},\epsilon})| = \operatorname{tr}\left(\Sigma^{-1}(F_{\mathbf{y},\epsilon})\ \frac{\partial}{\partial\epsilon}\Sigma(F_{\mathbf{y},\epsilon})\right)|\Sigma(F_{\mathbf{y},\epsilon})|\ .$$

Moreover

$$\frac{\partial}{\partial\epsilon}\Sigma(F_{\mathbf{y},\epsilon}) = \mathbf{w}\mathbf{w}'-\Sigma\ ,$$

where $\mathbf{w}$ and Σ are as defined above. Thus

$$\begin{aligned}\frac{\partial}{\partial\epsilon}\ln|\Sigma(F_{\mathbf{y},\epsilon})| &= \frac{1}{|\Sigma(F_{\mathbf{y},\epsilon})|}\ \mathrm{tr}\left(\Sigma^{-1}(F_{\mathbf{y},\epsilon})\ \frac{\partial}{\partial\epsilon}\Sigma(F_{\mathbf{y},\epsilon})\right)|\Sigma(F_{\mathbf{y},\epsilon})| \\ &= \mathrm{tr}\left(\left[(1+\epsilon)\Sigma^{-1} - \epsilon\Sigma^{-1}\mathbf{w}\mathbf{w}'\Sigma^{-1}\right](\mathbf{w}\mathbf{w}' - \Sigma)\right)\end{aligned}$$

that gives

$$\left[\frac{\partial}{\partial\epsilon}\ln|\Sigma(F_{\mathbf{y},\epsilon})|\right]_{\epsilon=0} = \mathrm{tr}\left(\Sigma^{-1}(\mathbf{w}\mathbf{w}' - \Sigma)\right) = \phi_F(\mathbf{y},\mathbf{y}) - d\ . \tag{5.4}$$

Similarly

$$\left[\frac{\partial}{\partial\epsilon}\ln|\Sigma(G_{\mathbf{y},\epsilon})|\right]_{\epsilon=0} = \phi_G(\mathbf{y},\mathbf{y}) - d\ . \tag{5.5}$$

Putting equations (5.2), (5.3), (5.4), and (5.5) together gives the desired result. □

It is clear that $IF(\mathbf{y}; Q(\mathbf{z}; F, G))$ is unbounded in $\mathbf{y}$. In fact it is a quadratic in $\mathbf{y}$. This shows that quadratic discriminant analysis is a non-robust procedure and may provide misleading results if the data are contaminated.

5.3 IF for RD

Recall the rank based discriminant function given in Equation 4.1

$$RD(\mathbf{z}; F, G) = P_F\left\{D(\mathbf{z}; F, G) \geq D(\mathbf{X}; F, G)\right\} - P_G\left\{D(\mathbf{z}; F, G) \leq D(\mathbf{Y}; F, G)\right\}\ ,$$

where $D(\mathbf{X}; F, G)$ is a discriminant function. We will restrict our attention to the cases where $D(\mathbf{z}; F, G)$ is defined as a difference between the Type D depth of $\mathbf{z}$ in F and $\mathbf{z}$ in G

$$D(\mathbf{z}; F, G) = DF(\mathbf{z}; F) - DF(\mathbf{z}; G) \ .$$

The Type D depth of the point $\mathbf{z}$ in F is defined as (Zuo and Serfling, 2000)

$$DF(\mathbf{z}; F) = \inf\{P_F(C) : \mathbf{z} \in C \in \Gamma\}$$

where Γ is a specified class of subsets of $\mathbb{R}^d$. For instance, if Γ is the class of halfspaces in $\mathbb{R}^d$, then DF is the halfspace depth. As in Wang and Serfling (2006), we assume that

1. If $C \in \Gamma$, then $\overline{C^c} \in \Gamma$, and
2. $\max_{\mathbf{z}} DF(\mathbf{z}; F) < 1$,

where A^c and $\overline{A}$ denote the complement and the closure of the set A.

Let

$$H_F(t; F, G) = P_F\left\{D(\mathbf{X}; F, G) \leq t\right\} \quad \text{and} \quad H_G(t; G, F) = P_G\left\{D(\mathbf{X}; G, F) \leq t\right\} \ .$$

Note that

$$RD(\mathbf{z}; F, G) = H_F(D(\mathbf{z}; F, G); F, G) - H_G(D(\mathbf{z}; G, F); G, F) \ .$$

Theorem 5.2. *If $D(\mathbf{v}; F, G)$ is continuous in $\mathbf{v}$, then the influence function of $H_F(t; F, G)$ for a fixed t is given by*

$$\mathrm{IF}(\mathbf{y}; H_F(t; F, G)) = \begin{cases} t * h_F(t) - H_F(t) & \text{if } D(\mathbf{y}; F, G) > t \\ 1 + t * h_F(t) - H_F(t) & \text{if } D(\mathbf{y}; F, G) \leq t \ , \end{cases}$$

where $h_F(t) = dH_F(t)/dt$.

Proof. It is easy to see that for Type D depth functions DF,

$$D(\mathbf{z}; F_{\mathbf{y},\epsilon}, G_{\mathbf{y},\epsilon}) = (1-\epsilon)DF(\mathbf{z}; F) + \epsilon - (1-\epsilon)DF(\mathbf{z}, G) - \epsilon = (1-\epsilon)D(\mathbf{z}; F, G) \ .$$

Then following an approach similar to Wang and Serfling (2006) we obtain

$$\begin{aligned} H_{F_{\mathbf{y},\epsilon}}(t; F_{\mathbf{y},\epsilon}, G_{\mathbf{y},\epsilon}) &= P_{F_{\mathbf{y},\epsilon}} \{(1-\epsilon)D(\mathbf{X}; F, G) \leq t\} \\ &= (1-\epsilon)P_F \{(1-\epsilon)D(\mathbf{X}; F, G) \leq t\} + \epsilon I \{(1-\epsilon)D(\mathbf{y}; F, G) \leq t\} \\ &= (1-\epsilon)P_F \{(1-\epsilon)D(\mathbf{X}; F, G) \leq t \ \cap \ \mathbf{X} \in \mathcal{S}_{\mathbf{y}}\} + \\ &\qquad (1-\epsilon)P_F \{(1-\epsilon)D(\mathbf{X}; F, G) \leq t \ \cap \ \mathbf{X} \notin \mathcal{S}_{\mathbf{y}}\} + \\ &\qquad\quad \epsilon I \{(1-\epsilon)D(\mathbf{y}; F, G) \leq t\} \ , \end{aligned}$$

where $\mathcal{S}_{\mathbf{y}} = \{\mathbf{x} : D(\mathbf{x}; F, G) \geq D(\mathbf{y}; F, G)\}$. Consider the following two cases:

(1) $D(\mathbf{y}; F, G) > t$ **:** For $\epsilon > 0$ sufficiently small

$$\{\mathbf{v} : D(\mathbf{v}; F, G) \leq t/(1-\epsilon)\} \cap \mathcal{S}_{\mathbf{y}} = \emptyset$$

$$\{\mathbf{v} : D(\mathbf{v}; F, G) \leq t/(1-\epsilon)\} \cap \mathcal{S}_{\mathbf{y}}^c = \{\mathbf{v} : D(\mathbf{v}; F, G) \leq t/(1-\epsilon)\}$$

$$\mathbf{y} \notin \{\mathbf{v} : D(\mathbf{v}; F, G) \leq t/(1-\epsilon)\}$$

Thus

$$H_{F_{\mathbf{y},\epsilon}}(t; F_{\mathbf{y},\epsilon}, G_{\mathbf{y},\epsilon}) = (1-\epsilon) H_F(t/(1-\epsilon))$$

which gives

$$\mathrm{IF}(\mathbf{y}; H_F(t; F, G)) = \frac{\partial}{\partial \epsilon} H_{F_{\mathbf{y},\epsilon}}(t; F_{\mathbf{y},\epsilon}, G_{\mathbf{y},\epsilon}) \bigg|_{\epsilon=0} = t * h_F(t) - H_F(t)\ .$$

(2) $D(\mathbf{y}; F, G) \leq t$ **:** Once again for $\epsilon > 0$ sufficiently small

$$\{\mathbf{v} : D(\mathbf{v}; F, G) \leq t/(1-\epsilon)\} \cap \mathcal{S}_{\mathbf{y}} = \{\mathbf{v} : D(\mathbf{y}; F, G) \leq D(\mathbf{v}; F, G) \leq t/(1-\epsilon)\}$$

$$\{\mathbf{v} : D(\mathbf{v}; F, G) \leq t/(1-\epsilon)\} \cap \mathcal{S}_{\mathbf{y}}^c = \{\mathbf{v} : D(\mathbf{v}; F, G) \leq D(\mathbf{y}; F, G)\}$$

$$\mathbf{y} \in \{\mathbf{v} : D(\mathbf{v}; F, G) \leq t/(1-\epsilon)\}$$

Thus

$$H_{F_{\mathbf{y},\epsilon}}(t; F_{\mathbf{y},\epsilon}, G_{\mathbf{y},\epsilon}) = (1-\epsilon) H_F(t/(1-\epsilon)) + \epsilon$$

which gives

$$\mathrm{IF}(\mathbf{y}; H_F(t;F,G)) = \frac{\partial}{\partial \epsilon} H_{F_{\mathbf{y},\epsilon}}(t; F_{\mathbf{y},\epsilon}, G_{\mathbf{y},\epsilon})\Big|_{\epsilon=0} = 1 + t * h_F(t) - H_F(t)\ .$$

□

The influence function of $H_G(t; G, F)$ may be obtained following similar steps.

We now give the influence function of the rank discriminant function.

Theorem 5.3. *If $D(\mathbf{z}; F, G)$ is continuous, then the influence function of $RD(\mathbf{z}; F, G)$ is*

$$\mathrm{IF}(\mathbf{y}; RD(\mathbf{z}, F, G)) = \begin{cases} H_G(D(\mathbf{z}; G, F); G, F) - H_F(D(\mathbf{z}; F, G); F, G) + 1\ , \\ \qquad\qquad \text{if } D(\mathbf{y}; F, G) < D(\mathbf{z}; F, G) \\ H_G(D(\mathbf{z}; G, F); G, F) - H_F(D(\mathbf{z}; F, G); F, G)\ , \\ \qquad\qquad \text{if } D(\mathbf{y}; F, G) = D(\mathbf{z}; F, G) \\ H_G(D(\mathbf{z}; G, F); G, F) - H_F(D(\mathbf{z}; F, G); F, G) - 1\ , \\ \qquad\qquad \text{if } D(\mathbf{y}; F, G) > D(\mathbf{z}; F, G)\ . \end{cases}$$

Proof. Following steps that are similar to the proof of Theorem 5.2 and some algebraic manipulation, we have

$$
\begin{aligned}
RD(\mathbf{z}; F_{\mathbf{y},\epsilon}, G_{\mathbf{y},\epsilon}) &= H_{F_{\mathbf{y},\epsilon}}(D(\mathbf{z}; F_{\mathbf{y},\epsilon}, G_{\mathbf{y},\epsilon}); F_{\mathbf{y},\epsilon}, G_{\mathbf{y},\epsilon}) - \\
&\qquad H_{G_{\mathbf{y},\epsilon}}(D(\mathbf{z}; G_{\mathbf{y},\epsilon}, F_{\mathbf{y},\epsilon}); G_{\mathbf{y},\epsilon}, F_{\mathbf{y},\epsilon}) \\
&= (1-\epsilon)\left[H_F(D(\mathbf{z}; F, G); F, G) - H_G(D(\mathbf{z}; G, F); G, F)\right] + \\
&\qquad \epsilon[I\left\{D(\mathbf{y}; F, G) \le D(\mathbf{z}; F, G)\right\} - \\
&\qquad\quad I\left\{D(\mathbf{y}; G, F) \le D(\mathbf{z}; G, F)\right\}] .
\end{aligned}
$$

Note that $I\left\{D(\mathbf{y}; G, F) \le D(\mathbf{z}; G, F)\right\} = I\left\{D(\mathbf{y}; F, G) \ge D(\mathbf{z}; F, G)\right\}$ and therefore $I\left\{D(\mathbf{y}; F, G) \le D(\mathbf{z}; F, G)\right\} - I\left\{D(\mathbf{y}; G, F) \le D(\mathbf{z}; G, F)\right\}$ equals 0, 1, or -1 depending on whether $D(\mathbf{y}; F, G) = D(\mathbf{z}; F, G)$, $D(\mathbf{y}; F, G) < D(\mathbf{z}; F, G)$, or $D(\mathbf{y}; F, G)| > D(\mathbf{z}; F, G)$, respectively. Differentiating with respect to ϵ and letting $\epsilon = 0$ gives the desired result. □

Since H_F and H_G are CDFs, this influence function remains bounded in $\mathbf{y}$ as long as the Type D depth function used is continuous. As $\mathrm{IF}(\mathbf{y}; RD(\mathbf{z}, F, G))$ is a step function, RD has finite gross error sensitivity but infinite local shift sensitivity. This behavior is similar to the behavior of the influence function of univariate quantiles.

5.4 Sensitivity Curves for the Projection Based Classifier

To view the effect of a single point contamination introduced into the training sample on the probability of misclassification error graphically, one needs a sample version of the influence function provided in Equation (5.1). The quantity:

$$SC_n(\mathbf{y}) = n\left[T_n(\mathbf{y}_1, \ldots, \mathbf{y}_{n-1}, \mathbf{y}) - T_{n-1}(\mathbf{y}_1, \ldots, \mathbf{y}_{n-1})\right] \;, \tag{5.6}$$

where $\mathbf{y}_1, \ldots, \mathbf{y}_{n-1}$ represents the training samples and $\mathbf{y}$ is a new point that is introduced into each of the two training samples (Tukey, 1971). T_n is the probability of misclassification error. SC_n is known as the *sensitivity curve* and it shows the effect of the added new point on the classifier as the value of this new point changes. The sensitivity curve of a robust procedure is bounded in much the same way the influence function is bounded.

We use a simple case where two groups with training sample of sizes 100 each and testing samples of sizes 1000 each are generated from four dimensional normal distributions with means $(0, 0, 0, 0)$ and $(2, 0, 0, 0)$. To simplify things further, we used the identity covariance structure for both the distributions. We introduced a single point $(i, 0, 0, 0), i = -200, \ldots, 200$ into both the training samples and for each point we calculate the probability of misclassification error with and without the added point using a leave-one-out cross validation for the uncontaminated testing samples. The sensitivity of the probability of misclassification for each classifier is then calculated using the Equation (5.6) and all these values are plotted in a graph.

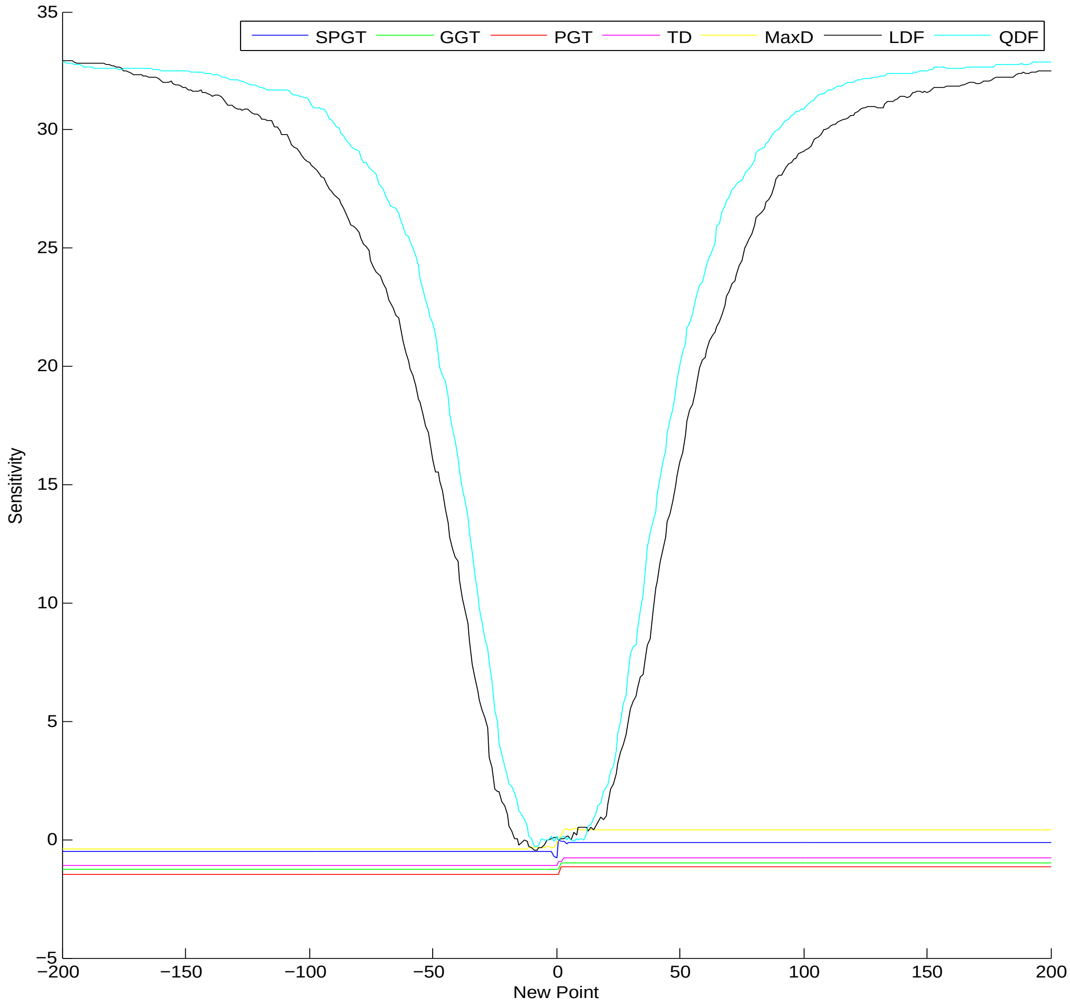

Figure 5.1: Sensitivity Curve for the Projection Based Methods

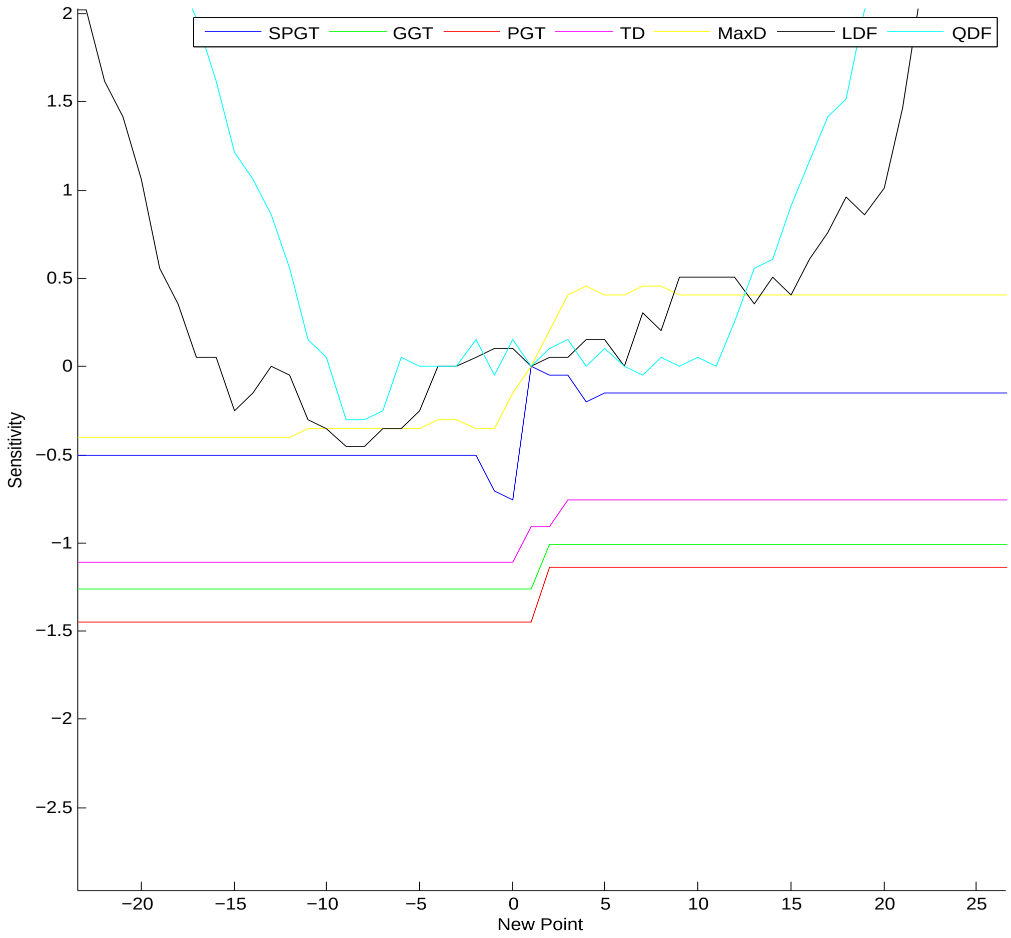

Figure 5.2: A Zoomed View of the Sensitivity Curve for the Projection Based Methods

The sensitivity curve in Figure 5.1 clearly shows that the classifiers LDF and QDF appear to have an unbounded misclassification error sensitivity. As the magnitude of the new point increases, so does the sensitivity of the misclassification error for these classifiers. On the other hand MaxD and all the projection based classifiers: SPGT, GGT, PGT and TD seem to be bounded. Although they are affected by the presence of the outlier, their sensitivity is eventually bounded. A zoomed-in view of the plot when the newly added point is around the origin is shown in Figure 5.2. It shows that the sensitivity curves for the projection based methods are bounded, with PGT being the least affected. A theoretical evaluation of the influence function of these projection based classifiers appears to be extremely complicated although the sensitivity curves give us a strong indication that the influence function will be bounded.

Chapter 6

Conclusion

Two types of nonparametric procedures for discriminant analysis were considered; one based on transvariation probabilities and the other based on ranking discriminant functions.

The first method is a nonparametric discriminant analysis procedure that uses the method of projection pursuit in tandem with the idea of transvariation probabilities. In particular, group separation is measured using the two-group transvariation probability (Gini, 1916) as a projection index. Allocation of the new observation is performed either using a symmetrized group-group transvariation probability or a smoothed version of point-group transvariation. These procedures are shown to give smaller misclassification error rates compared to linear and quadratic discriminant functions and maximum L_1 depth classifier when the training samples are drawn from symmetric and skewed distributions and especially when the difference between the training sample sizes gets larger. Moreover these methods give smaller misclassification error rates compared to Montanari's (Montanari, 2004) transvariation based classifier that uses distances to allocate the new observation when the training samples are drawn from skewed distributions. An extensive simulation study and applications on three real data sets (Fisher's IRIS, Leukemia and Colon datasets) are used to illustrate this behavior.

Although computation time can be a hinderance for projection pursuit techniques, especially for large dimensions, we were able to optimize our computation by generating projection directions using a multiscale procedure. To begin with, we proceed by first generating a few points on the hypercube, that are projected onto the hypersphere and then repeatedly "zooming-in" to interesting hyperarcs on the surface of the d-dimensional hypersphere where more directions are considered. We took five lattice points per dimension on the hypercube. No significant improvement in misclassification error rate was noticed when seven or nine lattice points were used instead of five. We have found that this makes for a faster computation time without compromising the results.

To illustrate the robustness of the proposed procedures, we provide sensitivity curves (Tukey, 1971) where a new point is introduced into each training sample. Then this new point is changed and its effect on the misclassification error rate is measured. The curves provided in Chapter 5 show that the newly proposed classifiers have bounded sensitivity to the inclusion of the new point with respect to the misclassification error rate.

The second method is another nonparametric method that is based on ranking discriminant functions. In discriminant analysis, when an investigator has no prior reason to prefer one population to another, balance among the groups in terms of the probabilities of correct classification is a desirable property of a discriminant function. We asked ourselves whether the method of ranking discriminant functions given in Randles et al. (1978a) would work in balancing the probabilities of correct classification for the recently proposed maximum L_1 depth classifier (Jörnsten, 2004) and MCD (Minimum Covariance Determinant) based quadratic discriminant function (Hubert and Van Driessen, 2004). Not surprisingly, our extensive simulation study showed that balance between the probabilities of correct classification is achieved, while maintaining the overall robustness of the procedure under a variety of distributional settings.

As it turns out, ranking discriminant functions does more than just provide balance between the the probabilities of correct classification while maintaining the total probability of misclassification error at the level of the original discriminant functions. Ranking also gives smaller and more consistent total probabilities of misclassification error rates in cases where the underlying distributions are heavy tailed. This is reminiscent of the use of ranking in univariate and certain multivariate problems to arrive at procedures that are robust against a variety of violation of assumptions (Hettmansperger and McKean, 1998). A somewhat surprising revelation was that this benefit persisted even when the underlying discriminant function itself was robust such as the one given by Hubert and Van Driessen (2004). In particular, in the simulation settings that we investigated, the ranked version of the MCD based classifier of Hubert and Van Driessen (2004) gives the best performance of all the methods studied in terms of total probability of correct classification, balance, and standard errors when the two underlying distributions are Cauchy.

In Chapter 5 we derived the influence function for the quadratic discriminant function as well as the rank based discriminant function that is given in Chapter 4. The influence function for the quadratic discriminant function is unbounded, as expected, but the influence function for the rank based discriminant function is a step function and hence robust against gross error contamination.

Future Work

The work presented in this dissertation is by no stretch of imagination complete. A plethora of things can be done taking the work presented here as a starting point.

For the projection based classifiers, more efficient projection pursuit techniques that would let one classify observations from dimensions much higher are desired. Projection indices that are robust and at the same time amenable to a more efficient method of searching for the most interesting view of the data will make projection pursuit based classification more appealing. Moreover, each transvariation in the GGT classifier can be smoothed as in the SPGT classifier. A better smoothing function that works better than the t-CDF proposed in SPGT classifier can be found. Finding the optimal smoother for the t-CDF or for any other function theoretically is also an interesting problem.

For the rank based classifiers, one can always find robust distance measures that are more robust to deviations that we can then rank. That way the ranked version would be that much better and balanced. As mentioned in Ng and Randles (1983), the ratio of the misclassification error rates for the two samples can be controlled.

This idea could be applied to classify data from the medical field where the ratio of false-positive to false-negative can be controlled at any desired level.

Both types of classifiers only consider the two group classification. The extension to more than two groups is not immediate and is interesting. Theoretically, the influence function for the discriminant function can be derived for the projection based classifiers. This will confirm the results that are seen in the sensitivity curves. Finally, the training data breakdown point could be another theoretical result that could be found for both types of classifiers.

Bibliography

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci USA*, 96(12):6745–6750.

Anderson, T. W. (1984). *An introduction to multivariate statistical analysis*. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. John Wiley & Sons Inc., New York, second edition.

Andrews, D. F., Bickel, P. J., Hampel, F. R., Huber, P. J., Rogers, W. H., and Tukey, J. W. (1972). *Robust estimates of location: Survey and advances*. Princeton University Press, Princeton, N.J.

Billor, N., Abebe, A., Turkmen, A., and Nudurupati, S. V. (2008). Classification based on depth transvariations. *J. Classification*, (in press).

Brzezinski, J. R. (1999). *Logistic Regression Modeling for Context-Based Classification*. IEEE Computer Society, Washington, DC, USA.

Campbell, N. A. (1978). The influence function as an aid in outlier detection in discriminant analysis. *Applied Statistics*, 27(3):251–258.

Chen, Z. Y. (1989). Robust linear discriminant procedures using projection pursuit methods. Unpublished PhD Dissertation, University of Michigan.

Chen, Z.-Y. and Muirhead, R. J. (1994). A comparison of robust linear discriminant procedures using projection pursuit methods. In *Multivariate analysis and its applications (Hong Kong, 1992)*, volume 24 of *IMS Lecture Notes Monogr. Ser.*, pages 163–176. Inst. Math. Statist., Hayward, CA.

Chork, C. Y. and Rousseeuw, P. J. (1992). Integrating a high-breakdown option into discriminant analysis in exploration geochemistry. *Journal of Geochemical Exploration*, 43(3):191–203.

Christmann, A., Fischer, P., and Joachims, T. (2002). Comparison between various regression depth methods and the support vector machine to approximate the minimum number of misclassifications. *Comput. Statist.*, 17(2):273–287.

Christmann, A. and Rousseeuw, P. J. (2001). Measuring overlap in binary regression. *Comput. Statist. Data Anal.*, 37(1):65–75.

Crimin, K., McKean, J. W., and Sheather, S. J. (2007). Discriminant procedures based on efficient robust discriminant coordinates. *J. Nonparametr. Stat.*, 19(4-5):199–213.

Croux, C. and Dehon, C. (2001). Robust linear discriminant analysis using *S*-estimators. *Canad. J. Statist.*, 29(3):473–493.

Croux, C., Filzmoser, P., and Joossens, K. (2008). Classification efficiencies for robust linear discriminant analysis. *Statist. Sinica*, 18(2):581–599.

Croux, C., Filzmoser, P., and Oliveira, M. R. (2007). Algorithms for projection pursuit: robust principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 87(2):218–225.

Croux, C. and Haesbroeck, G. (2000). Principal component analysis based on robust estimators of the covariance or correlation matrix: Influence functions and efficiencies. *Biometrika*, 87(3):603–618.

Croux, C. and Joossens, K. (2005). Influence of observations on the misclassification probability in quadratic discriminant analysis. *J. Multivariate Anal.*, 96(2):384–403.

Dillon, W. R. (1979). The performance of the linear discriminant function in nonoptimal situations and the estimation of classification error rates: A review of recent findings. *Journal of Marketing Research*, 16:370–381.

Donoho, D. (1982). Breakdown properties of multivariate location estimators. In *PhD Qualifying paper*. Department of Statistics, Harvard University.

Donoho, D. L. and Gasko, M. (1992). Breakdown properties of location estimates based on halfspace depth and projected outlyingness. *Ann. Statist.*, 20(4):1803–1827.

Dudoit, S., Fridlyand, J., and Speed, T. P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Amer. Statist. Assoc.*, 97(457):77–87.

Eddy, W. F. (1985). Ordering of multivariate data. In *In Computer Science and Statistics: The Interface(L. Billard, ed.)*, pages 25–30. North-Holland, Amsterdam.

Fang, K. T. and Wang, Y. (1994). *Number-theoritic methods in statistics*. Chapman and Hall, London.

Filzmoser, P., Serneels, S., Croux, C., and Van Espen, P. J. (2006). Robust multivariate methods: The projection pursuit approach. *From Data and Information Analysis to Knowledge Engineering*, pages 270–277.

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

Friedman, J. H. (1987). Exploratory projection pursuit. *Journal of the American Statistical Association*, 82(397):249–266.

Friedman, J. H. and Tukey, J. W. (1974). Projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C 23(9):881–890.

Ghosh, A. K. and Chaudhuri, P. (2005). On maximum depth and related classifiers. *Scand. J. Statist.*, 32(2):327–350.

Gini, C. (1916). Il concetto di transvariazione e le sue prime applicazioni. *Giornale degli economisti Rivista di statistica.*

Golberg, M. A. (1972). The derivative of a determinant. *Amer. Math. Monthly*, 79:1124–1126.

Goldstein, M. (1987). [a review of multivariate analysis]: Comment. *Statistical Science*, 2(4):418–420.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531– 537.

Gunn, S. R. (1998). Support vector machines for classification and regression. Techinical Report, University of Southampton.

Hardy, A. (1991). On tests concerning the existence of a classification. *Belgian Journal of Operations Research, Statistics and Computer Sciences*, (31):111–126.

Hawkins, D. M. and McLachlan, G. J. (1997). High-breakdown linear discriminant analysis. *Journal of the American Statistical Association*, 92(437):136–143.

He, X. and Fung, W. K. (2000). High breakdown estimation for multiple populations with applications to discriminant analysis. *J. Multivariate Anal.*, 72(2):151–162.

Hellman, M. E. (1970). Nearest neighbor classification rule with a reject option. *IEEE Transactions on Systems Science and Cybernetics*, SSC6(3):179–&.

Hettmansperger, T. P. and McKean, J. W. (1998). *Robust nonparametric statistical methods*, volume 5 of *Kendall's Library of Statistics*. Edward Arnold, London.

Hettmansperger, T. P. and Randles, R. H. (2002). A practical affine equivariant multivariate median. *Biometrika*, 89(4):851–860.

Hills, M. (1967). Discrimination and allocation with discrete data. *Applied Statistics*, 16(3):237–250.

Hoberg, R. (2003). Clusteranalyse. In *Klassifikation und Datentiefe*. Eul, Lohmar.

Hollander, M. and Wolfe, D. A. (1999). *Nonparametric statistical methods*. Wiley Series in Probability and Statistics: Texts and References Section. John Wiley & Sons Inc., New York, second edition. A Wiley-Interscience Publication.

Huang, Y., Kao, T.-L., and Wang, T.-H. (2007). Influence functions and local influence in linear discriminant analysis. *Comput. Statist. Data Anal.*, 51(8):3844–3861.

Huber, P. J. (1977). Robust covariances. In *Statistical decision theory and related topics, II (Proc. Sympos., Purdue Univ., Lafayette, Ind., 1976)*, pages 165–191. Academic Press, New York.

Huber, P. J. (1981). *Robust statistics*. John Wiley & Sons Inc., New York. Wiley Series in Probability and Mathematical Statistics.

Huber, P. J. (1985). Projection pursuit. *Ann. Statist.*, 13(2):435–525. With discussion.

Huber, P. J. (1989). Programs for Exploratory Projection Pursuit. *Artemis Systems, Carlisle, MA*.

Hubert, M. and Van Driessen, K. (2004). Fast and robust discriminant analysis. *Comput. Statist. Data Anal.*, 45(2):301–320.

Hugg, J., Rafalin, R., Seyboth, K., and Souvaine, D. (2006). An experimental study of old and new depth measures. In *Workshop on Algorithm Engineering and Experiments (ALENEX06)*, pages 51–64. Springer-Verlag Lecture Notes in Computer Science, New York.

Johnson, M. E., Wang, C., and Ramberg, J. S. (1979). Robustness of fisher's linear discriminant function to departures from normality. In *Los Alamos Techinical Report LA-8068-MS*. Los Alamos, NM 87545.

Johnson, N. L. (1949). Systems of frequency curves generated by methods of translation. *Biometrika*, 36:149–176.

Jones, M. C. (1983). The projection pursuit algorithm for exploratory data anlaysis. *PhD Thesis, University of Bath.*

Jones, M. C. and Sibson, R. (1987). What is projection pursuit? *J. Roy. Statist. Soc. Ser. A*, 150(1):1–36. With discussion.

Joossens, K. and Croux, C. (2004). Empirical comparison of the classification performance of robust linear and quadratic discriminant analysis. In *Theory and applications of recent robust methods*, Stat. Ind. Technol., pages 131–140. Birkhäuser, Basel.

Jörnsten, R. (2004). Clustering and classification based on the L_1 data depth. *J. Multivariate Anal.*, 90(1):67–89.

Lachenbruch, P. A. (1975). *Discriminant analysis.* Hafner Press, New York.

Lachenbruch, P. A., Sneeringer, C., and Revo, L. T. (1973). Robustness of the linear and quadratic discriminant function to certain types of non-normality. *Comm. Statist.*, 1(1):39–56.

LeBlanc, M. and Tibshirani, R. (1996). Combining estimates in regression and classification. *J. Amer. Statist. Assoc.*, 91(436):1641–1650.

Liu, R. Y. (1990). On a notion of data depth based on random simplices. *Ann. Statist.*, 18(1):405–414.

Liu, R. Y. (1992). Data depth and multivariate rank tests. In *L_1-statistical analysis and related methods (Neuchâtel, 1992)*, pages 279–294. North-Holland, Amsterdam.

Liu, R. Y., Parelius, J. M., and Singh, K. (1999). Multivariate analysis by data depth: descriptive statistics, graphics and inference. *Ann. Statist.*, 27(3):783–858. With discussion and a rejoinder by Liu and Singh.

Liu, R. Y. and Singh, K. (1993). A quality index based on data depth and multivariate rank tests. *J. Amer. Statist. Assoc.*, 88(421):252–260.

Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Science of India*, pages 49–55.

Marsaglia, G. (1972). Choosing a point from the surface of a sphere. *The Annals of Mathematical Statistics*, 43(2):645–646.

McLachlan, G. J. (1992). *Discriminant analysis and statistical pattern recognition.* Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons Inc., New York. A Wiley-Interscience Publication.

Mojirsheibani, M. (1997). A consistent combined classification rule. *Statist. Probab. Lett.*, 36(1):43–47.

Mojirsheibani, M. (1999). Combining classifiers via discretization. *J. Amer. Statist. Assoc.*, 94(446):600–609.

Mojirsheibani, M. (2000). A kernel-based combined classification rule. *Statist. Probab. Lett.*, 48(4):411–419.

Montanari, A. (2004). Linear discriminant analysis and transvariation. *J. Classification*, 21(1):71–88.

Mosler, K. (2002). *Multivariate dispersion, central regions and depth*, volume 165 of *Lecture Notes in Statistics.* Springer-Verlag, Berlin. The lift zonoid approach.

Mosler, K. and Hoberg, R. (2003). Classification based on data depth, Bulliten of the ISI 54th session.

Nason, G. (1995). Three-dimensional projection pursuit. *Applied Statistics*, 44(4):411–430.

Ng, T. H. and Randles, R. H. (1983). Rank procedures in many population forced discrimination problems. *Comm. Statist. A—Theory Methods*, 12(17):1943–1959.

Nguyen, D. V. and Rocke, D. M. (2002). Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, 18:39–50.

Pao, Y. H. (1989). *Adaptive pattern Recognition and Neural Network*. Addition-Wesley Publishing Company, Inc.

Posse, C. (1990). An effective two-deimensional projection pursuit algorithm. *Communications in Statistics: Simulation and Computation*, 19:1143–1164.

Posse, C. (1992). Projection pursuit discriminant analysis for two groups. *Comm. Statist. Theory Methods*, 21(1):1–19.

Posse, C. (1995). Projection pursuit exploratory data analysis. *Computational Statistics & Data Analysis*, 20(6):669–687.

Powell, M. J. D. (1964). Efficient method for finding minimum of function of several-variables without calculating derivatives. *Computer Journal*, 7(2):155–&.

Randles, R. H., Broffitt, J. D., Ramberg, J. S., and Hogg, R. V. (1978a). Discriminant analysis based on ranks. *J. Amer. Statist. Assoc.*, 73(362):379–384.

Randles, R. H., Broffitt, J. D., Ramberg, J. S., and Hogg, R. V. (1978b). Generalized linear and quadratic discriminant functions using robust estimates. *J. Amer. Statist. Assoc.*, 73(363):564–568.

Rätsch, G. (1998). Ensemble learning methods for classification. Diploma thesis (in german).

Rätsch, G., Onoda, T., and Müller, K.-R. (1998). Soft margins for AdaBoost. Technical Report NC-TR-1998-021, Department of Computer Science, Royal Holloway, University of London, Egham, UK. Submitted to Machine Learning.

Rosenbrock, H. H. (1960). An automatic method for finding the gretest or least value of a function. *Computer Journal*, 3(3):175–184.

Rousseeuw, P. (1985). Multivariate estimation with high breakdown point. In *Mathematical statistics and applications, Vol. B (Bad Tatzmannsdorf, 1983)*, pages 283–297. Reidel, Dordrecht.

Rousseeuw, P. and Van Driessen, K. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223.

Rousseeuw, P. J. (1984). Least median of squares regression. *J. Amer. Statist. Assoc.*, 79(388):871–880.

Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust Regression and Outlier Detection.* Wiley, New York.

Rousseeuw, P. J. and Vanzomeren, B. C. (1990). Unmasking multivariate outliers and levereage points. *Journal of the American Statistical Association*, 85(411):633–639.

Rousseeuw, P. J. and Yohai, V. (1984). Robust regression by means of S-estimators. *Robust and Nonlinear Time Series Analysis. Lecture Notes in Statistics*, 26:256–272.

Seber, G. A. F. (1984). *Multivariate Observations.* John Wiley, New York.

Serfling, R. J. (1980). *Approximation theorems of mathemtical statistics.* John Wiley & Sons Inc., New York. Wiley Series in Probability and Mathematical Statistics.

Singh, K. (1991). A notion of majority depth. In *Technical Report.* Department of Statistics, Rutgers University.

Ting, K. (2002). *Cost-sensitive classification using decision trees.* Idea Group Publishing.

Tukey, J. (1974). Address to international congress of mathematicians. Vancouver.

Tukey, J. W. (1971). *Explaratory Data Analysis.* Addison-Wesley, Reading.

Tyler, D. E. (1987). A distribution-free M-estimator of multivariate scatter. *Ann. Statist.*, 15(1):234–251.

Van Ness, J. W. and Yang, J. J. (1998). Robust discriminant analysis: training data breakdown point. *J. Statist. Plann. Inference*, 67(1):67–83.

Vardi, Y. and Zhang, C.-H. (2000). The multivariate L_1-median and associated data depth. *Proc. Natl. Acad. Sci. USA*, 97(4):1423–1426 (electronic).

Wang, J. and Serfling, R. (2006). Influence functions for a general class of depth-based generalized quantile functions. *J. Multivariate Anal.*, 97(4):810–826.

Yenyukov, I. S. (1988). Detecting Structures by Means of Projection Pursuit. *Proceeding of COMPSTAT*, 88:48–58.

Yenyukov, I. S. (1989). Indices for projection pursuit. *E. Diday (Ed.), Data Analysis, Learning Symbolic and Numeric Language*, pages 181–189.

Zuo, Y. and Serfling, R. (2000). General notions of statistical depth function. *Ann. Statist.*, 28(2):461–482.

Appendices

Table 1: Application on Real Data: Performance of Projection Based Methods Using Leave-one-out Misclassification Error Rates

Dist	SPGT	PGT	GGT	TD	MaxD	LDF	QDF
Iris							
Original	0.0933	0.0467	0.0400	0.3133	0.0333	0.1067	0.0333
Outliers	0.0774	0.0397	0.0387	0.2968	0.0774	0.2129	0.1226
Leukemia	0.0278	0.0290	0.0139	0.1111	0.0417	0.0417	0.0417
Colon	0.1290	0.1525	0.1290	0.1290	0.1774	0.1290	0.1613

Table 2: Performance of Projection Based Methods Using Monte Carlo Simulation: Average Misclassification Error Rates and Standard Errors on 50 Replications of 1000 Test Cases per Group

Dist	SPGT	PGT	GGT	TD	MaxD	LDF	QDF
N(I),N(I)	0.1641	0.1625	0.1624	0.1624	0.1665	0.1635	0.1659
(150,150)	0.0091	0.0086	0.0087	0.0083	0.0079	0.0075	0.007
N(I),N(I)	0.1638	0.1953	0.1696	0.1607	0.1711	0.1623	0.1685
(50,250)	0.0097	0.015	0.0122	0.0094	0.0114	0.0085	0.0098
t(I),t(I)	0.2165	0.216	0.2137	0.213	0.2932	0.2465	0.3698
(150,150)	0.0097	0.0109	0.011	0.0107	0.0372	0.0405	0.0738
t(I),t(I)	0.2258	0.4075	0.2227	0.2177	0.3093	0.26	0.3833
(50,250)	0.018	0.0711	0.0177	0.0151	0.0387	0.0501	0.0673
C(I),C(I)	0.2617	0.2643	0.2565	0.2557	0.4208	0.4492	0.5
(150,150)	0.0116	0.0152	0.0114	0.0116	0.0468	0.0777	0.0056
C(I),C(I)	0.2747	0.4938	0.2659	0.2615	0.4201	0.4501	0.4984
(50,250)	0.024	0.0239	0.0218	0.0208	0.0482	0.072	0.0095
LN(I),LN(I)	0.1595	0.1588	0.1581	0.1783	0.2361	0.2576	0.2273
(150,150)	0.0104	0.0099	0.0088	0.0119	0.0338	0.0243	0.0249
LN(I),LN(I)	0.1644	0.1971	0.1713	0.1856	0.2713	0.2761	0.2426
(50,250)	0.0106	0.0173	0.0139	0.0112	0.0405	0.0362	0.0338

Table 2 Continued

Dist	SPGT	PGT	GGT	TD	MaxD	LDF	QDF
N(I),N(W)	0.0954	0.1033	0.105	0.1003	0.0663	0.0969	0.0656
(150,150)	0.0095	0.0124	0.012	0.0088	0.0062	0.007	0.0059
N(I),N(W)	0.0993	0.1001	0.0974	0.1039	0.0698	0.0999	0.0686
(50,250)	0.012	0.0171	0.0124	0.0111	0.0052	0.0086	0.0051
t(I),t(W)	0.2095	0.2093	0.2043	0.2045	0.247	0.2222	0.3152
(150,150)	0.012	0.015	0.0131	0.0119	0.054	0.0425	0.0693
t(I),t(W)	0.2151	0.337	0.2093	0.2077	0.267	0.2249	0.3313
(50,250)	0.0145	0.0818	0.0119	0.0129	0.0553	0.0354	0.0554
C(I),C(W)	0.2601	0.2567	0.2531	0.2505	0.4007	0.4264	0.4888
(150,150)	0.0145	0.0173	0.0141	0.0131	0.0542	0.0706	0.0176
C(I),C(W)	0.2645	0.4967	0.2546	0.2525	0.3728	0.4326	0.4675
(50,250)	0.0191	0.0186	0.0132	0.0126	0.0478	0.1004	0.0207
LN(I),LN(W)	0.1271	0.1253	0.1258	0.1493	0.1729	0.2273	0.2382
(150,150)	0.01	0.01	0.0093	0.0141	0.0228	0.0216	0.0293
LN(I),LN(W)	0.132	0.144	0.1313	0.1537	0.1889	0.2376	0.2339
(50,250)	0.0136	0.0144	0.0137	0.0139	0.036	0.0284	0.039

Table 2 Continued

Dist	SPGT	PGT	GGT	TD	MaxD	LDF	QDF
N(I),t(I)	0.1817	0.1932	0.19	0.1851	0.3331	0.2054	0.207
(150,150)	0.0115	0.0163	0.0108	0.0117	0.0623	0.0443	0.0251
N(I),t(I)	0.1868	0.4262	0.2093	0.1893	0.3418	0.2022	0.2123
(50,250)	0.0104	0.0688	0.0132	0.0097	0.0458	0.0184	0.0267
N(I),C(I)	0.1844	0.2312	0.2213	0.2074	0.4516	0.3288	0.2356
(150,150)	0.0099	0.0253	0.0133	0.0082	0.0505	0.0841	0.0263
N(I),C(I)	0.1884	0.5016	0.2366	0.2057	0.4691	0.3314	0.2357
(50,250)	0.0171	0.0044	0.0172	0.0091	0.0341	0.0819	0.0238
LN(I),N(I)	0.131	0.1293	0.1286	0.131	0.1506	0.1506	0.2098
(150,150)	0.0091	0.0085	0.0087	0.0083	0.0169	0.0143	0.0256
LN(I),N(I)	0.1487	0.1529	0.1715	0.1375	0.1523	0.1522	0.2004
(50,250)	0.0163	0.0141	0.0153	0.0156	0.0248	0.0202	0.0271
t(I),C(I)	0.2333	0.2434	0.2392	0.2354	0.4077	0.3543	0.3918
(150,150)	0.0108	0.0212	0.0088	0.0084	0.0534	0.0796	0.0154
t(I),C(I)	0.2341	0.5024	0.2429	0.233	0.399	0.3405	0.3841
(50,250)	0.013	0.002	0.0159	0.0106	0.0532	0.0882	0.0177
LN(I),t(I)	0.1979	0.2032	0.204	0.1894	0.2268	0.2077	0.2452
(150,150)	0.0156	0.0154	0.0161	0.0125	0.0279	0.017	0.0709
LN(I),t(I)	0.225	0.2483	0.3979	0.1993	0.2546	0.2158	0.263
(50,250)	0.0234	0.023	0.0717	0.0204	0.04	0.0246	0.0653
LN(I),C(I)	0.2532	0.2789	0.2742	0.2433	0.3916	0.2805	0.3628
(150,150)	0.0233	0.0188	0.0253	0.0225	0.0426	0.0446	0.016
LN(I),C(I)	0.3024	0.3286	0.5018	0.2698	0.4091	0.3232	0.3608
(50,250)	0.0327	0.0328	0.0013	0.0374	0.0411	0.0765	0.0203

Table 2 Continued

Dist	SPGT	PGT	GGT	TD	MaxD	LDF	QDF
N(W),t(I)	0.1552	0.1793	0.1782	0.1655	0.1908	0.1746	0.0848
(150,150)	0.0248	0.0156	0.023	0.0182	0.0417	0.0305	0.0104
N(W),t(I)	0.1566	0.1672	0.139	0.157	0.1741	0.1502	0.0807
(50,250)	0.026	0.0164	0.0303	0.0195	0.0498	0.0242	0.0072
N(W),C(I)	0.1804	0.2203	0.2194	0.205	0.4146	0.3474	0.1149
(150,150)	0.0185	0.0163	0.0302	0.0139	0.0566	0.0985	0.0228
N(W),C(I)	0.1861	0.2189	0.1689	0.2046	0.3866	0.3238	0.0991
(50,250)	0.0291	0.0181	0.0288	0.0168	0.0741	0.0991	0.0232
LN(W),N(I)	0.0145	0.0141	0.0116	0.0681	0.0733	0.1327	0.047
(150,150)	0.004	0.0042	0.0043	0.0142	0.0247	0.027	0.0106
LN(W),N(I)	0.0155	0.0188	0.0177	0.0725	0.0886	0.1417	0.0483
(50,250)	0.0047	0.0076	0.0084	0.0128	0.0353	0.026	0.0149
t(W),C(I)	0.2296	0.2365	0.2406	0.231	0.3288	0.3734	0.4131
(150,150)	0.0145	0.012	0.022	0.0105	0.0512	0.0875	0.0183
t(W),C(I)	0.2425	0.238	0.3266	0.2331	0.3274	0.3587	0.4031
(50,250)	0.0154	0.0135	0.0734	0.0104	0.0535	0.0996	0.0349
LN(W),t(I)	0.0584	0.0555	0.0551	0.084	0.1107	0.1532	0.2045
(150,150)	0.0075	0.0063	0.0065	0.0134	0.0239	0.03	0.0487
LN(W),t(I)	0.0637	0.0594	0.0599	0.0915	0.1125	0.159	0.1933
(50,250)	0.0106	0.0119	0.0135	0.0159	0.0246	0.0292	0.0428
LN(W),C(I)	0.1193	0.1156	0.1145	0.1373	0.1648	0.2324	0.4103
(150,150)	0.0135	0.0125	0.0127	0.0204	0.0427	0.0573	0.0735
LN(W),C(I)	0.1277	0.1493	0.2292	0.1445	0.1814	0.2564	0.4515
(50,250)	0.0187	0.0201	0.0495	0.0288	0.0561	0.0741	0.0369

Table 3: Application on Real Data: Performance of Rank Based Methods Using Leave-one-out Misclassification Error Rates

Data	RS1	RS2	RS	S1	S2	S	RL1	RL2	RL	L1	L2	L
Iris												
Original	.060	.040	.053	.000	.500	.167	.270	.260	.267	.280	.260	.273
Outlier	.130	.098	.119	.010	.549	.192	.510	.451	.490	.490	.294	.424
Leukemia	.064	.120	.083	.064	.160	.097	.043	.040	.042	.021	.080	.042
Colon	.136	.200	.177	.409	.100	.210	.091	.125	.113	.091	.125	.113

Table 3 Continued

Data	RQ1	RQ2	RQ	Q1	Q2	Q	RM1	RM2	RM	M1	M2	M
Iris												
Original	.060	.040	.053	.010	.100	.040	.060	.060	.060	.060	.100	.073
Outlier	.110	.137	.119	.010	.843	.291	.080	.078	.080	.060	.118	.080
Leukemia	.064	.120	.083	.064	.160	.097	.064	.120	.083	.106	.160	.125
Colon	.136	.175	.161	.273	.100	.161	.091	.100	.097	.136	.075	.097

Table 4: Performance of Rank Based Methods Using Monte Carlo Simulation: Average Misclassification Error Rates and Standard Errors for Bivariate Data on 50 Replications of 1000 Test Cases per Group

Dist	RS1	RS2	RS	S1	S2	S	RL1	RL2	RL	L1	L2	L
N(I_2),N(I_2)	.168	.166	.167	.160	.175	.167	.163	.165	.164	.159	.166	.162
(50,50)	.036	.037	.010	.037	.045	.012	.035	.038	.009	.024	.027	.009
N(I_2),N(I_2)	.157	.177	.167	.184	.154	.169	.153	.173	.163	.159	.164	.162
(25,75)	.038	.039	.011	.047	.038	.015	.036	.035	.009	.028	.028	.009
t_2(I_2),t_2(I_2)	.268	.267	.268	.276	.261	.268	.252	.242	.247	.241	.254	.248
(50,50)	.060	.063	.035	.100	.099	.037	.083	.073	.061	.082	.094	.059
t_2(I_2),t_2(I_2)	.250	.279	.265	.292	.246	.269	.233	.259	.246	.269	.226	.248
(25,75)	.065	.053	.030	.114	.109	.039	.083	.076	.054	.114	.082	.063
C(I_2),C(I_2)	.350	.337	.343	.363	.330	.347	.385	.397	.391	.417	.412	.415
(50,50)	.067	.072	.046	.127	.114	.050	.149	.143	.136	.242	.224	.102
C(I_2),C(I_2)	.345	.352	.349	.427	.282	.354	.371	.397	.384	.433	.363	.398
(25,75)	.070	.082	.056	.154	.151	.056	.140	.138	.126	.240	.228	.102
N(I_2),N(V)	.126	.134	.130	.190	.077	.133	.146	.150	.148	.162	.129	.145
(50,50)	.035	.035	.008	.042	.026	.011	.036	.041	.009	.028	.027	.008
N(I_2),N(V)	.133	.141	.137	.216	.072	.144	.149	.156	.152	.168	.128	.148
(25,75)	.044	.047	.011	.052	.024	.018	.044	.053	.012	.029	.028	.009
t_2(I_2),t_2(V)	.237	.237	.237	.342	.145	.244	.236	.239	.238	.236	.240	.238
(50,50)	.041	.055	.029	.100	.063	.032	.052	.062	.044	.062	.081	.044
t_2(I_2),t_2(V)	.237	.261	.249	.383	.127	.255	.234	.228	.231	.246	.213	.230
(25,75)	.064	.054	.026	.116	.055	.042	.075	.054	.046	.078	.069	.041
C(I_2),C(V)	.318	.320	.319	.409	.237	.323	.342	.375	.358	.357	.409	.383
(50,50)	.058	.053	.040	.163	.135	.050	.114	.121	.110	.237	.227	.089
C(I_2),C(V)	.320	.378	.349	.490	.227	.358	.380	.418	.399	.449	.392	.421
(25,75)	.077	.078	.049	.204	.142	.068	.155	.148	.144	.295	.263	.089

Table 4 Continued

Dist	RQ1	RQ2	RQ	Q1	Q2	Q	RM1	RM2	RM	M1	M2	M
N(I_2),N(I_2)	.166	.165	.165	.162	.166	.164	.172	.165	.169	.165	.169	.167
(50,50)	.038	.037	.009	.026	.029	.010	.043	.036	.012	.034	.037	.012
N(I_2),N(I_2)	.171	.159	.165	.166	.164	.165	.176	.168	.172	.180	.158	.169
(25,75)	.046	.042	.011	.029	.031	.012	.051	.047	.013	.038	.031	.012
t_2(I_2),t_2(I_2)	.276	.287	.282	.289	.347	.318	.257	.244	.250	.252	.253	.253
(50,50)	.065	.070	.049	.165	.210	.080	.048	.045	.026	.050	.053	.023
t_2(I_2),t_2(I_2)	.268	.280	.274	.324	.315	.320	.259	.247	.253	.265	.238	.251
(25,75)	.070	.074	.049	.207	.188	.085	.064	.058	.028	.044	.048	.027
C(I_2),C(I_2)	.405	.395	.400	.526	.436	.481	.344	.335	.340	.348	.340	.344
(50,50)	.099	.084	.068	.376	.379	.044	.058	.057	.037	.108	.077	.056
C(I_2),C(I_2)	.388	.435	.411	.312	.648	.480	.356	.332	.344	.391	.312	.351
(25,75)	.094	.090	.078	.303	.316	.033	.076	.071	.039	.125	.102	.055
N(I_2),N(V)	.128	.135	.131	.133	.120	.126	.130	.137	.133	.138	.124	.131
(50,50)	.035	.039	.010	.027	.027	.006	.040	.048	.011	.028	.027	.009
N(I_2),N(V)	.122	.141	.131	.148	.118	.133	.152	.120	.136	.154	.116	.135
(25,75)	.030	.036	.010	.032	.028	.011	.040	.038	.015	.038	.036	.012
t_2(I_2),t_2(V)	.255	.252	.253	.242	.368	.305	.226	.217	.221	.214	.246	.230
(50,50)	.059	.061	.041	.204	.197	.085	.047	.050	.021	.032	.074	.032
t_2(I_2),t_2(V)	.250	.253	.251	.222	.371	.296	.231	.218	.225	.238	.213	.225
(25,75)	.054	.072	.039	.150	.165	.062	.047	.053	.020	.040	.056	.025
C(I_2),C(V)	.358	.384	.371	.391	.541	.466	.303	.314	.308	.277	.379	.328
(50,50)	.082	.087	.066	.374	.347	.051	.051	.060	.029	.038	.106	.047
C(I_2),C(V)	.362	.405	.384	.300	.649	.475	.301	.316	.309	.331	.333	.332
(25,75)	.100	.097	.079	.354	.326	.026	.053	.065	.034	.088	.114	.040

Table 4 Continued

Dist	RS1	RS2	RS	S1	S2	S	RL1	RL2	RL	L1	L2	L
N(I_2),C(I_2)	.197	.221	.209	.651	.045	.348	.335	.352	.344	.144	.499	.321
(50,50)	.056	.044	.025	.170	.029	.072	.136	.141	.133	.134	.221	.100
N(I_2),C(I_2)	.179	.243	.211	.633	.046	.339	.336	.392	.364	.131	.498	.314
(25,75)	.056	.050	.025	.149	.022	.065	.149	.180	.157	.102	.231	.109
N(I_2),t_2(I_2)	.194	.215	.205	.356	.112	.234	.204	.215	.210	.170	.239	.204
(50,50)	.052	.042	.019	.116	.038	.043	.062	.044	.032	.054	.049	.031
N(I_2),t_2(I_2)	.206	.239	.223	.410	.105	.257	.206	.235	.221	.177	.247	.212
(25,75)	.051	.052	.029	.134	.035	.054	.061	.069	.045	.049	.059	.044
C(I_2),t_2(I_2)	.328	.276	.302	.131	.508	.320	.380	.354	.367	.472	.242	.357
(50,50)	.065	.055	.036	.062	.145	.054	.143	.151	.142	.196	.150	.110
C(I_2),t_2(I_2)	.344	.276	.310	.199	.468	.334	.357	.318	.338	.498	.184	.341
(25,75)	.080	.063	.042	.132	.193	.068	.144	.120	.115	.217	.096	.091
N(I_2),C(V)	.140	.183	.161	.620	.028	.324	.301	.327	.314	.111	.477	.294
(50,50)	.043	.039	.017	.193	.018	.088	.148	.128	.129	.106	.214	.092
N(I_2),C(V)	.143	.223	.183	.648	.027	.338	.308	.382	.345	.113	.486	.300
(25,75)	.065	.049	.029	.190	.021	.086	.151	.140	.135	.097	.215	.086
N(I_2),t_2(V)	.144	.176	.160	.359	.065	.212	.178	.209	.193	.151	.228	.189
(50,50)	.042	.033	.014	.125	.025	.052	.050	.032	.024	.046	.048	.021
N(I_2),t_2(V)	.153	.191	.172	.420	.052	.236	.185	.215	.200	.167	.221	.194
(25,75)	.052	.047	.024	.121	.024	.051	.047	.059	.035	.045	.049	.029
C(I_2),t_2(V)	.288	.259	.273	.213	.344	.278	.358	.349	.354	.445	.277	.361
(50,50)	.047	.053	.033	.086	.128	.040	.122	.135	.121	.198	.154	.120
C(I_2),t_2(V)	.323	.269	.296	.258	.346	.302	.340	.325	.332	.480	.216	.348
(25,75)	.067	.061	.049	.127	.158	.051	.131	.123	.117	.235	.127	.107

Table 4 Continued

Dist	RQ1	RQ2	RQ	Q1	Q2	Q	RM1	RM2	RM	M1	M2	M
N(I_2),C(I_2)	.197	.218	.208	.027	.457	.242	.177	.179	.178	.109	.236	.172
(50,50)	.044	.034	.020	.031	.113	.043	.054	.033	.020	.031	.055	.020
N(I_2),C(I_2)	.195	.229	.212	.025	.468	.246	.201	.165	.183	.139	.232	.185
(25,75)	.050	.043	.023	.024	.100	.040	.071	.038	.027	.056	.058	.030
N(I_2),t_2(I_2)	.200	.201	.200	.113	.285	.199	.190	.182	.186	.146	.221	.183
(50,50)	.053	.029	.020	.037	.072	.025	.049	.030	.017	.030	.037	.014
N(I_2),t_2(I_2)	.204	.225	.215	.125	.310	.217	.207	.189	.198	.190	.212	.201
(25,75)	.066	.041	.024	.053	.090	.030	.068	.042	.025	.065	.043	.029
C(I_2),t_2(I_2)	.333	.294	.313	.723	.093	.408	.272	.268	.270	.323	.230	.277
(50,50)	.069	.057	.043	.135	.134	.040	.059	.051	.029	.094	.041	.039
C(I_2),t_2(I_2)	.318	.316	.317	.681	.110	.395	.279	.263	.271	.370	.205	.287
(25,75)	.060	.080	.046	.169	.100	.058	.062	.044	.029	.112	.047	.043
N(I_2),C(V)	.161	.163	.162	.010	.414	.212	.154	.129	.142	.067	.215	.141
(50,50)	.057	.030	.022	.011	.077	.034	.056	.030	.023	.026	.044	.018
N(I_2),C(V)	.144	.194	.169	.013	.418	.216	.151	.138	.144	.097	.202	.149
(25,75)	.040	.042	.021	.017	.084	.036	.053	.040	.022	.053	.055	.019
N(I_2),t_2(V)	.151	.161	.156	.061	.275	.168	.155	.137	.146	.111	.177	.144
(50,50)	.034	.034	.015	.025	.072	.027	.045	.031	.016	.030	.033	.012
N(I_2),t_2(V)	.140	.185	.163	.080	.256	.168	.166	.149	.157	.113	.177	.145
(25,75)	.045	.041	.016	.040	.091	.030	.052	.046	.021	.043	.043	.013
C(I_2),t_2(V)	.354	.308	.331	.654	.143	.398	.269	.240	.254	.301	.216	.259
(50,50)	.080	.086	.069	.197	.185	.056	.040	.046	.019	.053	.053	.029
C(I_2),t_2(V)	.333	.291	.312	.634	.165	.399	.290	.253	.272	.326	.206	.266
(25,75)	.066	.061	.049	.236	.189	.070	.058	.062	.029	.080	.056	.038

Table 5: Performance of Rank Based Methods Using Monte Carlo Simulation: Average Misclassification Error Rates and Standard Errors for 4D Data on 50 Replications of 1000 Test Cases per Group

Dist	RS1	RS2	RS	S1	S2	S	RL1	RL2	RL	L1	L2	L
N(I_4),N(I_4)	.186	.171	.179	.190	.169	.180	.177	.159	.168	.169	.163	.166
(50,50)	.039	.039	.014	.048	.054	.017	.038	.034	.012	.031	.027	.011
N(I_4),N(I_4)	.188	.185	.187	.243	.143	.193	.168	.169	.169	.177	.162	.170
(25,75)	.039	.040	.014	.073	.042	.022	.032	.033	.010	.029	.027	.011
$t_2(I_4)$,$t_2(I_4)$	.317	.326	.321	.349	.302	.325	.270	.273	.271	.262	.280	.271
(50,50)	.065	.054	.039	.151	.132	.046	.073	.081	.060	.065	.095	.060
$t_2(I_4)$,$t_2(I_4)$	.322	.329	.326	.408	.261	.335	.277	.286	.282	.314	.251	.283
(25,75)	.067	.057	.039	.148	.126	.039	.075	.078	.060	.090	.080	.061
C(I_4),C(I_4)	.420	.423	.422	.440	.396	.418	.418	.410	.414	.415	.425	.420
(50,50)	.069	.077	.050	.150	.150	.045	.081	.112	.083	.158	.169	.080
C(I_4),C(I_4)	.414	.441	.427	.509	.349	.429	.425	.457	.441	.560	.343	.452
(25,75)	.073	.070	.048	.164	.163	.043	.109	.115	.092	.233	.213	.069
N(I_4),N(W)	.079	.077	.078	.100	.047	.073	.103	.118	.111	.177	.030	.104
(50,50)	.020	.033	.009	.020	.023	.008	.021	.046	.016	.028	.025	.012
N(I_4),N(W)	.081	.080	.081	.115	.033	.074	.096	.119	.108	.184	.016	.100
(25,75)	.025	.037	.010	.026	.016	.008	.019	.048	.018	.030	.011	.011
$t_2(I_4)$,t_2(W)	.245	.246	.246	.402	.111	.257	.245	.252	.248	.263	.235	.249
(50,50)	.057	.062	.037	.134	.073	.052	.072	.076	.060	.061	.100	.055
$t_2(I_4)$,t_2(W)	.246	.271	.258	.463	.086	.274	.244	.268	.256	.296	.203	.250
(25,75)	.043	.060	.032	.139	.055	.054	.085	.091	.075	.111	.067	.070
C(I_4),C(W)	.363	.408	.386	.547	.246	.396	.409	.435	.422	.382	.477	.430
(50,50)	.065	.067	.042	.186	.137	.055	.118	.094	.088	.204	.172	.078
C(I_4),C(W)	.352	.413	.383	.573	.196	.385	.400	.443	.421	.492	.372	.432
(25,75)	.074	.077	.049	.191	.138	.058	.129	.129	.117	.247	.183	.104

Table 5 Continued

Dist	RQ1	RQ2	RQ	Q1	Q2	Q	RM1	RM2	RM	M1	M2	M
N(I_4),N(I_4)	.177	.176	.176	.180	.171	.175	.188	.192	.190	.177	.197	.187
(50,50)	.038	.036	.015	.034	.033	.013	.043	.047	.017	.038	.040	.017
N(I_4),N(I_4)	.183	.186	.185	.215	.161	.188	.215	.212	.213	.223	.172	.198
(25,75)	.045	.048	.018	.045	.029	.017	.064	.054	.029	.070	.038	.026
t_2(I_4),t_2(I_4)	.336	.324	.330	.359	.416	.387	.293	.304	.298	.308	.303	.306
(50,50)	.071	.074	.051	.236	.218	.063	.050	.052	.028	.058	.069	.036
t_2(I_4),t_2(I_4)	.322	.335	.329	.366	.393	.380	.317	.315	.316	.400	.268	.334
(25,75)	.081	.067	.051	.203	.206	.063	.065	.063	.036	.093	.063	.042
C(I_4),C(I_4)	.443	.444	.443	.499	.481	.490	.408	.403	.405	.420	.440	.430
(50,50)	.082	.082	.056	.345	.347	.024	.067	.065	.043	.118	.126	.044
C(I_4),C(I_4)	.452	.459	.456	.323	.654	.489	.415	.416	.415	.535	.340	.438
(25,75)	.069	.081	.051	.261	.275	.023	.088	.079	.034	.123	.104	.046
N(I_4),N(W)	.078	.077	.077	.094	.051	.072	.082	.084	.083	.097	.056	.076
(50,50)	.023	.037	.012	.016	.018	.007	.023	.042	.014	.020	.019	.009
N(I_4),N(W)	.074	.092	.083	.102	.044	.073	.096	.087	.091	.107	.052	.080
(25,75)	.021	.038	.012	.020	.016	.007	.031	.040	.022	.024	.020	.011
t_2(I_4),t_2(W)	.234	.285	.259	.216	.379	.297	.214	.213	.214	.203	.236	.220
(50,50)	.050	.058	.037	.144	.152	.059	.038	.064	.024	.038	.057	.026
t_2(I_4),t_2(W)	.258	.285	.272	.213	.386	.299	.226	.226	.226	.249	.190	.220
(25,75)	.056	.068	.040	.075	.133	.041	.054	.069	.031	.043	.055	.028
C(I_4),C(W)	.389	.421	.405	.360	.581	.471	.326	.364	.345	.301	.407	.354
(50,50)	.071	.085	.055	.347	.322	.033	.045	.057	.033	.072	.101	.036
C(I_4),C(W)	.388	.440	.414	.213	.702	.458	.354	.346	.350	.393	.351	.372
(25,75)	.064	.063	.031	.190	.183	.024	.071	.068	.029	.095	.095	.042

Table 5 Continued

Dist	RS1	RS2	RS	S1	S2	S	RL1	RL2	RL	L1	L2	L
N(I_4),C(I_4)	.159	.225	.192	.899	.007	.453	.344	.402	.373	.151	.495	.323
(50,50)	.045	.033	.023	.091	.007	.043	.120	.114	.110	.106	.164	.087
N(I_4),C(I_4)	.159	.264	.212	.884	.009	.446	.314	.442	.378	.097	.554	.326
(25,75)	.062	.049	.029	.138	.012	.063	.124	.126	.115	.082	.171	.079
N(I_4),t_2(I_4)	.177	.241	.209	.627	.046	.337	.215	.256	.236	.172	.282	.227
(50,50)	.049	.039	.021	.177	.033	.075	.078	.068	.056	.060	.074	.053
N(I_4),t_2(I_4)	.195	.291	.243	.658	.040	.349	.201	.270	.235	.186	.259	.222
(25,75)	.072	.052	.038	.159	.023	.069	.076	.068	.053	.057	.067	.047
C(I_4),t_2(I_4)	.384	.303	.343	.087	.705	.396	.393	.340	.367	.487	.214	.351
(50,50)	.047	.057	.036	.076	.175	.064	.111	.103	.094	.170	.100	.083
C(I_4),t_2(I_4)	.402	.315	.359	.154	.665	.410	.431	.353	.392	.609	.161	.385
(25,75)	.073	.065	.037	.137	.214	.060	.115	.115	.106	.178	.092	.078
N(I_4),C(W)	.108	.174	.141	.881	.002	.442	.314	.401	.357	.085	.509	.297
(50,50)	.037	.034	.017	.137	.004	.067	.128	.137	.124	.066	.191	.079
N(I_4),C(W)	.107	.227	.167	.911	.002	.457	.341	.453	.397	.104	.526	.315
(25,75)	.054	.051	.026	.108	.003	.053	.122	.152	.118	.087	.173	.078
N(I_4),t_2(W)	.121	.150	.135	.529	.013	.271	.174	.218	.196	.164	.226	.195
(50,50)	.041	.036	.018	.191	.013	.091	.054	.080	.058	.047	.103	.057
N(I_4),t_2(W)	.123	.190	.156	.618	.009	.313	.179	.224	.201	.172	.209	.190
(25,75)	.044	.043	.017	.163	.010	.078	.069	.073	.051	.055	.078	.048
C(I_4),t_2(W)	.346	.291	.319	.121	.585	.353	.397	.381	.389	.485	.278	.381
(50,50)	.069	.063	.043	.106	.171	.059	.113	.117	.107	.172	.120	.108
C(I_4),t_2(W)	.335	.284	.310	.190	.476	.333	.378	.347	.362	.539	.188	.363
(25,75)	.063	.070	.044	.105	.175	.057	.133	.115	.112	.194	.095	.104

Table 5 Continued

Dist	RQ1	RQ2	RQ	Q1	Q2	Q	RM1	RM2	RM	M1	M2	M
N(I_4),C(I_4)	.165	.203	.184	.007	.396	.201	.142	.171	.156	.077	.233	.155
(50,50)	.055	.039	.022	.009	.063	.028	.050	.038	.021	.036	.042	.020
N(I_4),C(I_4)	.162	.233	.197	.012	.411	.211	.179	.170	.174	.141	.206	.174
(25,75)	.066	.042	.028	.019	.073	.030	.064	.032	.027	.067	.052	.027
N(I_4),t_2(I_4)	.207	.217	.212	.090	.314	.202	.190	.192	.191	.158	.228	.193
(50,50)	.047	.036	.024	.042	.081	.025	.058	.037	.025	.050	.054	.028
N(I_4),t_2(I_4)	.221	.244	.233	.130	.307	.218	.227	.209	.218	.230	.219	.224
(25,75)	.070	.047	.029	.065	.074	.031	.073	.047	.039	.082	.054	.040
C(I_4),t_2(I_4)	.385	.325	.355	.665	.088	.376	.327	.303	.315	.406	.246	.326
(50,50)	.046	.056	.038	.067	.032	.022	.065	.053	.021	.081	.050	.029
C(I_4),t_2(I_4)	.389	.303	.346	.664	.115	.389	.343	.308	.325	.432	.220	.326
(25,75)	.073	.067	.041	.119	.125	.036	.081	.057	.034	.085	.047	.028
N(I_4),C(W)	.111	.154	.132	.003	.293	.148	.103	.101	.102	.035	.171	.103
(50,50)	.040	.033	.017	.009	.058	.027	.035	.029	.014	.018	.033	.013
N(I_4),C(W)	.119	.170	.145	.004	.297	.151	.137	.114	.125	.094	.140	.117
(25,75)	.053	.040	.022	.005	.041	.019	.057	.032	.022	.063	.039	.231
N(I_4),t_2(W)	.131	.144	.137	.041	.241	.141	.111	.114	.112	.086	.139	.113
(50,50)	.040	.036	.019	.021	.061	.023	.039	.033	.015	.028	.039	.014
N(I_4),t_2(W)	.129	.186	.157	.070	.222	.146	.131	.126	.128	.140	.131	.136
(25,75)	.053	.036	.019	.044	.081	.026	.052	.047	.028	.063	.043	.029
C(I_4),t_2(W)	.370	.314	.342	.646	.114	.380	.243	.256	.250	.271	.233	.252
(50,50)	.074	.066	.056	.103	.044	.037	.053	.057	.031	.070	.053	.032
C(I_4),t_2(W)	.360	.308	.334	.643	.115	.379	.287	.232	.259	.331	.182	.257
(25,75)	.076	.084	.060	.162	.105	.054	.068	.042	.028	.069	.031	.030