NEW SOLUTION ALGORITHMS FOR THE CLASSIFICATION PROBLEM

By

WILLIAM JAMES BANKS, B.A., C.A., M.B.A.

A Dissertation

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Doctor of Philosophy

in Business Administration

McMaster University

NEW SOLUTION ALGORITHMS FOR THE CLASSIFICATION PROBLEM

DESCRIPTIVE NOTE

DOCTOR OF PHILOSOPHY (1991)
(Business Administration)

McMASTER UNIVERSITY
Hamilton, Ontario

TITLE: New Solution Algorithms for the Classification Problem

AUTHOR: William James Banks
B.A. (University of Guelph)
C.A. (Province of Ontario)
M.B.A. (York University)

SUPERVISOR: Dr. Prakash L.Abad

NUMBER OF PAGES: ix,193

ABSTRACT

This thesis proposes a new mixed integer programming model of the two group classification problem. A branch-and-bound procedure is designed to solve the mixed integer programming model. The proposed procedure is more efficient than recently published procedures in the literature.

Three heuristics are also designed from the branch-and-bound procedure. The Three Seed and N Seed heuristics provide solutions that are "close" to the optimal solution for substantially less computational effort.

This thesis also proposes a quadratic transformation method which provides a solution that is competitive with the classical quadratic discriminant analysis method.

ACKNOWLEDGEMENTS

I am deeply indebted to Dr. Prakash L. Abad, my thesis supervisor for his patience and direction regarding my academic development. His encouragement was particularly valuable during times when my morale was low.

I also wish to acknowledge the efforts of my committee members, Dr. G.O. Wesolowsky and Dr. I. Krinsky, for their advice and patience throughout the dissertation process.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The purpose of this research is to further the literature regarding the development of linear programming models that provide a linear decision rule as a solution to the two group classification problem. The two group classification problem, sometimes referred to as the two group discriminant problem, is the assignment of an observation into one of two populations based on the attributes of the observation. The solution to the classification problem is a decision rule, based on historical samples from the two populations, that assigns a new observation to the population that is most likely to be correct. The classification rule minimizes the total expected cost of misclassification.

The classification problem occurs in a number of areas such as business, accounting, artificial intelligence and psychology. Recent business examples include a study by Cooper (1979) predicting new product success, a study by Capon (1982) examining credit scoring models and a study by

Kaplan and Urwitz (1979) predicting bond ratings. A study in the accounting area by Stone and Rasp (1991) examines the tradeoff in classification techniques for accounting choice studies. A study in the artificial intelligence area by Bobrowski (1986) examines classification with symmetric models. A study in the psychology area was performed by Lastovicka, Murry, Joachimsthaler, Bhalla and Scheurich (1987) which examined the prediction of adolescent drinking-and-driving behaviour.

The first classification technique was proposed by Fisher (1936). Fisher's technique develops a linear decision rule. Smith (1947) proposed a quadratic decision rule. These formulations, along with several mathematical programming formulations are discussed in the review of the literature section.

In this research, a new mixed integer programming model of the two-group classification problem is proposed and a branch-and-bound solution method is designed for solving the model. Three related linear programming heuristics are also introduced. Particular attention in studying these new methods is focused on computational effort. In addition to computational effort, the heuristic solutions are compared for nearness to the optimal solution.

Also, a new quadratic transformation solution method is developed. The method employs a quadratic transformation of the observation data and a variable selection criterion prior to solving the problem utilizing the proposed linear programming heuristics.

As outlined in Glover, Keene and Duea (1988) linear programming (LP) classification models offer several advantages over the classical multivariate model. Linear programming models are:

1. free from underlying parametric assumptions,
2. extendable to integer domains and more complex problem formulations,
3. less sensitive to outliers since the model is based on the L_1 metric rather than the L_2 metric,
4. capable of assigning individual weighting to each of the observations in determining boundary placement, and
5. capable of measuring the solution sensitivity to the placement and weighting of individual observations.

Chapter 2 of this research presents a review of Fisher's linear discriminant analysis method, Smith's quadratic discriminant analysis method and the research into mathematical programming solution methods. Chapter 3 discusses the proposed model and branch-and-bound algorithm. Also the new heuristics are described. In chapter 4 there is an outline of the experimental design and in chapter 5 the experimental results are presented. The quadratic transformation method and experimentation is included in chapter 6. A discussion of the conclusions of this research is included in chapter 7.

The contributions of this study are:

1) the proposal of a new optimization model and branch-and-bound procedure that is efficient in terms of computational effort when compared to existing mathematical programming methods,
2) the proposal of a heuristic solution that generates "near" optimal solutions and is efficient in terms of computational effort and
3) the proposal of a quadratic transformation method that improves the linear mathematical programming solution methods for comparison to quadratic methods such as Smith's quadratic discriminant analysis.

A summary of the notation used in this research is shown in the appendix to chapter 1.

APPENDIX TO CHAPTER 1

NOTATION

A_{ij} variable representing known value of attribute j for observation i

C is the unknown cutoff value (unrestricted on sign)

D relative priorities

F_p is the prior probability of being from group p

H_p is the cost of misclassifying an observation from group p

LPN is a large positive number constant

M_p is the number of observations in group p

Q_i = (1 if observation i is the greatest distance from
(the cutoff hyperplane
(0 otherwise

R_i = (1 if observation i is misclassified
(0 otherwise

S_i represents the correct classification deviation

SPN small positive number

T_i represents the misclassification deviation

W_j is the unknown weight for attribute j (unrestricted on sign)

Y_i is the dependent variable in the Logit method

Z is the value of the objective function

$$\alpha = \frac{F_2H_2}{M_2} / \frac{F_1H_1}{M_1}$$

CHAPTER 2

REVIEW OF THE LITERATURE

There are a number of methods discussed in the literature as solutions to the two group classification problem. In section 2.1, some of the classical classification methods are described and in section 2.2 the mathematical programming methods are described. In section 2.3 a number of studies comparing the different methods are outlined along with the conclusions of the researchers. A summary is included in section 2.4.

2.1 Classical Classification Techniques

In this section, a discussion of Fisher's (1936) linear discriminant analysis (LDA), Smith's (1947) quadratic discriminant analysis (QDA) and the logistic discriminant function (logit) are included. These three methods have been used as "bench marks" in some of the mathematical programming comparative studies discussed in section 2.3.

2.1.1 Linear Discriminant Analysis (LDA)

The first solution technique to the classification problem was proposed by Fisher(1936) and is the most widely known technique. The technique minimizes the probability of misclassifications under certain assumptions. The Fisher technique, called discriminant analysis, is included in the popular software package SPSSX (Statistical Package for the Social Sciences).

Before discussing the LDA method, the assumptions of the technique as summarized in Klecka (1980) are shown below. Each observation i is described by several characteristics or attributes $(A_{i1}, A_{i2}, \ldots, A_{ij})$.

1. Each group is drawn from a population with a multivariate normal distribution on the attributes of the observations.
2. The variance-covariance matrices for each group must be equal.
3. The attributes of the observations must be independent of each other.
4. The two groups must be mutually exclusive (an observation can only belong to one group).
5. There are at least two statistically independent cases per group.

6. There can be any number of attributes for each observation provided that the number of attributes is less than the total number of observations minus two.

7. The observation attributes are measured at the interval level (ie: they are continuous).

The form of the linear discriminant function is shown below:

$$\mathrm{SCORE}_i = \sum_{j=1}^{J} W_j A_{ij} \qquad (2.1)$$

where:
A_{ij} is the known value for observation i of attribute j
W_j is the unknown weight for attribute j (unrestricted on sign)

The unknown weights (ie: W_j variables) are chosen so that the values of SCORE_i differ as much as possible between the groups. To do this we maximize the ratio:

$$\frac{\text{between-groups sum of squares}}{\text{within-groups sum of squares}} \qquad (2.2)$$

The next step in LDA is to determine a rule to classify the observations into groups given the above linear discriminant function. As discussed in Norusis (1985), the method used by the popular software package SPSSX uses

Baye's rule. Probabilities are calculated assuming the linear discriminant function values ($SCORE_i$) are normally distributed. The objective is to classify the observation to the group with the highest probability given the linear discriminant function value. The probabilities are shown below:

$$\text{Probability}(G_p|SCORE_i) = \frac{\text{Probability}(SCORE_i|G_p)\ F_p}{\sum_{p=1}^{2} \text{Probability}(SCORE_i|G_p)\ F_p} \quad (2.3)$$

where:
F_p is the prior probability of being from group p
G_p represents group p (p=1,2)

The prior probability, F_p is an estimate of the likelihood that an observation belongs to group p when we do not know the value of the linear discriminant value. F_p corresponds to the probabilities in the population. Since each observation must belong to one and only one of the two groups, the prior probabilities must sum to one.

An observation is assigned to a group, based on the linear discriminant function value, for which the posterior probability (ie: probability($G_p|SCORE_i$)) is the largest. The Baye's theorem approach, according to Lachenbruch (1975), minimizes the total probability of a misclassification.

Unequal costs for misclassification can be incorporated into the classification rule by adjusting the prior probabilities. In many cases it is more serious to make one kind of misclassification than another. Lachenbruch (1975) establishes that minimizing the total misclassification costs with unequal costs is equivalent to minimizing the total error rate in the equal costs case but using cost adjusted prior probabilities. The cost adjusted prior probabilities of group membership for the two group case are:

$$\text{ADJUSTED } F_p = \frac{H_p F_p}{H_1 F_1 + H_2 F_2} \quad \text{for } p=1,2 \qquad (2.4)$$

where:
H_p the cost of misclassifying an observation from group p

Eisenbeis (1977) and Eisenbeis (1978) outline several pitfalls in the application of LDA in business, finance and economics. Of the applied LDA papers appearing in the literature many have suffered from statistical problems that have limited the practical usefulness of their results. Problems occur when the assumptions of the model are violated. The general criticism of the LDA procedure is carried forward by Joy and Tollefson (1975), (1979), Altman and Eisenbeis (1978) and Scott(1978).

Tiku, Tan and Balakrishnan (1986) discuss the effects on LDA of a violation of the assumption of normality. They conclude that linear discriminant analysis is not robust to departures from normality. This conclusion is supported by Lachenbruch (1975). A statistical procedure such as LDA has traditionally been called robust if both its type I error (significance level) and its type II error (power) are not affected drastically by departures from normality.

In summary the LDA approach provides the optimal linear solution if the assumptions are not violated. It is a heuristic if the assumptions are violated.

2.1.2 Quadratic Discriminant Analysis (QDA)

Quadratic discriminant analysis (QDA) was proposed by Smith (1947). QDA is included in the IMSL (1987) (International Mathematical and Statistical Library) FORTRAN subroutines. QDA is described in Lachenbruch (1975) and Dillon and Goldstein (1984).

The assumption in LDA that the variance-covariance matrices for the two groups must be equal for each group is relaxed in QDA. The assumption regarding multivariate normal populations and the other assumptions of LDA hold for QDA.

The first step develops a quadratic discriminant function rather than the linear discriminant function of LDA. The quadratic discriminant function for the two observation attribute case is shown below:

$$SCORE_i = W_1(A_{i1})^2 + W_2A_{i1}A_{i2} + W_3(A_{i2})^2 + W_4A_{i1} + W_5A_{i2} + C \quad (2.5)$$

The next step, similar to LDA, assigns the observation to the group that has the highest posterior probability.

Lachenbruch (1975) concluded in his research that deviations from normality tend to affect the QDA function seriously. He states that QDA is not robust to non-normality, particularly if the distribution has longer tails than the normal distribution.

2.1.3 Logit Method

The logit method, which is a conditional probability model, has received support as a classification method. Wiginton (1980) proposed the logit method as an alternative to the LDA method to classify credit applicants.

The logit method is based on using the logistic function and maximum likelihood estimation, to estimate the logit parameters. The development of the logistic function is shown in Aldrich and Nelson (1984). An excellent summary description of Logit is included in Mahmood and Lawrence (1987). The logistic function is:

$$\text{Probability}(Y_i=1) = \frac{\exp(\text{SCORE}_i)}{1+\exp(\text{SCORE}_i)} = \frac{1}{1+\exp(-\text{SCORE}_i)} \quad (2.6)$$

where:

Y_i is the dependant variable for observation i and is restricted to binary 0,1

$$\text{SCORE}_i = \sum_{j=1}^{J} W_j A_{ij}$$

The logistic function is a continuous probability function and can take on any value between 0 and 1. If the probability of Y_i=1 is greater than .5 than the observation is classified as a group 2 observation. Otherwise the observation is classified as a group 1 observation. This assumes that group 1 observations are represented by 0 and group 2 observations are represented by 1.

The values of the unknown weights, W_j's in SCORE_i are determined using the maximum likelihood estimation method. The development of the unknown weights is shown in Aldrich and Nelson (1984).

After the unknown W_j's have been determined in the first step, then the logistic function is used in the second step to classify the observations into one of two groups. This is accomplished by first calculating $SCORE_i$, the linear combination of observation attributes, and using this value in the logistic function.

The assumptions of the two group logit model, as stated in Aldrich and Nelson (1984) are:

1. The dependent random variable is assumed to be binary.
2. The outcomes on Y are assumed to be mutually exclusive and collectively exhaustive.
3. Y is assumed to depend on the attributes of the observation.
4. The observations are statistically independent of each other.
5. There are no exact or near linear dependencies among the attributes of the observation.

The LDA and QDA assumption of multivariate normal distribution on the attributes of the observations and the LDA assumption of the equality of the variance-covariance matrices for each group have been dropped for the logit model. Therefore the logit model appears to have greater practical usefulness than the LDA or QDA methods, due to the elimination of two important assumptions.

2.2 Mathematical Programming Methods

The mathematical programming methods to solve the two group classification problem are reviewed in this section. Unlike the classical methods, mathematical programming procedures are relatively recent. The first methods were proposed by Smith (1968)(1969), Liittschwager and Wang (1978), Hand (1981) and Freed and Glover (1981).

Mathematical programming methods do not have the assumptions of the classical methods. The observations do not need to be drawn from multivariate normal distributions and the variance-covariance matrices for the groups do not need to be equal. Also the attributes of the observations do not need to be independent of each other. This lack of assumptions should provide mathematical programming methods with an advantage, since the classical assumptions can't always be satisfied in "real world" classification problems.

Mathematical programming methods are single step methods in comparison to the classical methods that are two step. In this single step method the unknown weights, W_j's and the classification of an observation to a group are performed simultaneously. A classification hyperplane is developed.

The development of the classification hyperplane is discussed in section 2.2.1. The optimal solution methods are discussed in section 2.2.2 and the heuristic solution methods are discussed in section 2.2.3.

2.2.1 Classification Hyperplane

Mathematical programming models develop a classification hyperplane that separates the two groups. The classification hyperplane is given by:

$$\sum_{j=1}^{J} W_j A_{ij} = C \qquad (2.7)$$

where:
A_{ij} is the variable term representing the known value of attribute j for observation i
C is the unknown cutoff level (unrestricted on sign)
W_j are the unknown weights (unrestricted on sign)

The values of W_j and C are determined simultaneously from historic samples from the populations.

The distance of an observation i with attribute values $(A_{i1}, A_{i2}, \ldots, A_{ij})$ from the hyperplane is given by the absolute value of the difference between $\Sigma W_j A_{ij}$ and C of the above equation.

Observations for which the absolute value of the difference is zero are on the hyperplane and are considered correctly classified in this research. In Koehler and Erenguc (1990) only the observations not on the hyperplane are termed misclassified if they do not belong to the group specified by the discriminant process. This research is consistent with the Koehler and Erenguc (1990) research. In Rubin (1990) these observations on the hyperplane are rated as one-half of a misclassification. The counting of observations on the hyperplane as not misclassified will result in a positive bias in our misclassification rates when compared to Rubin (1990).

2.2.2 Optimization Methods

This section includes a discussion of the mathematical programming methods that are formulated to provide the optimal solution. They are presented in historical order.

2.2.2.1 Liittschwager and Wang (1978)

The Liittschwager and Wang (1978) formulation is a mixed integer program. The integer variables are binary. Their model is shown on table 2.1.

TABLE 2.1

LIITTSCHWAGER AND WANG (1978) MODEL

MINIMIZE: $Z = F_1H_1 \sum_{i=1}^{M_1} R_i/M_1 + F_2H_2 \sum_{i=M_i+1}^{M_1+M_2} R_i/M_2$

SUBJECT TO:

(1) $\sum_{j=1}^{J} W_jA_{ij} \leq C + LPN\ R_i$ for $i=1,2,\ldots,M_1$

(2) $\sum_{j=1}^{J} W_jA_{ij} \geq C - LPN\ R_i$ for $i=M_1+1,\ldots,M_1+M_2$

(3) $-1 + 2B_j \leq W_j \leq 1 - 2E_j$ for $j=1,2,\ldots,J$

(4) $\sum_{j=1}^{J} (B_j + E_j) = 1$

where:

A_{ij} is the value for observation i of attribute j
B_j binary normalization unknown (0/1)
C unknown cutoff value (unrestricted on sign)
E_j binary normalization unknown (0/1)
F_p prior probability of being from group p
H_p cost of misclassifying an observation from group p
LPN large positive number
M_p the number of observations in group p
R_i = (1 if observation i is misclassified
(0 otherwise
W_j the unknown weight for attribute j (unrestricted on sign)
Z value of the objective function

The objective function minimizes the expected total cost of misclassification. The R_i terms are 0-1 variables. The value of R_i is to be 1 if observation i is misclassified and is to be 0 if observation i is correctly classified. The sample probability that a group p observation is misclassified can be estimated by $\Sigma R_i/M_p$ (where i is summed over the group p observations). An observation has prior probabilities of F_1 of being from population group 1 and F_2 from population group 2 with $F_1+F_2=1$. The costs of misclassifying an observation from population group 1 and from population group 2 are G_1 and G_2, respectively. Therefore, the objective function minimizes the expected total cost of misclassification.

Constraints (1) and (2) are used to develop the hyperplane between the two groups. Since it may be impossible to classify all observations correctly, using the cutoff value C, the constraint includes the binary minimization variable, R_i, multiplied by a large constant, LPN, to allow for misclassifications. The lower bound on LPN formulated in Liittschwager and Wang (1978) is:

$$\text{LOWER BOUND ON LPN} = 2J \, \text{MAX}_{ij} \, |A_{ij}| \qquad (2.8)$$

$$\text{for } i=1,2,\ldots,M_1+M_2$$
$$\text{for } j=1,2,\ldots,J$$

Liittschwager and Wang included constraints (3) and (4) to ensure that the trivial null solution of the C and all the W_j's being equal to zero, is avoided. The constraints force at least one W_j to be equal to +1 or -1. If one W_j is equal to +1 or -1 then at least one attribute will be considered in the classification. This adds 2J binary unknown variables to the formulation.

This formulation is a mixed integer problem with M_1+M_2+2J integer variables and J+1 continuous variables. Therefore, there is a finite but large number of feasible solutions. Due to the large number of feasible solutions for their model and the related computational burden, only small problems can be effectively solved using the generalized branch-and-bound algorithms that are included in commercial linear programming software packages such as LINDO. For large problems, a heuristic is more suitable. Liittschwager and Wang (1978) developed a heuristic for the two group classification problem which has only two observation attributes. This heuristic is too limiting for "real world" problems that have several observation attributes.

2.2.2.2 Bajgier and Hill (1982)

Bajgier and Hill (1982) also explored the mixed integer formulation to solve the two group classification problem. Their formulation is shown on table 2.2.

The objective function of the Bajgier and Hill (1982) model minimizes the total number of misclassifications (R_i), minimizes the sum of misclassification deviations (T_i) and maximizes the sum of correct classification deviations (S_i). These objectives are weighted by the researcher depending on the relative importance of each. Constraints (1) and (2) set the values for C and the W_j's. Constraint (3) ensures that if the misclassification deviation is greater than zero then the integer variable, R_i which counts the number of misclassifications is equal to one.

As discussed in Rubin (1990a) Bajgier and Hill avoided the trivial null solution (ie: where the W_j's and C terms are equal to zero) by fixing C at an arbitrary positive value.

Bajgier and Hill stated that computational time was burdensome for this zero-one formulation and that computational time grows very rapidly with the number of zero-one variables. They conclude that the mixed integer

TABLE 2.2

BAJGIER AND HILL (1982) MODEL

MINIMIZE: $Z = D_1 \sum_{i=1}^{M_1+M_2} R_i + D_2 \sum_{i=1}^{M_1+M_2} T_i - D_3 \sum_{i=1}^{M_1+M_2} S_i$

SUBJECT TO:

(1) $\sum_{j=1}^{J} W_j A_{ij} + S_i - T_i = C$ for $i=1,2,\ldots,M_1$

(2) $\sum_{j=1}^{J} W_j A_{ij} - S_i + T_i = C$ for $i=M_1+1,\ldots,M_1+M_2$

(3) $LPN\ R_i \geq T_i$ for $i=1,2,\ldots,M_1+M_2$

where:

A_{ij}	the value of observation i of attribute j
D_1, D_2, D_3	relative priorities set by the researcher
C	unknown cutoff value (unrestricted on sign)
LPN	large positive number
M_p	the number of observations in group p
R_i	= (1 if observation i is misclassified (0 otherwise
S_i	represents the correct classification deviation
T_i	represents the misclassification deviation
W_j	the unknown weight for attribute j (unrestricted on sign)
Z	the value of the objective function

approach is unlikely to be practical for classification problems unless special purpose algorithms or heuristics are developed.

2.2.2.3 Wang (1982)

Wang (1982) also presented a mixed integer formulation as a solution to the classification problem, similar to the Liittschwager and Wang (1978) formulation but with substantially more binary variables. The Wang (1982) formulation results in M_1xM_2 binary unknown variables. He states that the use of generalized branch-and-bound procedures to solve this formulation is quite difficult except when the sample size is very small. Wang develops an efficient algorithm for the two group classification problem which has only two observation attributes.

2.2.2.4 Showers and Chakrin (1981) & Kolesar and Showers (1985)

Showers and Chakrin (1981) and Kolesar and Showers (1985) formulated a credit screening model using categorical data as a mixed integer programming model. The model is formulated to classify new telephone customers into a group that should pay a deposit and a group in which a deposit is not necessary.

The objective of their formulation was that the decision rule must be simple and easy to implement. Therefore, the known observation attributes (customer responses) and unknown weights were restricted to zero or one. Also the unknown cutoff was restricted to integer. This binary/integer restriction is not included in other mathematical programming formulations.

A difficulty with their approach is that the sample size must be very large to provide sufficient information to estimate the model parameters. Their sample consisted of 87,000 observations each with 72 binary attributes.

2.2.2.5 Koehler and Erenguc (1990)

Recently Koehler and Erenguc (1990) proposed a mathematical programming model and branch-and-bound algorithm to solve the two group classification problem. Their research includes an analysis of the computational effort required to solve the problem. The Koehler and Erenguc (1990) research will be used as a comparison for the proposed new optimal model and solution algorithm developed in this thesis.

In order to develop their algorithm, Koehler and Erenguc introduce several mathematical programming models.

Their main model, Practical Minimal Misclassifications (PMM), is shown on table 2.3.

The objective function minimizes the total number of misclassifications. Constraints (1) and (2) are the same as those included in the Liittschwager and Wang (1978) model (see table 2.1).

Constraint (3) is used to avoid the trivial null solution. This constraint sets one of the i observations as misclassified. In the design of their specialized branch-and-bound search procedure each of the i observations are in turn set as misclassified. They enumerate.

Since constraint (3) forces the solution to have at least one misclassification, the model is pre-tested by Koehler and Erenguc to determine if there is complete separation between the groups.

In order to solve the PMM model, Koehler and Erenguc develop the DREPMM model (Dual of the Relaxed Equivalent PMM model) along with a branch-and-bound algorithm. Their selection criterion and branching process is described in Koehler and Erenguc (1990).

A difficulty with the Koehler and Erenguc (1990) model is the selection of an appropriate value for LPN (ie: some positive number). As the value of LPN increases, then the value returned for the objective function (Z) decreases.

TABLE 2.3

KOEHLER AND ERENGUC (1990)
PMM MODEL

$$\text{MIN: } Z = \sum_{i=1}^{M_1} R_i + \sum_{i=M_1+1}^{M_1+M_2} R_i$$

SUBJECT TO:

(1) $\sum_{j=1}^{J} W_j A_{ij} \geq C - LPN\ R_i$ for $i=1,2,\ldots,M_1$

(2) $\sum_{j=1}^{J} W_j A_{ij} \leq C + LPN\ R_i$ for $i=M_1+1,\ldots,M_1+M_2$

(3) for at least one i:

$\sum_{j=1}^{J} W_j A_{ij} \leq C - SPN$ for $i=1,2,\ldots,M_1$

$\sum_{j=1}^{J} W_j A_{ij} \geq C + SPN$ for $i=M_1+1,\ldots,M_1+M_2$

where:

A_{ij} the value for observation i of attribute j
C the unknown cutoff value (unrestricted on sign)
R_i = (1 if observation i is misclassified
(0 otherwise
LPN large positive number
M_p the number of observations in group p
SPN small positive number relative to LPN
W_j unknown weight for attribute j (unrestricted on sign)
Z value of the objective function

In other words, as LPN increases then the probability of fathoming a particular subset of solutions (eliminate from further consideration), due to exceeding the upper bound in the branch-and-bound algorithm, will diminish. A second problem is to select an appropriate value for SPN (ie: some small positive number relative to LPN).

Koehler and Erenguc state that they know of no a priori method for choosing these values. They point out that they rely on a standard maxim in mixed integer programming to choose LPN large enough and SPN small enough. The LPN must be sufficiently large to make constraint (1) or (2) feasible when otherwise it would be violated.

In section 5 of this thesis, the results of the Koehler and Erenguc method are compared to the results of the new methods proposed in section 3 of this research.

2.2.2.6 Rubin (1990a)

Rubin's mixed integer model is shown on table 2.4.

The objective function minimizes the number of misclassifications and maximizes the minimum absolute deviation of a correctly classified deviation. The second term in the objective function is added since there may be many solutions with the same minimal misclassification rate on the training misclassification rate, but not all equally

TABLE 2.4

RUBIN (1990a) MODEL

MINIMIZE: $Z = \sum_{i=1}^{M_1+M_2} R_i - SPN\ B$

SUBJECT TO:

(1) $\sum_{j=1}^{J} W_j A_{ij} + B \leq C + LPN\ R_i$ for $i=1,2,\ldots,M_1$

(2) $\sum_{j=1}^{J} W_j A_{ij} - B \geq C - LPN\ R_i$ for $i=M_1+1,\ldots,M_1+M_2$

(3) $-1 \leq W_j \leq 1$ for $j=1,2,\ldots,J$

(4) $B \geq SPN$

where:

A_{ij} is the known value for observation i of attribute j

B unknown value representing the minimum absolute deviation of a correctly classified observation

C unknown cutoff value (unrestricted on sign)

LPN large positive number

R_i = (1 if observation i is misclassified
(0 otherwise

SPN small positive number

W_j the unknown weight for attribute j (unrestricted on sign)

Z value of the objective function

proficient in classify the holdout observations.

The trivial null solution is avoided by constraint (4) which ensures that at least one observation is correctly classified. The W_j's are restricted to the interval between -1 to +1.

Rubin develops heuristics based on his MIP model. They are discussed in the next section.

2.2.3 Heuristic Methods

This section includes a discussion of the mathematical programming methods that are formulated to provide a heuristic solution to the two group classification problem. They are presented in historical order. A number of studies have been performed comparing these mathematical programming methods with some of the classical methods. These studies are reviewed in section 2.3.

2.2.3.1 Freed and Glover (1981)

Freed and Glover (1981) proposed a series of four, two group heuristic classification models based on linear programming. Their first model (ie: called model A) is shown on table 2.5.

The objective function, minimizes the sum of misclassification deviations balanced by relative priorities ($\Sigma D_{1i}T_i$) and maximizes the sum of correct classification

TABLE 2.5

FREED AND GLOVER (1981)
MODEL A

MINIMIZE: $Z = \sum_{i=1}^{M_1+M_2} D_{1i}T_i - \sum_{i=1}^{M_1+M_2} D_{2i}S_i$

SUBJECT TO:

(1) $\sum_{j=1}^{J} W_jA_{ij} + S_i - T_i = C$ for $i=1,2,\ldots,M_1$

(2) $\sum_{j=1}^{J} W_jA_{ij} - S_i + T_i = C$ for $i=M_1+1,\ldots,M_1+M_2$

(3) $W_1 \geq 1$

where:

A_{ij} is the value of observation i for attribute j
C unknown cutoff value (unrestricted on sign)
D_{1i}, D_{2i} relative priorities for each observation
M_p the number of observations in group p
S_i represents the correct classification deviation
T_i represents the misclassification deviation
W_j unknown weight for attribute j (unrestricted on sign)
Z the objective function value

deviations balanced by relative priorities $(\Sigma D_{2i}S_i)$. Constraints (1) and (2) are used to develop the hyperplane between the two groups. Constraint (3) is proposed by Freed and Glover (1981) to prevent the trivial null solution where the C and W_j's are all equal to zero. This does not allow for the situation where $W_1 \leq 0$.

Freed and Glover's other models are variations of their model A allowing for maximum deviation rather the sum of deviations.

There has been a considerable reaction in the literature to Freed and Glover's paper. Glorfeld and Gaither (1982) are concerned about the purported simplicity of the classification by linear programming over classification by linear discriminant analysis (LDA). Freed and Glover (1982) responded to this criticism by stating that the answer to simplicity is subjective but that the average user is overmatched when confronted by formal statistical theory. In addition, the mathematical programming method requires little in the way of formal inferential theory and is virtually free of qualifying assumptions.

Markowski and Markowski (1985) further explored the approach by Freed and Glover and concluded that the placement of the origin can seriously effect the model's capacity to classify properly. More generally, the appearance of observations in all four quadrants promotes a breakdown of the formulation

2.2.3.2 Freed and Glover (1986)

Freed and Glover (1986) proposed a new class of heuristic linear programming procedures to solve the two group classification problem. Their paper presents three contrasting formulations. The formulations are 1) minimize maximum deviation (MMD), 2) minimize the sum of interior distances (MSID) and 3) minimize the sum of deviations (MSD). The models were designed by Freed and Glover (1986) to "provide generally satisfactory discriminators". Their MSD model is shown on table 2.6.

The objective function of the MSD model minimizes the sum of deviations, T_i, of misclassified observations from the cutoff value C. Constraints (1) and (2) are similar to those of Liittschwager and Wang (1978) except the minimization variable, T_i is continuous not binary.

Constraint (3) was added by Freed and Glover (1986) to prevent the trivial null solution by ensuring that not all of the weights and the cutoff are equal to zero. Freed and Glover for their research chose CONSTANT=10. However, they state that CONSTANT=1 or CONSTANT=100 would work as

TABLE 2.6

FREED AND GLOVER (1986)
MSD MODEL

MINIMIZE: $Z = \sum_{i=1}^{M_1+M_2} T_i$

SUBJECT TO:

(1) $\sum_{j=1}^{J} W_j A_{ij} \leq C + T_i$ for $i=1,2,\ldots,M_1$

(2) $\sum_{j=1}^{J} W_j A_{ij} \geq C - T_i$ for $i=M_1+1,\ldots,M_1+M_2$

(3) $\sum_{j=1}^{J} W_j + C =$ CONSTANT

where:
A_{ij} the value for observation i of attribute j
C unknown cutoff value (unrestricted on sign)
M_p the number of observations in group p
T_i represents the misclassification deviation
W_j unknown weight for attribute j (unrestricted on sign)
Z value of the objective function

well since CONSTANT only serves to scale the solution.

Duea (1985) in his doctoral thesis examined three possible methods (the methods were proposed by Dr. F. Glover) to avoid the trivial null solution using a version of Freed and Glover's model A (shown in table 2.5) to minimize the sum of deviations. The three methods are:

1) $C = \text{CONSTANT}$ (2.9)

2) $\sum_{j=1}^{J} W_j = \text{CONSTANT}$ (2.10)

3) $\sum_{j=1}^{J} W_j + C = \text{CONSTANT}$ (2.11)

Duea concludes that the second method is preferable in that it generates the same solution for all arithmetic shifts of the problem within the first quadrant. Duea also concludes that as the problem is rotated about the origin the relative importance of the attributes changes. The best results were obtained when both attributes carried equal weights in determining the value of the classification hyperplane. He does note, however that the problem needs to be solved twice, once with a positive CONSTANT and once with a negative CONSTANT.

2.2.3.3 Glover, Keene and Duea (1988)

Glover, Keene and Duea (GKD)(1988) proposed a new hybrid discriminant model as a heuristic solution. Their hybrid model 4 is shown on table 2.6. The heuristic solution to the classification problem is provided by solving this single linear program. However the GKD model requires that the researcher choose a parameterization for the model before it is used. GKD in their paper outline four such parameterizations and state that further research is required to determine which parameterization is the "best" for which situations.

In section 5, the results of the GKD hybrid model 4 are included as a comparison to the results obtained by the methods proposed in section 3 of this research.

A model similar to GKD is also developed independently in Banks (1988) where it is called the Two Group Classification Linear Programming Model (TGCLP). The models are identical for the case having equal cost of misclassification, equal prior probabilities of group membership and equal sample sizes from the two groups.

TABLE 2.7

GLOVER, KEENE & DUEA (GKD)(1988)
HYBRID MODEL 4

$$\text{MIN: } Z = \sum_{i=1}^{M_1+M_2} b_i T_i - \sum_{i=1}^{M_1+M_2} d_i S_i$$

SUBJECT TO:

$$(1) \quad \sum_{j=1}^{J} W_j A_{ij} + S_i - T_i = C \quad \text{for } i=1,2,\ldots,M_1$$

$$(2) \quad \sum_{j=1}^{J} W_j A_{ij} - S_i + T_i = C \quad \text{for } i=M_1+1,\ldots,M_1+M_2$$

$$(3) \quad \sum_{i=1}^{M_1+M_2} S_i = 10$$

Parameterization used in section 4.2 of this thesis:

$b_i = 1$
$d_i = 0$

Notation:

A_{ij} is the value of observation i of attribute j
C is the unknown cutoff value (unrestricted on sign)
M_p is the number of observations in group p
S_i represents the correct classification deviation
T_i represents the misclassification deviation
W_j is the unknown weight for attribute j (unrestricted on sign)
Z is the value of the objective function

2.2.3.4 Glover (1990)

Glover (1990) discusses the limitations of constraint (3) in the GKD hybrid model. He introduces a new constraint (shown below) to prevent the trivial null solution.

$$\sum_{i=1}^{M_1+M_2} S_i - \sum_{i=1}^{M_1+M_2} T_i = \text{CONSTANT} \quad (2.12)$$

2.2.3.5 Koehler and Erenguc (1990)

Recently, Koehler and Erenguc (1990) proposed a new heuristic called the Breadth Scan of the Practical Minimal Misclassification model (BPMM). The BPMM heuristic is really a partial search procedure based upon their PMM model. The Practical Minimal Misclassification model (PMM) used in their algorithm is shown on table 2.3. Their algorithm requires the solution to several linear programs whereas the GKD model only requires the solution to one linear program.

Koehler and Erenguc compared their heuristic linear solution to the optimal solution in a systematic experiment. The experimental factors were distribution kurtosis and variance along with procedure. The BPMM results were very close to the optimal linear solution. The difficulty with the Koehler and Erenguc model is that two parameters must be specified a priori.

Koehler and Erenguc also included a summary of the computational effort required by their BPMM heuristic. This summary is used, in section 5, as a comparison to the new methods proposed in this thesis (section 3).

2.2.3.6 Rubin (1990a)

Rubin (1990a) developed two heuristics based on his MIP model discussed in section 2.2.2.6. He tested the two heuristics in a systematic experiment along with his MIP model, LDA and the GKD hybrid model. Rubin used an 'easy' and a 'hard' treatment with four different training sample sizes.

The mixed integer solution and the heuristic solutions were more accurate on the training samples than were the LDA or GKD hybrid model. For the holdout sample, the mixed integer solution and the heuristic solutions were more accurate for the case when the correlation between the groups was high, the populations had unequal covariance matrices and the separation of the populations was low.

He concludes that his second heuristic is a cost-efficient alternative to the MIP model when large training sample sizes are available.

2.3 Comparative Studies

There are several recent published papers such as Mahmood and Lawrence (1987), Markowski and Markowski (1987), Joachimsthaler and Stam (1988), Rubin (1990), Koehler and Erengu. (1990) and Stam and Joachimsthaler (1990), comparing the various two group mathematical programming classification methods. These papers all include Fisher's (1936) LDA method as a "bench mark". Some papers also include Smith's (1947) QDA method and Logit as "bench marks". Several of these papers are discussed below.

2.3.1 Joachimsthaler and Stam (1988)

Joachimsthaler and Stam (1988) performed a systematic simulation experiment comparing LDA, QDA, Logit and the Freed and Glover (1986) MSD model shown on table 2.6.

The Joachimsthaler and Stam experiment consisted of 3 attributes for each observation with a training sample (ie: estimation or calibration sample) size of 50 observations from each group. The experimental factors were discriminant procedure (4 levels), distribution kurtosis (4 levels) and variance-covariance homogeneity (3 levels). There are 48 factor combinations. They did not test their results using a holdout (ie: validation) sample.

A summary of the Joachimsthaler and Stam results are shown on table 2.8.

The conclusions stated in Joachimsthaler and Stam (1988) study are: 1) the LDA, Logit and MSD procedures tend to produce very similar results, 2) QDA produced significantly lower misclassification rates than the other methods as the dispersion heterogeneity between the groups increased and 3) as the kurtosis increases, the Logit and MSD methods produced consistently lower misclassification rates (consistent with Glorfeld and Olsen (1982) study).

These conclusions are based upon only the tests using the training sample. They are likely to be biased when considering conclusions for the holdout sample.

TABLE 2.8

JOACHIMSTHALER AND STAM (1988)
TRAINING SAMPLE MISCLASSIFICATION RATE RESULTS

C	VH 1	VH 2	Group Mean 1	Group Mean 2	DK	LDA	QDA	LOG	MSD
1	1	1	0	.5	-1	.329	.326	.329	.332
2	1	1	0	.5	0	.319	.316	.324	.325
3	1	1	0	.5	1	.317	.317	.318	.318
4	1	1	0	.5	3	.308	.288	.312	.310
5	1	2	0	.6	-1	.314	.253	.324	.325
6	1	2	0	.6	0	.319	.262	.322	.314
7	1	2	0	.6	1	.282	.232	.317	.311
8	1	2	0	.6	3	.303	.274	.307	.307
9	1	4	0	.8	-1	.300	.127	.303	.289
10	1	4	0	.8	0	.289	.169	.298	.286
11	1	4	0	.8	1	.276	.178	.293	.282
12	1	4	0	.8	3	.280	.208	.281	.279
Mean: All Cells						.303	.246	.311	.307
Mean: Cells 1,2,3,4,10						.312	.283	.316	.314

where:

C	cell
VH	variance heterogeneity
DK	distribution kurtosis
LDA	linear discriminant analysis
QDA	quadratic discriminant analysis
LOG	logit
MSD	minimize sum of deviations

2.3.2 Koehler and Erenguc (1990)

Koehler and Erenguc (1990) compared their proposed optimal method (PMM) and proposed heuristic method (BPMM) to LDA using the same experimental design as Joachimsthaler and Stam (1988). Both training sample and holdout sample results were developed. There results are summarized on table 2.9.

Both the PMM optimal solution method and the BPMM heuristic solution method gave significantly lower training sample misclassification results than LDA. However, LDA misclassified significantly fewer observations in the holdout samples when the dispersion matrices are equal than did PMM or BPMM. But as the dispersion heterogeneity increased, PMM and BPMM gave smaller average misclassification rates than did LDA. In a very small experiment (1 problem per case for eight cases) Koehler and Erenguc investigated the effect of increasing the training sample size on the performance of the holdout sample. They conjectured from this small experiment that as the training sample size increases then the BPMM heuristic generates fewer holdout misclassifications than does LDA.

TABLE 2.9

KOEHLER AND ERENGUC (1990)
MISCLASSIFICATION RATE RESULTS

C	VH 1	VH 2	Group Mean 1	Group Mean 2	DK	TRAINING LDA	TRAINING BPMM	TRAINING PMM	HOLDOUT LDA	HOLDOUT BPMM	HOLDOUT PMM
1	1	1	0	.5	-1	.329	.273	.265	.353	.368	.370
2	1	1	0	.5	0	.323	.266	.258	.345	.363	.362
3	1	1	0	.5	1	.319	.262	.253	.337	.357	.355
4	1	1	0	.5	3	.311	.252	.243	.327	.344	.344
5	1	2	0	.6	-1	.343	.264	.255	.369	.354	.353
6	1	2	0	.6	0	.336	.262	.255	.362	.356	.354
7	1	2	0	.6	1	.331	.261	.252	.356	.352	.351
8	1	2	0	.6	3	.324	.256	.246	.347	.347	.349
9	1	4	0	.8	-1	.323	.225	.219	.352	.311	.309
10	1	4	0	.8	0	.316	.226	.220	.342	.318	.314
11	1	4	0	.8	1	.313	.226	.220	.333	.316	.312
12	1	4	0	.8	3	.305	.228	.220	.321	.315	.310
Mean: All Cells						.323	.250	.242	.345	.342	.340
Mean: Cells 1,2,3,4,10						.320	.256	.248	.341	.350	.349

where:
C cell
VH variance heterogeneity
DK distribution kurtosis
LDA linear discriminant analysis
BPMM breadth scan heuristic
PMM practical minimal misclassifications model

2.3.3 Rubin (1990)

Rubin (1990) compared a series of linear programming methods to parametric approaches in a systematic experiment. The two parametric approaches were LDA and QDA. The mathematical programming methods tested included two variations of the MSD model of Freed and Glover (1986) and eight parameterizations of the Hybrid model of Glover, Keene and Duea (1988). Rubin used both training and validation samples.

Rubin's primary conclusion was that Smith's (1947) QDA method is superior to all the linear programming methods tested. He further states that if linear programming methods are to be considered as an alternative to conventional procedures then they must be shown to outperform QDA.

2.3.4 Stam and Jones (1990)

Stam and Jones investigated the performance on small sample and medium-size samples of several techniques to solve the classification problem. Training and holdout samples were used. They conclude that for small samples (less than 60 observations in total from both groups) that the minimize sum of distances model (MSD model shown on table 2.6) and the LDA model are preferred over QDA.

2.4 Summary

A survey of the literature is included in Joachimsthaler and Stam (1990) and in Erenguc and Koehler (1990).

As a summary of the literature, the focus should be on the research of Glover, Keene and Duea (1988) and the research of Koehler and Erenguc (1990). Glover, Keene and Duea (1988) have presented a new hybrid model that appears promising but requires further investigation. Koehler and Erenguc (1990) have presented the most efficient (in a relative sense) method to date in the literature of deriving the optimal solution. They included an analysis of the computational effort in their article. Also they developed an efficient heuristic that generates near optimal solutions. This research will consider these methods in a series of systematic experiments along with LDA and QDA.

CHAPTER 3

PROPOSED METHODS

In this section a new mixed integer programming model of the classification problem is developed. A branch-and-bound algorithm is proposed to efficiently solve the model. Three heuristics are also presented.

The new optimization method is discussed in section 3.1 and the new heuristics are discussed in section 3.2.

3.1 Proposed Optimization Method

In this section a mixed integer programming (MIP) model is developed along with the branch-and-bound algorithm. This is also described in Banks and Abad (1991). The MIP model is shown on table 3.1.

The objective function is the same as that developed in Liittschwager and Wang (1978) and is rewritten below:

$$\text{MINIMIZE: } Z = F_1H_1 \sum_{i=1}^{M_1} R_i/M_1 + F_2H_2 \sum_{i=M_1+1}^{M_1+M_2} R_i/M_2 \qquad (3.1)$$

As stated in section 2.2.2.1, the objective function represents the expected cost of misclassification. The R_i

TABLE 3.1

MIXED INTEGER PROGRAMMING (MIP) MODEL

$$\text{MIN: } Z = \sum_{i=1}^{M_1} R_i + \alpha \sum_{i=M_1+1}^{M_1+M_2} R_i$$

SUBJECT TO:

(1) $\sum_{j=1}^{J} W_j A_{ij} + S_i - T_i = C$ for $i=1,2,\ldots,M_1$

(2) $\sum_{j=1}^{J} W_j A_{ij} - S_i + T_i = C$ for $i=M_1+1,\ldots,M_1+M_2$

(3) $R_i \geq T_i$ for $i=1,2,\ldots,M_1+M_2$

(4) $T_i \geq Q_i$ for $i=1,2,\ldots,M_1+M_2$

(5) $\sum_{i=1}^{M_1+M_2} Q_i = 1$

(6) $\text{FACTOR}(1-R_i) \geq S_i$ for $i=1,2,\ldots,M_1+M_2$

(7) $S_i \geq 0$ for $i=1,2,\ldots,M_1+M_2$

(8) $T_i \geq 0$ for $i=1,2,\ldots,M_1+M_2$

Notation:

A_{ij} is the value for observation i of attribute j

C is the unknown cutoff value (unrestricted on sign)

F_p is the prior probability of being from group p

H_p cost of misclassifying an observation from group p

M_p is the number of observations in group p

Q_i = (1 if observation i is the greatest distance
(from the cutoff hyperplane
(0 otherwise

R_i = (1 if observation i is misclassified
(0 otherwise

S_i represents the correct classification deviation

T_i represents the misclassification deviation

W_j unknown weight for attribute j (unrestricted on sign)

$\alpha = (F_2H_2/M_2)/(F_1H_1/M_1)$

terms are 0-1 variables. The value of R_i is to be 1 if observation i is misclassified and is to be 0 if observation i is correctly classified. The $\Sigma R_i/M_p$ term represents the proportion of misclassifications in sample group p. In order to obtain the expected misclassification proportion for the population, the sample proportions are multiplied by the population prior probabilities of group membership, F_p. Including costs of misclassification, H_p, the objective function represents the expected cost of misclassification.

Constraints (1) and (2) are used to determine the value of the W_j's and C for the classification hyperplane; $\Sigma W_j A_{ij}=C$. As will be shown later, S_i and T_i are complementary variables. If an observation is correctly classified the T_i term should be equal to zero and the S_i term (correct classification deviation) represents the absolute value of the difference between the cutoff value (C) and the value of the weighted linear combination of attributes ($\Sigma W_j A_{ij}$). If an observation is misclassified then the S_i term should be equal to zero and the T_i term (misclassification deviation) represents the absolute value of the difference between the cutoff value and the value of the weighted linear combination of attributes.

The purposes of constraints (3), (4) and (5) are to prevent the trivial null solution and to scale the solution.

The trivial null solution occurs when all the W_j terms are simultaneously equal to zero. With these constraints, the trivial null solution is non-optimal since for at least one observation in constraints (1) and (2); T_i=1 and S_i=0. This forces, in the event that all W_j terms are simultaneously equal to zero, that the C terms be equal to +1 or -1. In this event all the observations of one group are misclassified (ie: no separation). This is clearly a non-optimal solution.

With constraints (3), (4) and (5) the single observation which has the greatest misclassification deviation in the solution, has the deviation set equal to one unit. This scaling of the greatest misclassification deviation improves the fathoming speed in the related branch-and-bound search procedure and will be discussed later. These constraints also result in the following relationship:

$$0 \leq Q_i \leq T_i \leq R_i \leq 1 \qquad (3.2)$$

If, for example a particular Q_i is equal to one (Q_i is a binary variable) then the related T_i is equal to one and the related R_i is equal to one. If, however a particular Q_i is equal to zero, the related T_i can range between zero and one. If the related T_i is equal to zero then the R_i is also equal to zero since we are minimizing R_i in the objective function. This relationship is important in that during the standard branch-and-bound solution process for MIP's, the relaxed LP solution value will always be less than or equal to the corresponding integer solution.

Constraint (6) ensures that only one of S_i or T_i can be positive for observation i. This is important, since it ensures that an observation cannot be both correctly classified and misclassified at the same time in the solution. For example, if T_i is positive then R_i=1 and S_i=0. If T_i is equal to zero then R_i=0 and S_i may be zero or any positive number with an upper bound of FACTOR. Remember, S_i is the correct classification deviation. For large values of FACTOR, the upper bound on S_i is effectively eliminated. Therefore FACTOR should be set at a large value (ie: 10,000).

Constraints (7) and (8) ensure that S_i and T_i are non-negative.

This model does not provide for the case where there is complete separation between the groups (ie: no misclassifications). Complete separation could be tested for prior to using the model. A complete separation model is shown in Koehler and Erenguc (1990).

The linear programming (LP) model shown on table 3.2 is the relaxation of the MIP model. The objective function and constraints (1), (2) and (3) of the LP model are identical to those of the MIP model, except that the binary restriction on the R_i terms has been relaxed. Constraint (4) ensures that in the relaxation, the R_i terms do not exceed one.

Constraints (5) and (6) have been added to the LP model as a substitute for constraints (4), (5) and (6) in the MIP model. The new constraints combined with the proposed Enumeration/Branch-and-Bound (E/BAB) algorithm, presented later, will prevent the trivial null solution and scale the solution.

Constraints (7) and (8) are identical to the MIP model and are to ensure that S_i and T_i are non-negative.

The next three models are introduced to progressively further reduce the number of variables and constraints in the formulation. The purpose of this progression is to reduce the computational burden of deriving the optimal solution. A reduction of variables and constraints should achieve this purpose.

TABLE 3.2

LINEAR PROGRAMMING (LP) MODEL

$$\text{MIN: } Z = \sum_{i=1}^{M_1} R_i + \alpha \sum_{i=M_1+1}^{M_1+M_2} R_i$$

SUBJECT TO:

(1) $\sum_{j=1}^{J} W_j A_{ij} + S_i - T_i = C$ for $i=1,2,\ldots,M_1$

(2) $\sum_{j=1}^{J} W_j A_{ij} - S_i + T_i = C$ for $i=M_1+1,\ldots,M_1+M_2$

(3) $R_i \geq T_i$ for $i=1,2,\ldots,M_1+M_2$

(4) $R_i \leq 1$ for $i=1,2,\ldots,M_1+M_2$

(5) $T_k = 1$ (for some k in $\{1,2,\ldots,M_1+M_2\}$)

(6) $S_k = 0$

(7) $S_i \geq 0$ for $i=1,2,\ldots,M_1+M_2$

(8) $T_i \geq 0$ for $i=1,2,\ldots,M_1+M_2$

The equivalent linear programming (ELP) model, shown on table 3.3, was accomplished by preforming two substitutions in the LP model. The first substitution was to replace all R_i terms by their related T_i terms. In the MIP model both the R_i and T_i terms are necessary to calculate the integer value of the objective function. However, since the LP model is minimizing the linear combination of the continuous R_i terms then the R_i terms will always be equal to their related T_i terms. The second substitution was to eliminate the T_i terms using the equality constraints (constraints 1 and 2) in the LP model. The T_i terms are replaced by a function of $\{W_j, S_i$ and $C\}$.

Changes to the variable coefficients are used to force the greatest misclassification deviation to be equal to one unit, rather than constraints (5) and (6) of the LP model. This is easily accomplished by changing the right-hand side of ELP constraint (3) or (4) from 0 to 1 for the selected observation. Also the related S_i term must be removed from the model.

A modified ELP (MELP) model is shown on table 3.4. The only difference between the ELP and the MELP model is that the S_i variables in constraints (1) and (2) of the ELP model have been eliminated in the MELP model. The purpose of this modification will be discussed with the dual of the model.

TABLE 3.3

EQUIVALENT LINEAR PROGRAMMING (ELP) MODEL

$$\text{MIN: } Z = \sum_{i=1}^{M_1} \left(\sum_{j=1}^{J} W_j A_{ij} + S_i - C\right) + \alpha \sum_{i=M_1+1}^{M_1+M_2} \left(\sum_{j=1}^{J} -W_j A_{ij} + S_i + C\right)$$

SUBJECT TO:

$$(1) \quad \sum_{j=1}^{J} W_j A_{ij} + S_i - C \leq 1 \qquad \text{for } i=1,2,\ldots,M_1$$

$$(2) \quad -\sum_{j=1}^{J} W_j A_{ij} + S_i + C \leq 1 \qquad \text{for } i=M_1+1,\ldots,M_1+M_2$$

$$(3) \quad \sum_{j=1}^{J} W_j A_{ij} + S_i - C \geq 0 \qquad \text{for } i=1,2,\ldots,M_1$$

$$(4) \quad -\sum_{j=1}^{J} W_j A_{ij} + S_i + C \geq 0 \qquad \text{for } i=M_1+1,\ldots,M_1+M_2$$

$$(5) \quad S_i \geq 0 \qquad \text{for } i=1,2,\ldots,M_1+M_2$$

Note:
To force a particular observation (ie: k) to have the greatest misclassification deviation (ie: one unit) the following changes are made to the model:

- change the RHS of constraint (3) or (4) for observation i=k from 0 to 1
- eliminate the S_k variable from the model

TABLE 3.4

MODIFIED EQUIVALENT LINEAR PROGRAMMING (MELP) MODEL

$$\text{MIN: } Z = \sum_{i=1}^{M_1} \left(\sum_{j=1}^{J} W_j A_{ij} + S_i - C\right) + \alpha \sum_{i=M_1+1}^{M_1+M_2} \left(\sum_{j=1}^{J} -W_j A_{ij} + S_i + C\right)$$

SUBJECT TO:

$$(1) \quad \sum_{j=1}^{J} W_j A_{ij} \quad - \; C \quad \leq \; 1 \quad \text{for } i=1,2,\ldots,M_1$$

$$(2) \quad -\sum_{j=1}^{J} W_j A_{ij} \quad + \; C \quad \leq \; 1 \quad \text{for } i=M_1+1,\ldots,M_1+M_2$$

$$(3) \quad \sum_{j=1}^{J} W_j A_{ij} \; + S_i \; - \; C \quad \geq \; 0 \quad \text{for } i=1,2,\ldots,M_1$$

$$(4) \quad -\sum_{j=1}^{J} W_j A_{ij} \; + S_i \; + \; C \quad \geq \; 0 \quad \text{for } i=M_1+1,\ldots,M_1+M_2$$

$$(5) \quad S_i \; \geq \; 0 \quad \text{for } i=1,2,\ldots,M_1+M_2$$

Notes:

1. To force a particular observation (ie: k) to have the greatest misclassification deviation (ie: one unit) the following changes are made to the model:
   - change the RHS of constraint (3) or (4) for observation i=k from 0 to 1
   - eliminate the S_k variable from the model
2. The objective function can be rewritten as:

$$\text{MIN: } Z = \sum_{j=1}^{J} W_j \left(\sum_{i=1}^{M_1} A_{ij} - \alpha \sum_{i=M_1+1}^{M_1+M_2} A_{ij} \right)$$

$$+ \left(-M_1 + \alpha M_2 \right) C \quad + \quad \sum_{i=1}^{M_1} S_i \; + \; \alpha \sum_{i=M_1+1}^{M_1+M_2} S_i$$

In the following theorem, we show that the MELP model is equivalent to the ELP model. We also show that S_i and T_i variables in the LP model are complimentary variables.

THEOREM 1:

The optimal solution to the ELP and MELP models are identical.

PROOF:

Since S_i ,i belongs to $\{1,2,\ldots,M_1+M_2\}$, have positive coefficients in the objective function, clearly in any optimal solution to ELP:

$$S_i = \max\{0, C-\Sigma W_j A_{ij}\} \quad \text{for each } i \text{ in } \{1,2,\ldots,M_1\} \qquad \text{(A)}$$

and

$$S_i = \max\{0, \Sigma W_j A_{ij}-C\} \quad \text{for each } i \text{ in } \{M_1+1,\ldots,M_1+M_2\} \qquad \text{(B)}$$

Substituting (A) in constraint (1) of ELP in the following manner:

$$0 \le 1 + C - \Sigma\, W_j A_{ij} \quad \text{and}$$

$$C - \Sigma\, W_j A_{ij} \le 1 + C - \Sigma\, W_j A_{ij}$$

yields constraint (1) of MELP.

In a similar manner substituting (B) in constraint (2) of ELP for each S_i, where i belongs to $\{M_1+1,\ldots,M_1+M_2\}$ yields constraint (2) of MELP.

OBSERVATION 1:

S_i and T_i in the LP model are complementary variables.

The proof follows directly from (A) and (B) above and from the definition of T_i from constraints (1) and (2) of the LP model.

The resulting MELP model has M_1+M_2+J+1 variables and $2(M_1+M_2)$ constraints. The dual of the MELP model (DMELP) has $2(M_1+M_2)$ variables and M_1+M_2+J+1 constraints. Therefore, as discussed in Hillier and Lieberman (p.194)(1980), there should be less computational difficulty to find a solution to the DMELP model than the MELP model since there are fewer constraints than variables. The DMELP model is shown on table 3.5.

It should be noted that M_1+M_2 constraints in the DMELP model are simple upper bounds. The simple upper bounds are a direct result of eliminating the S_i terms in constraints (1) and (2) of the ELP model. The significance of using simple upper bounds is explained in the discussion of major and minor pivots in the experimental design, section 4.1. The resulting DMELP model, then has $2(M_1+M_2)$ variables, J+1 constraints and M_1+M_2 simple upper bounds.

In the DMELP model an observation is easily forced to be the greatest misclassification deviation of one unit by changing the coefficient in the objective function for V_i from 0 to 1. Also the simple upper bound for the particular V_i term must be discarded.

TABLE 3.5

DUAL MODIFIED EQUIVALENT LINEAR PROGRAMMING (DMELP) MODEL

MAX: $Z = -\sum_{i=1}^{M_1+M_2} Y_i$

SUBJECT TO:

(1) $-\sum_{i=1}^{M_1} A_{ij}(Y_i - V_i) + \sum_{i=M_1+1}^{M_1+M_2} A_{ij}(Y_i - V_i)$

$= \sum_{i=1}^{M_1} A_{ij} - \alpha \sum_{i=M_1+1}^{M_1+M_2} A_{ij}$ for j=1,2,...,J

(2) $\sum_{i=1}^{M_1} (Y_i - V_i) - \sum_{i=M_1+1}^{M_1+M_2} (Y_i - V_i) = - M_1 + \alpha M_2$

SIMPLE UPPER BOUNDS:

$V_i \leq 1$ for $i=1,2,\ldots,M_1$

$V_i \leq \alpha$ for $i=M_1+1,\ldots,M_1+M_2$

Note:

1. To force a particular observation (ie: k) to have the greatest misclassification deviation (ie: one unit) the following changes are made to the model:
   - change the coefficient in the objective function for V_k from 0 to 1 (This is equivalent to changing the RHS of constraint (3) or (4) in the MELP from 0 to 1 for i=k.)
   - eliminate the simple upper bound for variable k (This is equivalent to eliminating the S_k variable (ie:S_k=0) in the MELP model.)

2. Subsequently to set a particular observation to be misclassified, remove the simple upper bound on the particular V_i term and adjust the RHS's to remove the fractional miscl. deviation value from Z and add a constant related to misclassfication deviation.

An enumeration/branch-and-bound (E/BAB) algorithm has been formulated to provide the solution to the classification problem. The algorithm is shown on tables 3.6 and 3.7.

There are two stages in the E/BAB algorithm. The first stage is the enumeration stage (table 3.6). In this stage, the algorithm sets each observation in turn to have the largest misclassification deviation, that being one unit. For each enumeration a feasible integer solution is calculated and the upper bound is adjusted when necessary. There are M_1+M_2 subsets enumerated of which a large portion are fathomed. The remaining subsets along with the upper bound are carried forward to the second stage, the branch-and-bound algorithm.

In the second stage, the branch-and-bound algorithm (table 3.7), the remaining subset with the lowest objective function value is selected. Within that subset the remaining observation with the greatest misclassification deviation is selected for the partitioning.

TABLE 3.6

ENUMERATION ALGORITHM

1. INITIALIZATION STEP
1.1 Set the upper bound on the objective function, Z_U=+infinity
1.2 Set enumeration index k=0

2. ENUMERATION STEP
2.1 Enumeration index k=k+1

3. BOUND STEP
3.1 For each new subset, obtain a lower bound (Z_L) on the value of the objective function for the feasible solutions in the subset by solving the DMELP model. Z_L=Z (value of objective function) for this subset.
3.2 Also obtain a feasible integer solution (Z_R) for the subset[1]. If $Z_R<Z_U$ then reset $Z_U=Z_R$.

4. FATHOMING STEP
For each new subset, exclude it from further consideration in the branch-and-bound algorithm if:
Fathoming Test 1: $Z_L>Z_U$
Fathoming Test 2: The subset is found to contain no bounded solutions (primal infeasible).

5. STOPPING RULE
5.1 If $k<M_1+M_2$ then return to the enumeration step.
5.2 Otherwise reapply fathoming test 1 to all remaining subsets then stop (start branch-and-bound algorithm).

The calculation of Z_R is accomplished by:
1. Determining the W_j's and C from the dual prices of the LINDO results.
2. Calculate the related deviation values using the LINDO User's Subroutine.
3. If the misclassification deviation is greater than zero then the observation is misclassified.
4. Calculate the Z_R based on the number of misclassifications.

TABLE 3.7

BRANCH-AND-BOUND ALGORITHM

1. INITIALIZATION STEP

1.1 Set the upper bound on the objective function, Z_U= value from enumeration algorithm.

1.2 Before beginning the regular iterations through the steps below, apply just the fathoming step and stopping rule to the remaining subsets from the enumeration algorithm.

2. BRANCH STEP

2.1 Subset Selection

2.1.1 Select from the remaining subsets (those neither fathomed nor partitioned) the subset that has the lowest Z_L value.

2.2 Observation Selection for Partitioning

2.2.1 Select the observation with the largest misclassification deviation value, which has not previously been used in partitioning.

2.2.2 Set the selected observation as correctly classified or misclassified (see note 1 below).

3. BOUND STEP

3.1 For each new subset, obtain a lower bound (Z_L) on the value of the objective function for the feasible solutions in the subset by solving the DMELP model. Z_L=Z (value of objective function) for this subset.

3.2 Also obtain a feasible integer solution (Z_R) for the subset.

4. FATHOMING STEP

For each new subset, exclude it from further consideration (ie: fathom it) if:

Fathoming Test 1: $Z_L \geq Z_U-(\min\{1,\alpha\})$ (see note 2 below)

Fathoming Test 2: The subset is found to contain no bounded solutions (primal infeasible).

Fathoming Test 3: The solution to the subset is binary ($Z_L=Z_R$) and the best solution in the subset has been identified. If this occurs and $Z_R<Z_U$ then set $Z_U=Z_R$, store this solution as the incumbent solution and reapply Test 1 to remaining subsets.

Fathoming Test 4: If $Z_R<Z_U$ then reset $Z_U=Z_R$, store this solution as the incumbent solution and reapply Test 1 to remaining subsets.

TABLE 3.7 (CONTINUED)

5. STOPPING RULE

5.1 Stop when there are no remaining (unfathomed) subsets. The current incumbent solution is the optimal solution.

5.2 Otherwise return to the branch step.

Note 1: Partitioning Method:

1.) Correctly Classified

LP Model:	In the LP model to set as correctly classified we add the constraint $T_i=0$.
ELP Model:	This is accomplished in the ELP model by altering the RHS of the selected constraint (depending on the observation) from a value of 1 to 0.
DMELP Model:	In the DMELP model this is equivalent to changing the coefficient of the Y_i term in the objective function from -1 to 0.

2.) Misclassified

LP Model:	In the LP model to set as misclassified we add the constraints $S_i=0$ and $R_i=1$.
ELP MODEL:	This is accomplished in the ELP model by adding the constraint $S_i=0$ and adjusting the objective function. The objective function must be adjusted to remove the fractional T_i value from Z and add a constant value related to R_i. The constant is 1 if the observation is from $\{1,2,...,M_1\}$ and α otherwise.
DMELP Model:	In the DMELP model we remove the simple upper bound on the related V_i term and adjust the RHS's to remove the fractional T_i value from Z and add a constant related to R_i.

Note 2:

$$\alpha = \frac{F_2H_2}{M_2} \Big/ \frac{F_1H_1}{M_1}$$

In the enumeration/branch-and-bound (E/BAB) algorithm LP relaxations are solved by solving DMELP. It is believed that even though enumeration is used as the first stage of the algorithm, that the improved fathoming speed in the branch-and-bound (second stage) justifies it. The improved fathoming speed is related to setting the greatest misclassification deviation to be equal to one rather then being less than (or equal to) one. This increases the Z value (scaled) returned and therefore increases the lower bound (Z_L). Z_L is compared to the upper bound (Z_U) in the fathoming test. Z_L will always be less than or equal to the corresponding integer solution Z_R. If Z_R is less than Z_U then the upper bound is reduced.

A fortran subroutine was coded within the commercial linear programming package LINDO to handle the proposed optimal solution algorithm. The fortran code is shown in appendix A.

3.2 Proposed Heuristic Methods

In this section three new heuristics based on the proposed optimal solution method are presented. These heuristics are also developed in Abad and Banks (1991).

The proposed heuristics are divided into two stages. The first stage uses the Glover, Keene and Duea (1988)(GKD) hybrid model heuristic to provide an initial solution. This initial solution is then used to determine the seed observations that will be used for the second stage. The second stage uses the DMELP model (formulated in section 3.1) to improve the solution. The algorithm requires several LP's to be solved, similar to the BPMM heuristic of Koehler and Erenguc (1990).

Stage 1

As mentioned stage 1 uses the GKD hybrid model heuristic. In particular their hybrid model 4 is chosen with the parameterization having $b_i=1$ and $d_i=0$. The GKD hybrid model heuristic is shown on table 2.6.

The initial solution provided by the GKD hybrid model provides the information necessary to determine the seed observations for the next stage of the algorithm. The purpose of the seed observation is discussed (following) in stage 2. The first seed observation is the misclassified observation that is the farthest from the classification hyperplane, $\Sigma W_j A_{ij}=C$; (ie: the observation for which the misclassification deviation $| \Sigma_j W_j A_{ij} - C |$ is maximum).

The second seed observation is the second farthest misclassified observation from the classification hyperplane and the third seed observation is the third farthest misclassified observation from the classification hyperplane.

Stage 2

As previously mentioned the second stage uses the DMELP model formulated in section 3.1.

Constraints (5) and (6) of the LP model (Table 3.2) set some observation to have the farthest misclassification distance of one unit. In the proposed optimal solution method each observation is in turn set to be the farthest misclassified observation in the enumeration/ branch-and-bound algorithm (Table 3.6). This enumeration is necessary for the optimal solution but not for the heuristic solution. The first two proposed heuristics use the first or first three farthest misclassified observations in the solution to the GKD hybrid model (called seed observations) as the farthest misclassified observations in a limited enumeration/branch-and-bound search procedure. These two heuristics are called Single Seed (SS) and Three Seed (TS). The third heuristic called N Seed (NS) uses all the misclassified observations in the GKD hybrid model solution as candidates for the farthest misclassified observation in the limited enumeration/branch-and-bound search procedure. The heuristics are described in detail below.

3.2.1 Single Seed Heuristic (SS)

The SS heuristic is described below:

1. Solve the classification problem using the GKD hybrid model heuristic.

2. Determine the farthest misclassified observation from the GKD solution and set this as the seed observation.

3. Solve the DMELP model (table 3.5) formulation of the problem with the seed observation set as the farthest misclassified observation.

4. Determine from this resulting solution the observation that is farthest from the new classification hyperplane but that is not the seed observation and has not previously been set as misclassified in step 5.

5. Set this observation as misclassified using the procedure described in note 2 in table 3.5.

6. Solve the DMELP model formulation of the problem with the seed observation as the farthest and all set observations as misclassified.

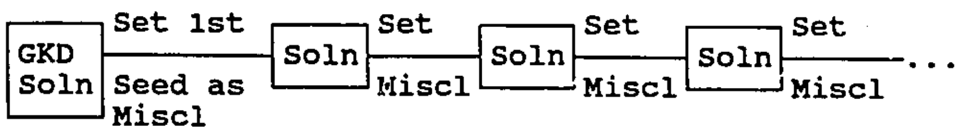
7. Repeat steps 4, 5 and 6 until all the observations in the solution that are misclassified have been set as misclassified.

8. Stop.

The SS heuristic results in the pattern of solutions shown in figure 3.1.

FIGURE 3.1

SINGLE SEED HEURISTIC SOLUTION PATTERN

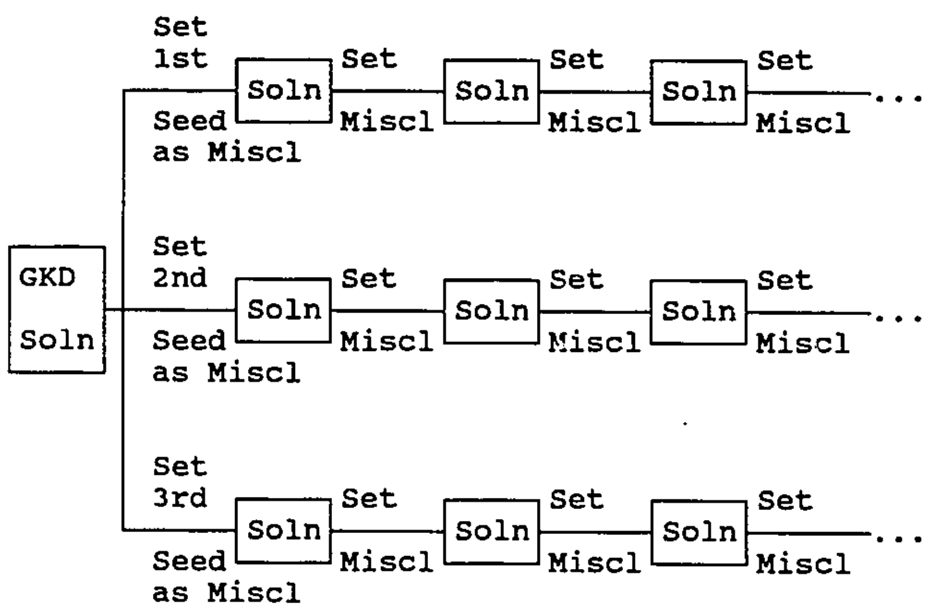


3.2.2 Three Seeds Heuristic (TS)

The TS heuristic is similar to the SS heuristic except that instead of only one start we use three starts to the algorithm as shown in figure 3.2. The related integer solution (Z_R) for each LP relaxation solved is also computed in order establish an upper bound (Z_U) that can be used to prevent unnecessary computations. The related integer solution (Z_R) is computed by assuming all misclassified deviations greater than zero are misclassified observations. If the solution to the LP relaxation (ie. Z_L) is greater than Z_U at any node on a branch, then that branch is fathomed.

FIGURE 3.2

THREE SEED HEURISTIC SOLUTION PATTERN



3.2.3 N Seeds Heuristic (NS)

The NS heuristic is similar to the TS heuristic except that instead of restricting ourselves to only three misclassified observations from the GKD solution we use all the misclassified observations of the GKD solution.

A fortran subroutine was coded within the commercial linear programming software package LINDO to handle the proposed heuristics.

CHAPTER 4

EXPERIMENTAL DESIGN

A systematic experiment is performed to compare the results of the proposed optimal solution method and proposed heuristic solution methods to the recent optimal solution method and heuristic solution method of Koehler and Erenguc (1990). Also included in the study are solutions developed using linear discriminant analysis (LDA) and the Glover, Keene and Duea (1988)(GKD) hybrid model heuristic.

In order to make a comparison to the recently published results of Koehler and Erenguc (1990) the same experimental design as used by Koehler and Erenguc (1990) is also used in this research. Koehler and Erenguc (1990) compared their results to the results of Joachimsthaler and Stam (1988) using the same experimental design as developed in Joachimsthaler and Stam (1988). Note, however that the problems in this research are generated under different hardware and software conditions than the Joachimsthaler and Stan (1988) research and Koehler and Erenguc (1990) research and therefore are not identical.

The following areas are examined in the experimental results (chapter 5):

1. The computational effort of the proposed optimal solution method is compared to the computational effort of the optimal solution method of Koehler and Erenguc (1990). Computational effort is defined in section 4.1.

2. The computational effort of the proposed SS, TS and NS heuristic solution methods are compared to the computational effort of the proposed optimal solution method and the proposed heuristic of Koehler and Erenguc (1990).

3. The proposed SS, TS and NS heuristic solution mean misclassification rate results are compared to the optimal solution.

4.1 Computational Effort

In the experimentation the same criterion as developed in Koehler and Erenguc (1990) is used to measure computational effort. The Koehler and Erenguc measures of computational effort are the number of major pivots, the number of minor pivots and the number of LP subsets that are solved in the branch-and-bound algorithm. Also since there are more variables in the proposed models then in the Koehler and Erenguc (1990) models, a discussion of work per pivot using the commercial software package LINDO is included.

As defined by Koehler and Erenguc (1990) a minor pivot is one in which the selected non-basic variable remains non-basic by going from its upper bound to the lower bound or from its lower bound to its upper bound. A major pivot is one where the selected non-basic variable is pivoted into the basis.

In the DMELP model (table 3.5), the dual V_i variables have simple upper bounds. Also, the V_i variables have lower bounds of zero (ie: V_i terms are restricted to be non-negative). If in a pivot the V_i variable changes from the lower bound of zero to the upper bound or changes from the upper bound to the lower bound of zero then this is a minor pivot.

Minor pivots, according to Zionts (Ch.8)(1974), are computationally less expensive because of the efficiency of the method of upper bounds. Zionts (1974) further explains that an upper bound iteration (ie: minor pivot) is accomplished by simply substituting for the entering variable. The upper bound method is included in most commercial linear programming packages such as LINDO.

There are M_1+M_2 more variables in the proposed DMELP model than in the Koehler and Erenguc (1990) model. These additional M_1+M_2 variables, however, all have zero coefficients in the objective function. LINDO and most LP software packages use the revised simplex method with the product form inverse.

As outlined in Hillier and Lieberman (p. 194)(1980) there are three factors that affect the computational performance of the model. They are:

1) the number of functional constraints.
2) the number of variables.
3) the density of the table of constraint coefficients.

Density is defined as the proportion of the constraint coefficients that are non-zero. The density effects the computational effort per pivot whereas the number of variables effects the number of pivots. The proposed DMELP model has the same number of functional constraints and has the same density as the Koehler and Erenguc (1990) model. The difference relates to the number of variables. Hillier and Lieberman (1980) continue in their discussion to state that the number of variables is a relatively minor factor in computational performance as compared to the number of constraints.

Schrage (1986) outlines the total work per pivot using the LINDO software. Work is defined as the number of multiplications and divisions per pivot. The total work per pivot is:

$$\text{Work} = 2(1 + \text{density})(\text{number of rows}) + (\text{density})(\text{number of variables priced out})$$

Density is defined by Schrage to be the average number of non-zero constraint coefficients per column.

In both the proposed DMELP model and the Koehler and Erenguc (1990) model the density is equal to J+1 (ie: J represents the number of attributes for each observation) and the number of rows is equal to J+1. The maximum number of variables priced out are those variables in the objective function that have non-zero coefficients. LINDO does not perform multiplications for zero valued coefficients. The proposed model has M_1+M_2 (ie: M_1+M_2 represents the total number of observations) non-zero coefficients in the objective function. This is the same as the Koehler and Erenguc (1990) model. Therefore the work per pivot for the proposed DMELP model is similar to that of the Koehler and Erenguc (1990) model.

o

4.2 Details of Experimental Design

The experimental design, as developed by Joachimsthaler and Stam (1988) and also used by Koehler and Erenguc (1990), contains three factors. The factors are classification procedure, kurtosis and dispersion (variance-covariance homogeneity).

The classification procedures in this research are:

1) proposed optimal solution method
2) proposed Single Seed (SS) heuristic
3) proposed Three Seed (TS) heuristic
4) proposed N Seed (NS) heuristic
5) Glover, Keene and Duea (GKD) hybrid heuristic
6) linear discriminant analysis (LDA)

The reported results for the Koehler and Erenguc (1990) optimal solution method and heuristic solution method are also used in the analysis of the results. The Koehler and Erenguc (1990) experimentation using their methods is not repeated in this research.

The second experimental factor, kurtosis was varied by Joachimsthaler and Stam (1988) to approximately correspond to samples drawn from the uniform (kurtosis=-1), normal (kurtosis=0), logistic (kurtosis=+1) and Laplace (kurtosis=+3) distributed populations. The simulation of the observations was accomplished using the Fleishman (1978) and Vale and Maurelli (1983) methodology.

The third experimental factor, dispersion was held constant for group 1 and varied for group 2. This violates the assumption of LDA (Linear Discriminant Analysis) that requires the variance-covariance matrices for both groups to be equal. Observations are generated for three levels of dispersion such that the variance-covariance matrix of the group 2 observations is:

1) equal to that of group 1 (homogeneous variances)
2) two times that of group 1 (heterogeneous variances)
3) four times that of group 1 (heterogeneous variances).

Group 1 observations have a mean value of zero and a standard deviation of one. The mean values for group 2 observations were chosen by Joachimsthaler and Stam (1988) such that the group overlap was held constant. They used the Mahalanobis distance[1] (a general measure of the distance between the two groups) as a measure of group overlap.

---

[1] Mahalanobis distance, the distance between group 1 and group 2 is

$$= \left[\sum_{j=1}^{J} \sum_{k=1}^{J} COV_{jk} \, (MEAN_{j1}-MEAN_{j2})(MEAN_{k1}-MEAN_{k2}) \right]^{1/2}$$

where:
J is the number of attribute variables
$MEAN_{j1}$ is the mean for the j attribute variable of group 1
COV_{jk} is an element from the inverse of the within-groups covariance matrix

This results in twelve (4x3) kurtosis-dispersion factor combinations in the experimental design. These are referred to as cells. The parameter values chosen by Joachimsthaler and Stan (1988) and used in this research and the research of Koehler and Erenguc (1990) are summarized on table 4.1. The study involved a complete factorial design with 72 (6x4x3) factor combinations.

Consistent with Koehler and Erenguc (1990), the experiment has three independent attributes for each simulated observation. Also equal costs of misclassification for both groups, equal prior probabilities of group membership and equal sample sizes (ie: $M_1=M_2=50$) for both groups were used.

Except for the proposed optimal method, there were 20 replications of the experiment in each cell. The planned number of 20[2] replications was estimated using the power approach of Neter, Wasserman and Kutner (1985). This resulted in 240 (12 cells x 20 replications) problems being solved.

---

[2](Neter, Wasserman and Kutner page 604) Δ/σ was estimated directly to be $\geq$ 1.00 for each pair comparison to be examined. Controlling the probability of a Type I error, α (probability of rejecting the null hypothesis when true) at .1 and the probability of a Type II error, β (probability of accepting the null hypothesis when it is false) at .1 then the planning sample size for the ANOVA is 18.

TABLE 4.1

PARAMETER SPECIFICATION IN SIMULATION EXPERIMENTS

CELL	GROUP 1			GROUP 2		
	MEAN	VARIANCE	KURTOSIS	MEAN	VARIANCE	KURTOSIS
1	.0	1	-1	.5	1	-1
2	.0	1	0	.5	1	0
3	.0	1	+1	.5	1	+1
4	.0	1	+3	.5	1	+3
5	.0	1	-1	.6	2	-1
6	.0	1	0	.6	2	0
7	.0	1	+1	.6	2	+1
8	.0	1	+3	.6	2	+3
9	.0	1	-1	.8	4	-1
10	.0	1	0	.8	4	0
11	.0	1	+1	.8	4	+1
12	.0	1	+3	.8	4	+3

For the proposed optimal method the experiment was only completed for cells 1,2,3,4 and 10 with 20 replications in each these cells. This partial completion of the experimental design is due to the computational burden of computing the optimal solution. All 12 cells were completed for all of the other classification procedures.

The results for both the training (ie: estimation) sample and holdout (ie: validation) sample are analyzed. There are 100 observations in each training sample and 500 observations in each holdout sample.

CHAPTER 5

EXPERIMENTAL RESULTS

The discussion on results is divided between the results of the proposed optimal method in section 5.1 and the results of the proposed heuristics in section 5.2. Both of these sections are further divided into training sample results, holdout sample results, computational effort results and a summary. In section 5.3 there is an experiment on the effects of increasing the training sample size on the holdout misclassification rates.

5.1 Optimization Method

The focus of the analysis of the proposed optimal solution method is on that of computational burden. This is the focus since other researchers such as Koehler and Erenguc (1990) have already proposed optimal solution methods. Computational effort is defined in section 4.1.

5.1.1 Training Sample Results

The training sample misclassification rates for the proposed optimal solution method are shown on table 5.1. In addition the results of LDA, the GKD hybrid heuristic and the proposed SS, TS and NS heuristics are also included. These results will be discussed in section 5.2 in the comparison of the heuristics to the optimal solution. Details of results by experiment are shown in appendix B.

The mean optimal rate of misclassification of .245 is .003 less than the Koehler and Erenguc (1990) result of .248 for these five cells. The difference between the proposed optimal method and the Koehler and Erenguc optimal method (ie: proposed - KE) for cells 1, 2, 3, 4 and 10 are (.007), (.013), .001, (.008) and .011 respectively. The two results appear similar considering the standard deviation of the proposed results of .034. The standard deviations for the Koehler and Erenguc results are not available.

TABLE 5.1
TRAINING SAMPLE: MISCLASSIFICATION RATE RESULTS

C[1]	VH[2] 1	VH[2] 2	Group Mean 1	Group Mean 2	DK[3]	LDA	GKD	SS	TS	NS	OPT[4]	KE[5]
1	1	1	0	.5	-1	.326	.314	.276	.268	.265	.258	.265
					SD[6]	.038	.039	.042	.042	.039	.035	n/a
2	1	1	0	.5	0	.309	.279	.265	.259	.258	.245	.258
					SD	.044	.046	.040	.037	.037	.036	n/a
3	1	1	0	.5	1	.317	.298	.273	.262	.262	.254	.253
					SD	.043	.049	.038	.037	.036	.034	n/a
4	1	1	0	.5	3	.307	.289	.260	.248	.245	.235	.243
					SD	.042	.042	.035	.032	.033	.032	n/a
5	1	2	0	.6	-1	.290	.279	.238	.234	.234	n/a	.255
					SD	.041	.045	.036	.036	.036	n/a	n/a
6	1	2	0	.6	0	.314	.314	.272	.267	.264	n/a	.255
					SD	.041	.040	.030	.029	.027	n/a	n/a
7	1	2	0	.6	1	.316	.306	.275	.268	.264	n/a	.252
					SD	.045	.047	.041	.039	.039	n/a	n/a
8	1	2	0	.6	3	.297	.302	.254	.248	.246	n/a	.246
					SD	.058	.052	.041	.042	.041	n/a	n/a
9	1	4	0	.8	-1	.299	.300	.254	.246	.241	n/a	.219
					SD	.058	.050	.045	.043	.044	n/a	n/a
10	1	4	0	.8	0	.296	.303	.253	.246	.244	.231	.220
					SD	.049	.048	.041	.038	.036	.035	n/a
11	1	4	0	.8	1	.279	.284	.248	.239	.236	n/a	.220
					SD	.049	.056	.047	.040	.036	n/a	n/a
12	1	4	0	.8	3	.286	.280	.239	.227	.222	n/a	.220
					SD	.043	.045	.035	.034	.033	n/a	n/a
All Cells												
Mean						.303	.296	.259	.251	.248	n/a	.242
Standard Deviation						.046	.047	.040	.037	.037	n/a	n/a
Cells 1,2,3,4,10												
Mean						.311	.297	.265	.257	.254	.245	.248
Standard Deviation						.043	.045	.039	.037	.036	.034	n/a

1 Cell
2 Variance Heterogeneity
3 Distribution Kurtosis
4 Optimal Method
5 Koehler and Erenguc (1990) Optimal Method
6 Standard Deviation

5.1.2 Holdout Sample Results

The holdout sample results for the optimal solution method are shown on table 5.2. LDA, GKD hybrid heuristic and the proposed SS, TS and NS heuristic results are also included on table 5.2 for the later comparison of the heuristic results to the optimal solution results.

The holdout sample misclassification rate of .341 is .008 below the Koehler and Erenguc (1990) rate of .349 for the same five cells. The difference between the proposed optimal method and the Koehler and Erenguc optimal method (ie: proposed - KE) for cells 1, 2, 3, 4 and 10 are (.020), (.022), .007, (.015) and .010 respectively. The two results appear similar considering the standard deviation of the proposed results of .029. The standard deviations for the Koehler and Erenguc results are not available.

TABLE 5.2

HOLDOUT SAMPLE: MISCLASSIFICATION RATE RESULTS

C	VH 1	VH 2	Group Mean 1	Group Mean 2	DK	LDA	GKD	SS	TS	NS	OPT	KE
1	1	1	0	.5	-1	.347	.348	.352	.354	.357	.350	.370
					SD	.026	.027	.029	.035	.036	.029	n/a
2	1	1	0	.5	0	.320	.322	.342	.340	.342	.340	.362
					SD	.014	.016	.025	.025	.023	.032	n/a
3	1	1	0	.5	1	.340	.344	.354	.356	.357	.362	.355
					SD	.013	.013	.016	.017	.018	.032	n/a
4	1	1	0	.5	3	.320	.323	.334	.334	.332	.329	.344
					SD	.017	.021	.030	.025	.027	.021	n/a
5	1	2	0	.6	-1	.349	.352	.341	.341	.342	n/a	.353
					SD	.026	.025	.027	.027	.028	n/a	n/a
6	1	2	0	.6	0	.346	.349	.342	.346	.348	n/a	.354
					SD	.029	.032	.026	.024	.020	n/a	n/a
7	1	2	0	.6	1	.328	.331	.343	.334	.337	n/a	.351
					SD	.019	.021	.038	.027	.026	n/a	n/a
8	1	2	0	.6	3	.329	.337	.340	.341	.339	n/a	.349
					SD	.016	.017	.021	.020	.022	n/a	n/a
9	1	4	0	.8	-1	.322	.341	.319	.321	.327	n/a	.309
					SD	.032	.035	.037	.035	.034	n/a	n/a
10	1	4	0	.8	0	.317	.340	.318	.321	.327	.324	.314
					SD	.023	.027	.029	.031	.030	.030	n/a
11	1	4	0	.8	1	.314	.332	.319	.312	.314	n/a	.312
					SD	.025	.031	.029	.024	.025	n/a	n/a
12	1	4	0	.8	3	.305	.319	.318	.321	.320	n/a	.310
					SD	.028	.029	.044	.045	.045	n/a	n/a
All Cells												
Mean						.328	.337	.335	.335	.337	n/a	.340
Standard Deviation						.023	.025	.030	.029	.029	n/a	n/a
Cells 1,2,3,4,10												
Mean						.329	.335	.340	.341	.343	.341	.349
Standard Deviation						.019	.022	.026	.027	.027	.029	n/a

5.1.3 Computational Effort Results

As previously mentioned the focus of the optimal solution method analysis is on the issue of computational effort. The computational effort of the proposed optimal solution method is compared to the reported results on computational effort for the Koehler and Erenguc (1990) optimal method. Note, however, that the simulated observations in this research were generated under different hardware and software conditions than the Koehler and Erenguc research and therefore are not identical. In the Koehler and Erenguc (1990) article computational effort is measured as the number of major pivots, the number of minor pivots and the number of LP subsets solved to derive the optimal solution. Computational effort is described in section 4.1.

The first comparison relates to the number of subsets or LP's solved in the E/BAB procedure to determine the optimal solution. The proposed algorithm examines only 9,419 subsets whereas the Koehler and Erenguc (1990) method examines 90,115 subsets. This is an 89.5% decrease in the number of subsets examined. This is an important reduction. Details for the five cells examined in the experiment are shown on table 5.3.

TABLE 5.3

NUMBER OF LP SUBSETS SOLVED

C	VH 1	VH 2	Group Mean 1	Group Mean 2	DK	SS	TS	NS	OPTIMAL	MEMO: KE/OPT
1	1	1	0	.5	-1	28	80	459	8931	105590
2	1	1	0	.5	0	27	73	367	9372	99417
3	1	1	0	.5	1	27	27	363	11192	98253
4	1	1	0	.5	3	26	71	310	8660	85496
5	1	2	0	.6	-1	25	65	340	n/a	95133
6	1	2	0	.6	0	28	70	341	n/a	100284
7	1	2	0	.6	1	29	76	360	n/a	102035
8	1	2	0	.6	3	27	72	291	n/a	96984
9	1	4	0	.8	-1	27	70	345	n/a	56739
10	1	4	0	.8	0	27	74	309	8940	61821
11	1	4	0	.8	1	26	70	268	n/a	65089
12	1	4	0	.8	3	25	63	240	n/a	66900
All Cells										
Mean						27	72	333	n/a	86145
Cells 1,2,3,4,10										
Mean						27	65	362	9419	90115

where:
KE/OPT Koehler and Erenguc (1990) optimal method

This reduction in the number of LP subsets solved primarily relates to the fathoming within the methods. The proposed procedure of setting the greatest misclassification deviation to be one unit in the enumeration stage will scale all the other misclassification deviations and return the largest possible objective function value (ie: Z_L) from each subset examined in the minimization. However, with an upper bound of one on each misclassification deviation then the lower bound Z_L will never exceed the corresponding integer solution Z_R.

The second comparison relates to the number of pivots that are required to arrive at the optimal solution. As mentioned previously the total pivots are divided into major pivots and minor pivots, similar to the Koehler and Erenguc (1990) paper. Total pivots are shown on table 5.4, major pivots are shown on table 5.5 and minor pivots are shown on table 5.6. The proposed method performs only 41,972 major pivots which is 18,380 fewer pivots than the 60,352 major pivots of the Koehler and Erenguc (1990) method. Also the proposed method performs only 13,988 minor pivots which is 199,016 less minor pivots than the 213,004 minor pivots of the Koehler and Erenguc (1990) method. In total the proposed algorithm only performs 55,961 pivots compared to the 273,357 pivots of the Koehler and Erenguc (1990) method.

5.1.4 Summary

In summary the proposed optimal solution method requires notably less computational effort than the recently published method proposed by Koehler and Erenguc (1990).

TABLE 5.4

NUMBER OF TOTAL PIVOTS

C	VH 1	VH 2	Group Mean 1	Group Mean 2	DK	SS	TS	NS	OPTIMAL	MEMO: KE/OPT
1	1	1	0	.5	-1	360	711	4505	60771	300094
2	1	1	0	.5	0	356	686	4154	55992	292332
3	1	1	0	.5	1	370	727	4155	60914	293688
4	1	1	0	.5	3	360	714	4114	49351	265522
5	1	2	0	.6	-1	347	659	3957	n/a	283900
6	1	2	0	.6	0	358	697	4372	n/a	297653
7	1	2	0	.6	1	360	684	4169	n/a	304349
8	1	2	0	.6	3	354	694	4052	n/a	294280
9	1	4	0	.8	-1	354	682	4177	n/a	200455
10	1	4	0	.8	0	353	707	4071	52775	215148
11	1	4	0	.8	1	357	699	3793	n/a	223296
12	1	4	0	.8	3	362	706	3898	n/a	228561
All Cells Mean						358	698	4118	n/a	266607
Cells 1,2,3,4,10 Mean						360	709	4200	55961	273357

where:
KE/OPT Koehler and Erenguc (1990) optimal method

TABLE 5.5

NUMBER OF MAJOR PIVOTS

C	VH		Group Mean		DK					MEMO:
	1	2	1	2		SS	TS	NS	OPTIMAL	KE/OPT
1	1	1	0	.5	-1	356	702	4465	47652	63776
2	1	1	0	.5	0	352	675	4112	43108	62720
3	1	1	0	.5	1	365	716	4114	46876	63171
4	1	1	0	.5	3	355	702	4080	36486	58610
5	1	2	0	.6	-1	344	652	3929	n/a	62667
6	1	2	0	.6	0	354	688	4333	n/a	64437
7	1	2	0	.6	1	356	684	4130	n/a	65390
8	1	2	0	.6	3	351	684	4019	n/a	64119
9	1	4	0	.8	-1	350	674	4149	n/a	50613
10	1	4	0	.8	0	349	695	4038	35739	53485
11	1	4	0	.8	1	353	688	3762	n/a	54565
12	1	4	0	.8	3	357	694	3868	n/a	55221
All Cells										
Mean						354	688	4083	n/a	59898
Cells 1,2,3,4,10										
Mean						355	698	4162	41972	60352

where:
KE/OPT Koehler and Erenguc (1990) optimal method

TABLE 5.6

NUMBER OF MINOR PIVOTS

C	VH 1	VH 2	Group Mean 1	Group Mean 2	DK	SS	TS	NS	OPTIMAL	MEMO: KE/OPT
1	1	1	0	.5	-1	4	9	40	13119	236318
2	1	1	0	.5	0	4	11	42	12884	229612
3	1	1	0	.5	1	5	11	41	14038	230517
4	1	1	0	.5	3	5	12	34	12865	206912
5	1	2	0	.6	-1	3	7	28	n/a	221233
6	1	2	0	.6	0	4	9	39	n/a	233216
7	1	2	0	.6	1	4	10	39	n/a	238959
8	1	2	0	.6	3	3	10	33	n/a	230161
9	1	4	0	.8	-1	4	8	28	n/a	149842
10	1	4	0	.8	0	4	12	33	17036	161663
11	1	4	0	.8	1	4	11	31	n/a	168731
12	1	4	0	.8	3	5	12	30	n/a	173340
All Cells Mean						4	10	35	n/a	206709
Cells 1,2,3,4,10 Mean						4	11	38	13988	213004

where:
KE/OPT Koehler and Erenguc (1990) optimal method

5.2 Heuristic Methods

The analysis of the results of the heuristic methods focuses on both computational effort and "closeness" to the optimal solution. Computational effort is important in determining the value of a heuristic. The reason for using a heuristic is that a "good" non-optimal solution can be achieved with far less effort than that required by the optimal solution.

The discussion on the results is divided into training sample results, holdout sample results, computational effort results and a summary.

5.2.1 Training Sample Results

The training sample misclassification rates for LDA, GKD hybrid model, Single Seed (SS) heuristic, Three Seed (TS) heuristic and the N Seed (NS) heuristic are shown on table 5.1. The misclassification rates for the optimal method and heuristic proposed by Koehler and Erenguc (1990) are shown on table 2.9.

The mean training sample misclassification rates (for cells 1,2,3,4,10) in decreasing order are:
LDA=.311, GKD=.297, SS=.265, TS=.257, NS=.254, OPT=.245.
It should be noted that the NS and TS heuristics produce

solutions that are only .009 and .012 respectively greater than the optimal solution. These solutions appear "close" to the optimal solution. Whereas, the SS, GKD and LDA methods produce solutions that are .020, .052 and .066 respectively greater than the optimal solution. These solutions do not appear "close" to the optimal solution.

In order to attempt to examine the "closeness" of the NS heuristic to the optimal solution an ANOVA comparison was performed for cells 1, 2, 3 and 4. The results are shown on table 5.7. Also the frequency was calculated showing the number of misclassifications by which the NS heuristic exceeds the optimal. These results are shown on table 5.8.

The hypothesis tested is :

H_0: mean of NS - mean of optimal = 0

H_1: mean of NS - mean of optimal $\neq$ 0

This ANOVA comparison is a two factor experiment with repeated measures on the procedures factor.

A variance stabilizing transformation was used prior to conducting the ANOVA in order to make statistical comparisons among the procedures. The transformation that has been suggested in Goldstein and Dillon (1978) and applied here is:

2(arcsin(sqrt(misclassification rate)))

TABLE 5.7

ANALYSIS OF VARIANCE
OPTIMAL vs N SEED HEURISTIC
TRAINING SAMPLE - CELLS 1,2,3,4
(Two Factors with Repeated Measures on Procedures)

Source	Sum of Squares	Degrees of Freedom	Mean Square	F	Tail Prob.
Kurtosis	.056	3	.019	1.44	.2391
ERROR	.990	76	.013		
Procedure	.019	1	.019	86.77	.0000
Proc x Kurt	.001	3	.000	1.83	.1488
ERROR	.017	76	.000		

This transformation has also been used by Bajgier and Hill (1982), Joachimsthaler and Stam (1988) and Koehler and Erenguc (1990) to stabilize the variances of the rates of misclassification. The skewness of the transformed misclassification rates was tested and determined to be not significantly different from zero. This supports the hypothesis that the transformed misclassification rates are normally distributed.

The ANOVA comparing the proposed optimal method to the N seed heuristic (table 5.7) shows that at the 5% level of significance there are no two-way interaction effects. The main effect of procedure is significant. Thus, there is a significant difference between the means of the proposed optimal method and the N seed heuristic in cells 1, 2, 3 and 4. Therefore even though, the mean misclassification rates appear "close" they are significantly different.

As a further analysis of "closeness" the number of misclassifications by which the NS heuristic exceeds the optimal solution is shown on table 5.8. In 35% of the 100 experiments, the NS heuristic returned the optimal solution. As a comparison, the Koehler and Erenguc (1990) heuristic returned the optimal solution 32% of the time for the same 5 cells. The NS heuristic never returned a solution that

TABLE 5.8

FREQUENCY OF MISCLASSIFICATIONS OF NS OVER OPTIMAL TRAINING SAMPLE

	NUMBER OVER OPTIMAL					
CELL	0	1	2	3	4	TOTAL
1	11	5	3	1	0	20
2	5	7	6	1	1	20
3	8	8	4	0	0	20
4	7	6	7	0	0	20
10	4	8	7	0	1	20
TOTAL	35	34	20	2	2	100

was more than 4 misclassifications greater than the optimal solution. The NS heuristic generates solutions that are "close" to the optimal.

In order to examine the "closeness" of LDA to the SS heuristic, the SS heuristic to the TS heuristic and the TS heuristic to the NS heuristic a series of ANOVA comparisons for all 12 cells of the experiment were performed. These ANOVA comparisons are three factor experiments with repeated measures on the procedures factor. Again, the variance stabilizing transformation was employed.

The hypothesis tested is :

H_0: mean of heuristic 1 - mean of heuristic 2 = 0

H_1: mean of heuristic 1 - mean of heuristic 2 $\neq$ 0

The ANOVA results for LDA vs SS show that there are no significant three-way interaction effects. The ANOVA results are shown on table 5.9. Also, at the 5% level of significance, there are no two-way interaction effects. The main effects of procedure and variance are significant. Thus, there is a significant difference between the means of the LDA method and the SS heuristic. In every cell on table 5.1 the mean misclassification rate for SS is less than that for LDA.

TABLE 5.9

ANALYSIS OF VARIANCE
LDA vs SINGLE SEED HEURISTIC
TRAINING SAMPLE
(Three Factors with Repeated Measures on Procedures)

Source	Sum of Squares	Degrees of Freedom	Mean Square	F	Tail Prob.
Variance	.204	2	.102	6.05	.0028
Kurtosis	.050	3	.017	.98	.4007
Var x Kurt	.156	6	.026	1.54	.1656
ERROR	3.846	228	.017		
Procedure	1.147	1	1.147	614.65	.0000
Proc x Var	.002	2	.001	.51	.5992
Proc x Kurt	.010	3	.003	1.79	.1501
Proc x Var x Kurt	.004	6	.001	.36	.9016
ERROR	.426	228	.002		

The ANOVA comparison for the SS and TS heuristics is shown on table 5.10. Again, there are no significant three-way or two-way interaction effects. The main effect of procedure is significant. Also, in every cell on table 5.1 the TS heuristic has a lower mean misclassification rate than the SS heuristic.

The final ANOVA comparison showed that there is no significant difference between the NS heuristic and the TS heuristic. The TS heuristic misclassification rate is only .008 greater than the NS rate of .251.

Since the TS heuristic generates solutions that are not significantly different from the NS heuristic, an analysis of the number of misclassifications that are returned by the TS heuristic compared to the optimal is shown on table 5.11. In 32 of the 100 experiments, the TS heuristic returned the optimal solution. The results have deteriorated slightly from the NS heuristic.

TABLE 5.10

ANALYSIS OF VARIANCE
SS vs TS HEURISTIC
TRAINING SAMPLE
(Three Factors with Repeated Measures on Procedures)

Source	Sum of Squares	Degrees of Freedom	Mean Square	F	Tail Prob.
Variance	.170	2	.085	5.47	.0048
Kurtosis	.096	3	.032	2.06	.1057
Var x Kurt	.158	6	.026	1.70	.1223
ERROR	3.543	228	.015		
Procedure	.039	1	.039	87.91	.0000
Proc x Var	.002	2	.001	2.42	.0909
Proc x Kurt	.002	3	.001	1.34	.2633
Proc x Var x Kurt	.001	6	.000	.21	.9719
ERROR	.101	228	.000		

TABLE 5.11

FREQUENCY OF MISCLASSIFICATIONS OF TS OVER OPTIMAL TRAINING SAMPLE

CELL	NUMBER OVER OPTIMAL 0	1	2	3	4	5	6	TOTAL
1	9	5	4	2	0	0	0	20
2	5	7	5	1	2	0	0	20
3	8	8	4	0	0	0	0	20
4	6	4	9	0	1	0	0	20
10	4	8	5	1	1	0	1	20
TOTAL	32	32	27	4	4	0	1	100

As previously mentioned these results are compared to the BPMM heuristic results and the PMM optimal solution results reported in the Koehler and Erenguc (1990) paper. Note, however that the simulated problems in this research were generated under different hardware and software conditions and so the problems are not identical to those in Koehler and Erenguc (1990).

The results for selected methods in decreasing misclassification rate order are:

SS=.265, TS=.257, BPMM=.256, NS=.254, PMM=.248, OPT=.245

The BPMM result is .008 greater than the PMM optimal and the NS heuristic result is .009 greater than the optimal solution. The BPMM and NS heuristic results are approximately equal distances from their respective optimal solutions.

5.2.2 Holdout Sample Results

The holdout sample misclassification results are shown on table 5.2. The selected mean misclassification rates (all cells) in decreasing order are: GKD=.337, NS=.337, SS=.335, TS=.335, LDA=.328. The Koehler and Erenguc (1990) holdout sample results shown in table 2.9 are: BPMM=.342, PMM(optimal)=.340 and LDA .345.

LDA has the highest mean training sample misclassification rate (.055 greater than the NS heuristic of .248) and the lowest mean holdout sample misclassification rate (.007 lower than the TS heuristic of .335). The TS heuristic generates holdout sample misclassification rates lower than the LDA method in only 2 of the twelve cells.

In section 5.3, a further experiment is performed to determine the effect of increasing the training sample size on the mean holdout misclassification rates of LDA, SS, TS and NS.

5.2.3 Computational Effort Results

As stated previously, computational effort is important in determining the value of a heuristic. Computational effort as measured by the number of LP subsets solved, the number of total pivots, the number of major pivots and the number of minor pivots are shown on tables 5.3, 5.4, 5.5 and 5.6. These measures of computational effort are the same as those recorded in the Koehler and Erenguc (1990) paper. All three of the proposed heuristics require notably less computational effort than that required for the optimal solution. The NS heuristic requires 96% fewer LP subsets to be solved than the optimal. Similar reductions of 92%, 90% and 99% are experienced for total pivots, major pivots and minor pivots respectively. Greater reductions are provided by the TS heuristic and the SS heuristic. Recall that the TS and NS heuristic generate training sample solutions that are "close" to the optimal.

Koehler and Erenguc (1990) reported their BPMM heuristic results for only the first cell of the experiment. Their heuristic required the solution of 72 LP's with total pivots of 6982 divided between major pivots of 6474 and minor pivots of 508. The TS heuristic for cell 1, required the solution of 80 LP's which is about the same as BPMM but only required 702 major pivots which is a substantial reduction (ie:89%) and only 9 minor pivots (ie: 98% reduction). The NS heuristic, for cell 1, required the solution of 459 LP's to be solved, which is a substantial increase over BPMM, but required only 4465 major pivots (31% less) and only 40 minor pivots (94% less).

5.2.4 Summary

In summary, the experiment has shown that the proposed TS and NS heuristics generate training sample solutions that are "close" to the optimal solution and that these solutions require less computational effort than the optimal solution. Also the TS and NS heuristics require fewer pivots than the BPMM heuristic of Koehler and Erenguc (1990). Also the TS heuristic in turn requires less computational effort than either the NS heuristic or the BPMM heuristic of Koehler and Erenguc (1990).

The SS heuristic, GKD hybrid model and LDA generate mean training sample misclassification rate solutions that vary significantly from the optimal.

5.3 Increasing Sample Size Experiment

An experiment was performed to determine the effect on the holdout misclassification rate of increasing the training sample size. Training sample sizes (M_1+M_2) tested were 100, 200 and 500. The corresponding holdout sample sizes were 500, 1000 and 2500.

Twenty experiments were performed for each of two cells of the original experiment for a total of 40 experiments. The two cells chosen are 2 and 9 which represent the case where all the LDA assumptions are met and the case where the kurtosis is -1 and the variance-covariance matrix of group 2 is four times that of group 1. The holdout misclassification rates and computational effort results are shown on table 5.12 (details in appendix B3).

As an observation we note that as the training sample size increases in both cells tested, the performance

TABLE 5.12
HOLDOUT MISCLASSIFICATION RATES
FOR INCREASING TRAINING SAMPLE SIZES

Size	LDA Mean SD	GKD Mean SD	GKD Diff[1]	S Seed Mean SD	S Seed Diff	T Seed Mean SD	T Seed Diff	N Seed Mean SD	N Seed Diff
CELL 2									
100	.320	.322	+.002	.342	+.022	.340	+.020	.342	+.022
	.014	.016		.025		.025		.023	
200	.333	.332	-.001	.343	+.010	.344	+.011	.344	+.011
	.016	.019		.022		.026		.026	
500	.336	.335	-.001	.337	+.001	.340	+.004	.344	+.008
	.006	.006		.006		.011		.011	
CELL 9									
100	.322	.300	-.022	.319	-.003	.321	-.001	.327	+.005
	.032	.050		.037		.035		.034	
200	.323	.338	+.015	.305	-.018	.309	-.014	.312	-.011
	.019	.013		.015		.015		.017	
500	.316	.331	+.015	.296	-.020	.295	-.021	.297	-.019
	.015	.014		.016		.016		.017	

Note: Computational Effort

Size	GKD Piv	Single Seed Maj	Single Seed Tot	Single Seed Sub	Three Seed Maj	Three Seed Tot	Three Seed Sub	N Seed Maj	N Seed Tot	N Seed Sub
CELL 2										
100	156	352	356	27	675	686	80	4112	4154	367
200	315	803	814	60	1605	1631	164	17245	17363	1054
500	782	2345	2376	159	4914	4985	443	103656	104148	4108
CELL 9										
100	157	350	354	27	674	682	70	4149	4177	345
200	326	797	806	55	1530	1550	144	16259	16350	1127
500	801	2200	2218	137	4137	4182	353	89646	89972	4572

Memo: Koehler and Erenguc (1990) Holdout Results[2]

	Size	LDA	BPMM	Diff
CELL 2	100	.378	.398	+.020
	200	.376	.368	+.008
	500	.382	.378	+.004
CELL 9	100	.314	.296	-.018
	200	.354	.306	-.048
	500	.370	.312	-.058

[1] Difference: Heuristic minus LDA
[2] KE performed 1 repetition per case whereas we performed 20 repetitions per case (however, there were only 5 repetitions per case with the 500 training sample size).

of the proposed LP heuristics improve compared to LDA. This is similar to the Koehler and Erenguc (1990) results, also shown on table 5.12. In addition the computational effort of the larger training sample sizes is not burdensome.

CHAPTER 6

QUADRATIC TRANSFORMATION METHOD

In this section a new heuristic solution method for the two group classification problem using a quadratic transformation of the observation data is presented. This method is also developed in Banks and Abad (1991a). QDA is used to compare the results of the quadratic transformation method when applied to the Glover, Keene and Duea (GKD) (1988) hybrid model 4, the SS heuristic, the TS heuristic and the NS heuristic.

The motivation underlying this research is to develop an LP method that is competitive not only with LDA but also with QDA. Previous research has concentrated on LDA. The method described in the next section is based on a quadratic transformation of the observation data. The number of observation attributes following the transformation are reduced using a variable selection criterion.

6.1 Description of the Quadratic Transformation Method

In this section the quadratic transformation and variable selection method is described.

Step 1: Transform the Observations

In the quadratic transformation method, first we transform the observation data. In the experiment, three attributes (J=3) for each observation, were used. This results after the transformation in nine attributes for each observation. The attributes before the transformation are A_{i1}, A_{i2} and A_{i3}. After the transformation the attributes are:

A_{i1}	A_{i2}	A_{i3}
$A_{i1}xA_{i1}$	$A_{i2}xA_{i2}$	$A_{i3}xA_{i3}$
$A_{i1}xA_{i2}$	$A_{i1}xA_{i3}$	$A_{i2}xA_{i3}$

These are the same terms as outlined in Smith (1947) in his description of QDA.

The unknown W_j variables in the GKD hybrid model formulation result, for J=3 in 3 unknown weight variables. The quadratic transformation of the GKD hybrid model formulation results, for J=3 in 9 unknown weight variables. In general, for the J attribute case this transformation results in 2J+(J!/(2!(J-2)!) attributes. For large J, this is a significant increase in the number of variables and may effect the computational effort and will be discussed later.

Step 2: Variable Selection

In order to reduce the number of attributes for each observation, we have applied a variable selection method similar to the Nath and Jones (1988) method[1] to distinguish between significant and nonsignificant attributes.

The variable selection procedure in the simulation experiment is as follows:

1. Solve several independent simulated problems using all of the transformed attributes. In the experiment, 20 problems in each cell were solved.

2. Record the W_j and C terms for each solution.

---

[1] The Nath and Jones method is developed for the case of a single sample of M_1+M_2 observations each having J attributes. They describe their method as a "jackknife" approach. Their procedure solves M_1+M_2 LP formulations of the classification problem. Each LP formulation utilizes only M_1+M_2-1 of the observations (ie: each observation in turn is deleted from the sample before solving the LP formulation). The resultant W_j's from each of the M_1+M_2 LP solutions are then used to calculate the sample mean and sample standard deviation of each W_j. The t-distribution is then used to measure the significance of each W_j term. Nath and Jones rank order the significance levels to determine the relative importance of each attribute. Attributes with significant t values are retained while those with nonsignificant t values are discarded to simplify the LP model.

3. Calculate the sample mean, sample standard deviation and t-statistic for each W_j term.

4. Attributes with significant t values are retained while those with nonsignificant t values are discarded to simplify the model. In the experiment a 5% level of significance was used.

Nath and Jones (1988) recommend a "jackknife" approach to variable selection. The advantage of the "jackknife" approach is that it can be used with just one sample. In practical implementation of our transformation procedure, one would need to use the "jackknife" approach to select variables.

Step 3: Solve

The problems are now solved with the reduced set of transformed attributes.

6.2 Experimental Design

The proposed quadratic transformation procedure is compared to quadratic discriminant analysis (QDA) using the same experimental design as outlined in section 4.2. Transformed LP solutions are developed using the GKD hybrid model, the SS heuristic, the TS heuristic and the NS heuristic. The experimental design, however is applied on a limited basis.

This experiment consists of the same three factors with 5 procedures, 4 kurtosis levels but only 1 dispersion level. The dispersion level is set such that the variance-covariance matrix for one group is 4 times greater than that for the other group. This violates the assumption of LDA but not QDA. This experiment examines cells 9, 10, 11 and 12 of the section 4.2 experimental design.

There were 3 attributes in each experiment with a training sample size of 100 observations from each group. Also equal costs of misclassification for each group and equal prior probabilities of group membership were used. Within this experiment all parameters are identical to those of chapter 4.

There were 20 repetitions of the experiment in each cell of the experimental design. This resulted in 80 problems being generated with each problem being solved by each of the five classification procedures.

A holdout sample analysis is also performed. There are 1000 observations in each holdout sample. The holdout observation samples were generated independent of the training samples.

6.3 Results

The discussion of the results are divided into training sample results, holdout sample results and computational effort results.

Observations which lie on the hyperplane dividing the two groups are counted as one-half of a misclassification. Previously, these observations were counted as correctly classified to be consistent with the Koehler and Erenguc (1990) research. Rubin (1990) counts hyperplane observations as one-half of a misclassification. This will result in a negative bias when comparing these results to the Koehler and Erenguc (1990) results or when comparing to the results in chapter 5.

6.3.1 Training Sample Results

The training sample misclassification rate for each of the four factor combinations in our experiment is shown in table 6.1. The top portion of the table shows the results for all nine attributes without the application of the variable selection method. The bottom portion of the table shows the results following the application of the variable selection method (ie: only the significant variables).

TABLE 6.1

TRAINING SAMPLE MISCLASSIFICATION RATE RESULTS
(Training sample size $M_1+M_2=200$)

ALL NINE ATTRIBUTES

Cell	VH[1] 1	VH[1] 2	Mean[2] 1	Mean[2] 2	Dist Kurt[3]	QDA	GKD	SS	TS	NS
9	1	4	0	.8	-1	.130	.127	.108	.097	.095
					SD[4]	.017	.022	.022	.016	.014
10	1	4	0	.8	0	.178	.190	.162	.155	.144
					SD	.026	.026	.026	.025	.019
11	1	4	0	.8	1	.200	.204	.182	.172	.162
					SD	.023	.025	.024	.025	.021
12	1	4	0	.8	3	.214	.202	.184	.176	.167
					SD	.029	.031	.033	.033	.026
Mean						.181	.181	.159	.150	.142

ONLY THE SIGNIFICANT ATTRIBUTES FOR LP METHODS

Cell	VH[1] 1	VH[1] 2	Mean[2] 1	Mean[2] 2	Dist Kurt[3]	QDA	GKD	SS	TS	NS
9	1	4	0	.8	-1	.130	.141	.112	.109	.108
					SD	.017	.020	.019	.016	.016
10	1	4	0	.8	0	.178	.185	.168	.157	.153
					SD	.026	.027	.028	.023	.020
11	1	4	0	.8	1	.200	.206	.196	.187	.180
					SD	.023	.024	.026	.026	.027
12	1	4	0	.8	3	.214	.203	.186	.181	.174
					SD	.029	.031	.031	.028	.022
Mean						.181	.184	.166	.159	.154

1 Variance Heterogeneity
2 Group Mean
3 Distribution Kurtosis
4 Standard Deviation

The removal of the insignificant attribute variables only caused a minor deterioration in the training classification results. The misclassification rate for GKD, SS, TS and NS increased by only .003, .007, .009 and .008 respectively.

A paired t-test was performed comparing the classification results (significant attributes only) for QDA to the TS heuristic. The mean of the TS heuristic of .166 is .015 lower than the QDA rate of .181. The paired t-test[2] showed that in each of the four cells of the experiment, that the TS heuristic generated solutions that were significantly different from QDA. There is a further slight improvement of the NS heuristic over the TS heuristic.

The significance of the W_j terms is shown on table 6.2. In cell 9, there were 3 significant attributes, in cell 11 there were 5 significant attributes and in cells 10 and 12 there were 6 significant attributes. It should be noted that the W_7, W_8 and W_9 terms were never significant in the experiment. There was a maximum of 2J significant

---

[2]Two-tailed test with 5% level of significance and 20-1 degrees of freedom. Critical t value is $t_{.025,19}=2.093$. The calculated t values for cells 9, 10, 11 and 12 are 5.43, 6.28, 3.69 and 8.38 respectively. Thus the difference is significant in all cells. The results were transformed using 2(arcsin(sqrt(misclassification rate))) for the t-test.

TABLE 6.2

SIGNIFICANCE TEST FOR W_j TERMS

		Cell 9 t		Cell 10 t		Cell 11 t		Cell 12 t	
W_1	A_{i1}	2.011	N	2.787	S	1.681	N	3.433	S
W_2	A_{i2}	1.016	N	3.582	S	3.332	S	4.916	S
W_3	A_{i3}	.457	N	2.247	S	2.339	S	3.861	S
W_4	$A_{i1}xA_{i1}$	17.794	S	7.133	S	6.509	S	6.188	S
W_5	$A_{i2}xA_{i2}$	14.631	S	7.850	S	6.428	S	6.014	S
W_6	$A_{i3}xA_{i3}$	18.109	S	9.622	S	6.519	S	5.585	S
W_7	$A_{i1}xA_{i2}$	1.701	N	.077	N	-.238	N	.837	N
W_8	$A_{i1}xA_{i3}$	-.857	N	2.019	N	1.174	N	-1.912	N
W_9	$A_{i2}xA_{i3}$	-.660	N	.424	N	.651	N	-.216	N

Notes:

1. S means significant difference ie:reject H_0
   N means no significant difference ie: accept H_0
   The critical t-value for 5% level of significance and 20-1 degrees of freedom is ±2.093 (two-tailed)
2. In our procedure, C was treated not as a constant but as a discriminant coefficient. The results are very similar when we scale the W_j's by dividing by C.

attributes after the variable selection.

6.3.2 Holdout Sample Results

The holdout misclassification rate results are shown on table 6.3. Similar to the training sample results table, the top portion of the table shows the results for all nine attributes and the bottom portion shows the results for the significant attributes only. There were no observations lying on the hyperplane when determining the holdout results.

The removal of the insignificant attributes resulted in an improvement in the holdout results. The holdout results for GKD, SS, TS and NS improved by .008, .013, .013 and .009 respectively.

The mean holdout results (for significant attributes) for GKD, SS, TS and NS exceed QDA by .010, .013, .013 and .011 respectively. However, in cell 9 the SS, TS and NS heuristics generated holdout results which were less

TABLE 6.3
HOLDOUT SAMPLE MISCLASSIFICATION RATE RESULTS

ALL NINE ATTRIBUTES

Cell	VH 1 2	Mean 1 2	Dist Kurt	QDA	GKD	SS	TS	NS
9	1 4	0 .8	-1	.154	.170	.161	.160	.160
			SD	.013	.023	.027	.019	.018
10	1 4	0 .8	0	.188	.212	.225	.220	.217
			SD	.011	.016	.029	.025	.017
11	1 4	0 .8	1	.207	.229	.239	.242	.235
			SD	.014	.019	.028	.027	.024
12	1 4	0 .8	3	.223	.235	.250	.252	.241
			SD	.012	.018	.030	.029	.026
Mean				.193	.211	.219	.219	.213

ONLY THE SIGNIFICANT ATTRIBUTES FOR LP METHODS

Cell	VH 1 2	Mean 1 2	Dist Kurt	QDA	GKD	SS	TS	NS
9	1 4	0 .8	-1	.154	.159	.143	.141	.141
			SD	.013	.019	.018	.017	.016
10	1 4	0 .8	0	.188	.206	.215	.208	.205
			SD	.011	.022	.025	.022	.020
11	1 4	0 .8	1	.207	.222	.226	.235	.230
			SD	.014	.017	.015	.020	.018
12	1 4	0 .8	3	.223	.225	.239	.239	.238
			SD	.012	.022	.031	.033	.031
Mean				.193	.203	.206	.206	.204

than the QDA results. A paired t-test[3] for cell 9 indicates that the TS heuristic generates holdout sample solutions significantly different from QDA. Also, in cells 10, 11 and 12, the paired t-tests indicate that QDA generates holdout sample solutions that are significantly different from the TS heuristic.

In Koehler and Erenguc (1990) and in section 5.3 an experiment is performed to determine the effects on holdout sample performance for a variety of training sample sizes. Both Koehler and Erenguc and section 5.3 show that as the sample size increases, the gap between training and holdout sample results narrows for LP based procedures. This is not surprising since as sample size tends to infinity, training and holdout sample performance should be similar. They also show that as training sample size increases the results of linear LP methods improve when assumptions of LDA are violated compared to linear discriminant analysis. Similar results are expected in the present study as sample size increases.

[3]Two-tailed test with 5% level of significance and 20-1 degrees of freedom. The critical t value is $t_{.025,19}$=2.093. The calculated t values for cells 9, 10, 11 and 12 are 2.92, -4.99, -7.32 and -2.99. The difference is TS-QDA.

6.3.3 Computational Effort Results

Computational effort is an important factor in determining the value of a heuristic. Computational effort as measured by the number of major pivots , number of minor pivots, number of total pivots and the number of LP subsets solved is shown on table 6.4. These are the same measures of computational effort as reported in Koehler and Erenguc (1990) and used in chapter 5.

As previously outlined in section 4.1 there are three factors that effect the computational performance of a linear program. They are 1) the number of functional constraints, 2) the number of variables and 3) the density of the table of constraint coefficients. Density is defined as the proportion of the constraint coefficients that are non-zero. The number of constraints and the density effect the computational effort per pivot whereas the number of variables effects the number of pivots.

The GKD model as applied to the significant transformed attributes has increased the number of variables from $2(M_1+M_2)+J+1$ to a maximum of $2(M_1+M_2)+2J+1$ (ie: from 204 to 207). It has the same number of functional constraints and has the same density as the GKD model before the use of the significant transformed attributes. Therefore the

TABLE 6.4

COMPUTATIONAL EFFORT RESULTS
FOR SIGNIFICANT ATTRIBUTES EXPERIMENT

GKD	Single Seed			Three Seed			N Seed		
Piv	Maj[1]	Tot[2]	Sub[3]	Maj	Tot	Sub	Maj	Tot	Sub
CELL 9									
295	635	637	23	1213	1217	52	8003	8012	198
CELL 10									
535	1170	1174	34	2356	2367	95	19113	19216	919
CELL 11									
502	1114	1121	39	2211	2231	109	17723	17830	1029
CELL 12									
589	1255	1264	38	2456	2480	108	18885	19022	1024
MEAN:									
480	1043	1049	34	2059	2074	91	15931	16020	792

[1] Major Pivots
[2] Total Pivots
[3] LP Subsets solved

amount of effort per pivot has not changed significantly. The computational effort of 480 pivots is not a burden.

The DMELP model (table 3.5) has before the application of only the significant transformed attributes, J+1 constraints, $2(M_1+M_2)$ variables and M_1+M_2 simple upper bounds. After the application of the significant transformed attributes the number of variables and number of simple upper bounds has not changed. However, there is an increase in the number of constraints, to a maximum in our experiment of 2J+1. The effort per pivot has increased. However, the computational effort of 2074 pivots for the TS heuristic is not burdensome.

6.4 Summary

The conclusions that can be drawn from this research are:

1) that the current set of LP classification models in the literature, such as the GKD model and TS heuristic can be easily transformed to be competitive with QDA when the variance-covariance matrices of the groups are unequal.

2) that the transformation from linear to quadratic significantly improves the solution.

3) that further research into transformed linear programming models is necessary.

4) that not only LDA but also QDA should be used as a "benchmark" in testing new LP formulations.

CHAPTER 7

CONCLUSIONS

The conclusions are divided into contributions in section 7.1 and future research in section 7.2.

7.1 Contributions

The first contribution of this thesis is the proposal of a new mixed integer programming model of the two group classification problem. A branch-and-bound procedure is designed to solve the mixed integer model. The contribution is that the proposed method is efficient in terms of computational effort when compared to the recent published research of Koehler and Erenguc (1990).

The second contribution is the proposal of new heuristics, called Three Seed and N Seed, that generate solutions that are "close" to the optimal solution. The Three Seed and N Seed heuristics require significantly less computational effort than the optimal solution and performs significantly fewer pivots than the recent BPMM heuristic of Koehler and Erenguc (1990). These heuristics generate training sample solution that are better than the GKD hybrid model.

The Three Seed heuristic generates solutions that are not significantly different from the N Seed heuristic. The advantage of the TS heuristic is the significant reduction in computational effort compared to the N Seed heuristic.

The next contribution of this research is the systematic examination of the effect that the training sample size has on the holdout sample results. The conclusion for this experiment is that as the size of the training sample increases then the performance for the linear programming methods tested improve compared to linear discriminant analysis. Linear programming methods are nonparametric.

The final contribution is the proposal and testing of a new quadratic transformation method that may after further development result in linear programming classification methods that are not only competitive with linear discriminant analysis but also competitive with quadratic discriminant analysis.

7.2 Future Research

The thesis has presented optimization and heuristic methods that have been tested using simulated continuous data. In the "real world" many of the two group classification problems use binary and integer data as well as continuous data. Further study should be on the effects of binary and integer data on the performance of the proposed methods with a comparison to linear discriminant analysis. Studies should include both simulated observation studies and studies using "real world" data.

The quadratic transformation method proposed in the thesis, looks promising, but requires further development.

Convex hull research should be investigated in order to develop heuristics, similar to the N Seed heuristic. Instead of using the GKD model to generate the seeds, possibly a convex hull algorithm could be used to generate the seeds.

APPENDIX A

FORTRAN PROGRAM ROUTINES

The following fortran programs have been written in support of the thesis:

1. Generator for the Sample Observations

2. Linear Program Generator for the GKD Hybrid Model

3. Linear Progran Generator for the DMELP Model

4. LINDO USER'S Subroutine for the Optimal Solution

5. LINDO USER'S Subroutine for the Single Seed and Three Seed Heuristics

6. LINDO USER'S Subroutine for the N Seed Heuristic

Included in this appendix is the LINDO USER'S Subroutine for the optimal solution.

OPTIMAL SOLUTION METHOD

LINDO USER'S FORTRAN SUBROUTINE

```
*
*
*     LINDO USER'S SUBROUTINE FOR OPTIMAL SOLUTION
*     DMELP MODEL FOR 3 ATTRIBUTES
*
      subroutine user(infile)
*
******************************************************
* NOTATION
* a(i,j)          =training value for obs i of attribute j
* adjust(subset,row)
*                 =array of partial solution parameters
*          row=1=no. of rows in the adjust array
*          row=2=farthest observation in enum stage
*          row=3=no. of next obs on which to branch
*          row=4,6,8,...=previously fixed obs
*          row=5,7,9,...=corr/miscl obs
* ah(i,j)         =holdout value for obs i of attribute j
* c(subset)       =cutoff for particular subset
* cost            =alpha in thesis
* d               =filename for training sample file
* descen(subset)       =indicator if subset has descendants
* dh              =filename for holdout sample file
* iterat          =cumulative number of total pivots
* m1,m2           =number of observations from each group
* mdlit           =LINDO internal variable for pivots
* minit           =cumulative number of minor pivots
* minor           =number of minor pivots in single LP
* optsub          =number for optimal subset
* predec          =number for predecessor subset
* rhs2..rhs5      =right-hand side for rows 2..5
* size            =expected num of LP subsets to be solved
* slpred(subset)       =filename for partial LINDO solution of
*                  predecessor
* subset          =subset currently being solved in B-A-B
* w(subset,j)     =weight for subset & attribute j
* zerone(subset)       =indicator if soln is integer (0-1)
* zr(subset)      =integer solution for the subset
* zr1             =integer solution for group 1
* zr2             =integer solution for group 2
* zl(subset)      =value of objective function for subset
* zu              =current upper bound on BAB
* zuold           =previous upper bound on BAB
* zzlp            =DMELP model stored in DB format
* zzzsoln         =output file of progress toward solution
******************************************************
```

```
*
* to set local conditions for this problem
      integer size,m1,m2
      real cost
      parameter(size=35000)
      parameter(m1=50,m2=50,cost=1.000)
*
* to set declarations
      common /VUBLP/ M, N
      logical TRUBLE
      character*14 flname,finame,slpred(size)
      character*80 CH
      character*70 d,dh,zzlp,zzzsoln
      integer minor, minit
      integer k, nosub(m1+m2), no
      integer empty(size)
      integer i,subset,iterat,zr1,zr2,zerone(size)
      integer adjust(size,125),observ,descen(size)
      integer predec,farthe,branch
      integer zltrun,zrtrun,optsub
      real oldyv(m1+m2+m1+m2),newyv(m1+m2+m1+m2),obj
      real a(m1+m2,3),zu,zuold,zl(size),wwwwwc(15)
      real ah(5*(m1+m2),3)
      real rhs2,rhs3,rhs4,rhs5
      real w(size,6),c(size)
      real zr(size),t(100),hight,lowzl
      real nrhs2,nrhs3,nrhs4,nrhs5
      real sum1(3),sum2(3)
      real comb(100)
      real miscl1, miscl2, tdev
      common dummy(14), mdlit
      TRUBLE= .false.
*
* format statements
 101  format(4x,f10.5,11x,f10.5,11x,f10.5)
 130  format(a)
 131  format(i1.1)
 200  format(a,i6.6)
 201  format(a,f17.6)
 202  format(a,f10.6,a,f10.6,a,f10.6)
 302  format(a,f10.6)
 400  format(10x,E13.8)
```

```
***********************************************
* to start new example                        *
***********************************************
*
 700  continue
      write(*,200)'enter 1 to continue ; 0 to quit'
      read(*,131,end=20)no
      if(no .ne. 1)then
        go to 750
      endif
*
* input names of files to be used
      write(*,200)'Name of training file (ie:d?)?'
      read(*,200,end=20)d
      write(*,200)'Name of holdout file (ie:d?)?'
      read(*,200,end=20)dh
      write(*,200)'Name of DMELP file (ie:lz?)?'
      read(*,200,end=20)zzlp
      write(*,200)'Name of file for solution(ie:so?)?'
      read(*,200,end=20)zzzsoln
*
* input training observation data values from file
      open(8,file=d,status='old')
      rewind(8)
      do i=1,m1+m2
        read(8,101) a(i,1),a(i,2),a(i,3)
      enddo
      close(8)
*
* to calculate the RHS'S for the initial problem input
      do j=1,3
         sum1(j)=0.
         sum2(j)=0.
      enddo
      do j=1,3
         do i=1,m1
           sum1(j)=sum1(j)+a(i,j)
         enddo
         do i=m1+1,m1+m2
           sum2(j)=sum2(j)+a(i,j)
         enddo
      enddo
```

```
      rhs2=sum1(1)-(cost*sum2(1))
      rhs3=sum1(2)-(cost*sum2(2))
      rhs4=sum1(3)-(cost*sum2(3))
      rhs5=0.
*
* to open the saved partial soln "names" file
      open(2,file='names',status='old')
      rewind(2)
*
* to open file for solutions history
      open(4,file=zzzsoln,status='new')
*
* to input holdout observation data values from files
      open(7,file=dh,status='old')
      rewind(7)
      do i=1,5*m1
        read(7,101) ah(i,1), ah(i,2), ah(i,3)
      enddo
      do i=(5*m1)+1,5*(m1+m2)
        read(7,101) ah(i,1), ah(i,2), ah(i,3)
      enddo
      close(7)
*
*
*
******************************************************
* enumeration algorithm stage                        *
******************************************************
*
* to initialize values
      subset=0
      iterat=0
      zu=10000
      zuold=10000
      minit=0
      do i=1,size
         empty(i)=0
      enddo
*
* to open file containing DMELP model of problem
* llunget is a modified version of LINDO's lunget
* which includes filename
      flname=zzlp
      call llunget(9,1,1,flname)
*
* to enumerate for each observation
      do i=1,m1+m2
```

```
*
* to retrieve DMELP model from file
        call retr(9)
*
* to increment counter of subsets
        subset=subset+1
*
* to record adjustments array values
        adjust(subset,1)=3
        adjust(subset,2)=i
*
* to set farthest observation in DMELP model
        call insert(1,m1+m2+i,1.,1)
*
* to set the simple upper bounds (SUB's)
        do k=1,m1+m2
          if(k .ne. i)then
              call setsub(m1+m2+k,1.)
          endif
        enddo
*
* to read old Y and V values from LINDO output
* for later determination of number of minor pivots
        if(i .ne. 1)then
          flname=finame
          open(3,file=flname,status='old')
          rewind(3)
          read(3,400) obj
          do k=1,m1+m2+m1+m2
             read(3,400) oldyv(k)
          enddo
          close(3)
        endif
*
* to retrieve partial solution
        if(i .ne. 1)then
          call clrbas
          flname=finame
          call llunget(3,1,0,flname)
          call rdbc(3)
          call klose(3)
          close(3)
        endif
*
* to run linear program
          call go(1000,kondn)
```

```
*
* to record cumulative total number of iterations
* from LINDO output
          iterat=iterat+mdlit
*
* to record infeasible (unbounded) zl values
          if(kondn .ne. 4)then
            zl(subset)=1000000
            go to 50
          endif
*
* to read feasible zl value from LINDO output
          call reprow(1,prim,dual)
          zl(subset)=prim
*
* to record new Y and V values from LINDO output
          do k=1,m1+m2+m1+m2
            call repvar(k,newyv(k),dual)
          enddo
*
* to calculate minor pivots
        if(i .ne. 1)then
        minor=0
        do k=m1+m2+1,m1+m2+m1+m2
         if(k .ne. i+100)then
         if(oldyv(k) .eq. 0. .and. newyv(k) .eq. 1.)then
               minor=minor+1
         endif
         if(oldyv(k) .eq. 1. .and. newyv(k) .eq. 0.)then
               minor=minor+1
         endif
         endif
        enddo
          minit=minit+minor
        endif
*
* to save the partial lindo solution
            read(2,130) flname
            open(3,file=flname,status='new')
            call sdbc(3)
            call klose(3)
            close(3)
*
* to save filname of partial LINDO solution
            slpred(subset)=flname
*
* to indicate solution is in secondary storage
            empty(subset)=1
            finame=flname
```

```
*
* to record weights and cutoff in summary file
          do k=2,5
            call reprow(k,prim,dual)
            wwwwwc(k)=dual
          enddo
          do k=1,3
            w(subset,k)=wwwwwc(k+1)
          enddo
          c(subset)=wwwwwc(5)
*
* to calculate the zr value for this subset
* and to record misclassification deviations
          zr1=0
          zr2=0
          do k=1,m1+m2
             t(k)=0.
          enddo
          do k=1,m1
          comb(k) = (w(subset,1) * a(k,1)) +
     &                (w(subset,2) * a(k,2)) +
     &                (w(subset,3) * a(k,3))
          if(comb(k) .gt. c(subset))then
              zr1=zr1+1
              t(k)=comb(k)-c(subset)
          endif
          enddo
          do k=m1+1,m1+m2
          comb(k) = (w(subset,1) * a(k,1)) +
     &                (w(subset,2) * a(k,2)) +
     &                (w(subset,3) * a(k,3))
          if(comb(k) .lt. c(subset))then
              zr2=zr2+1
              t(k)=c(subset)-comb(k)
          endif
          enddo
          zr(subset)=zr1+zr2*cost
```

```
*
* to reset zu if necessary and write to summary file
          if(zu .gt. zr(subset))then
            zu=zr(subset)
            write(4,200)'******************************'
            write(4,200)'Part soln: enumeration stage'
            write(4,200)'subset number = ',subset
            write(4,201)'zu = ',zu
            write(4,200)'cumulative iter= ',iterat
            write(4,200)'cumulative minor iter=',minit
            write(4,202)'w1=',w(subset,1),
     &                  ' w2=',w(subset,2),
     &                  '  c=',c(subset)
            write(4,302)'w3=',w(subset,3)
            write(4,200)'num of miscl in group 1= ',zr1
            write(4,200)'num of miscl in group 2= ',zr2
            write(4,201)'exp cost of miscl= ',zr(subset)
          endif
*
* to select the farthest observation for next branch
          observ=0
          hight=0.
          do k=1,m1+m2
            if(k .ne. i .and. t(k) .gt. hight)then
              hight=t(k)
              observ=k
            endif
          enddo
          adjust(subset,3)=observ
*
* to determine if solution is integer for ti values
*         zerone=1 if integer
*               =0 if fractional
*         zl should always be less than zr
*         if zl+.0001≥zr then soln is integer
          zerone(subset)=0
          zltrun=zl(subset)*1000
          zrtrun=zr(subset)*1000
          if(zltrun+1 .ge. zrtrun)then
            zerone(subset)=1
          endif
*
* to set at no descendant subsets
* ie: to permit later branching from this subset
*         descen=1 if yes
*               =0 if no
          descen(subset)=0
```

```
*
*
 50       continue
*
* to reinitialize LINDO problem space for next problem
          call init
        enddo
*
        write(4,200)'*****************************'
        write(4,200)'total enumeration pivots = ',iterat
        write(4,200)'total enum minor  pivots = ',minit
*
*
*******************************************************
*         Branch-and-Bound Algorithm                  *
*******************************************************
*
 85     continue
*
* to increment counter
        subset=subset+1
*
* to retrieve the DMELP model unchanged
        call retr(9)
*
* to determine the subset for branching
* ie: not binary, no descendants
        predec=0
        lowzl=10000.
        do k=1,subset-1
        if(zl(k) .le. zu-1.)then
         if(zerone(k) .eq. 0 .and. descen(k) .eq. 0)then
            if(zl(k) .le. lowzl)then
                lowzl=zl(k)
                predec=k
            endif
         endif
        endif
```

```
*
* to remove excess files from secondary storage
* in order to save disk file storage space
          if(zl(k).gt.zu-1. .or. descen(k).eq.1
     &      .or.zerone(k).eq.1)then
            if(empty(k) .eq. 1)then
              CH='rm '//slpred(k)
              call system(CH)
              empty(k)=0
            endif
          endif
        enddo
*
* if no predecessor subset the go to optimal section
* ie: no remaining subsets for branching
        if(predec .eq. 0)then
            go to 105
        endif
*
* to reset that predecessor subset now has decendants
        descen(predec)=1
*
* to set indicator for no simple upper bounds
      do k=1,m1+m2
         nosub(k)=0
      enddo
*
* to reconstruct predecessor subset
** to reset farthest obs & set no SUB for this obs
          farthe=adjust(predec,2)
          call insert(1,m1+m2+farthe,1.,1)
          nosub(1)=farthe
```

```
*
** to reset other fixed obs & calculate new RHS's &
** determine which obs should not have SUB's
      nrhs2=rhs2
      nrhs3=rhs3
      nrhs4=rhs4
      nrhs5=rhs5
      k=1
      if(adjust(predec,1) .gt. 3)then
         do i=4,adjust(predec,1)
*
** for correctly classified obs
           if(adjust(predec,i+1) .eq. 0)then
              call insert(1,adjust(predec,i),0.,1)
           endif
*
** for misclassified obs in group 1
           if(adjust(predec,i) .le. m1 .and.
     &          adjust(predec,i+1) .ne. 0)then
             k=k+1
             nosub(k)=adjust(predec,i)
             nrhs2=nrhs2-a(adjust(predec,i),1)
             nrhs3=nrhs3-a(adjust(predec,i),2)
             nrhs4=nrhs4-a(adjust(predec,i),3)
             nrhs5=nrhs5+1.
           endif
*
** for misclassified obs in group 2
           if(adjust(predec,i) .gt. m1 .and.
     &          adjust(predec,i+1) .ne. 0)then
             k=k+1
             nosub(k)=adjust(predec,i)
             nrhs2=nrhs2+cost*a(adjust(predec,i),1)
             nrhs3=nrhs3+cost*a(adjust(predec,i),2)
             nrhs4=nrhs4+cost*a(adjust(predec,i),3)
             nrhs5=nrhs5-cost
           endif
            i=i+1
           enddo
              call insert(2,0,nrhs2,1)
              call insert(3,0,nrhs3,1)
              call insert(4,0,nrhs4,1)
              call insert(5,0,nrhs5,1)
      endif
```

```
*
**********************************************
* to branch as correctly classified          *
**********************************************
*
            branch=adjust(predec,3)
            call insert(1,branch,0.,1)
*
* to set simple upper bounds (SUB's)
            do i=1,m1+m2
              no=0
              do k=1,m1+m2
                 if(nosub(k) .eq. i)then
                   no=1
                 endif
              enddo
              if(no .eq. 0)then
                 call setsub(m1+m2+i,1.)
              endif
            enddo
*
* to retrieve partial solution
            flname=slpred(predec)
            call clrbas
            call llunget(3,1,0,flname)
            call rdbc(3)
            call klose(3)
            close(3)
*
* to read old Y and V values from LINDO output
            open(3,file=flname,status='old')
            rewind(3)
            read(3,400) obj
            do k=1,m1+m2+m1+m2
               read(3,400) oldyv(k)
            enddo
            close(3)
*
* to run linear program
            call go(1000,kondn)
*
* to record cumulative number of iterations
            iterat=iterat+mdlit
```

```
*
* to record infeasible (unbounded) zl values
            if(kondn .ne. 4)then
              zl(subset)=1000000
              go to 60
            endif
*
* to record feasible zl values
            call reprow(1,prim,dual)
            zl(subset)=prim
*
* to read new Y and V values from LINDO output
          do k=1,m1+m2+m1+m2
            call repvar(k,newyv(k),dual)
          enddo
*
* to calculate minor pivots
          minor=0
          do k=m1+m2+1,m1+m2+m1+m2
            no=0
            do j=1,m1+m2
             if(nosub(j) .eq. k-100)then
               no=1
             endif
            enddo
            if(no .eq. 0)then
             if(oldyv(k) .eq. 0. .and.
     &              newyv(k) .eq. 1.)then
               minor=minor+1
             endif
             if(oldyv(k) .eq. 1. .and.
     &             newyv(k) .eq. 0.)then
               minor=minor+1
             endif
            endif
          enddo
          minit=minit+minor
```

```
*
* to change zl for previously miscl observations
            if(adjust(predec,1) .gt. 3)then
              do i=4,adjust(predec,1)
                if(adjust(predec,i+1) .eq. 1)then
                  if(adjust(predec,i) .le. m1)then
                    zl(subset)=zl(subset)+1
                  else
                    zl(subset)=zl(subset)+cost
                  endif
                endif
                i=i+1
              enddo
            endif
*
* to save the partial lindo solution
            if(zl(subset) .le. zu)then
              read(2,130) flname
              open(3,file=flname,status='new')
              call sdbc(3)
              call klose(3)
              close(3)
*
* to save filename of partial LINDO solution
              slpred(subset)=flname
*
* to indicate that the solution is in secondary storage
              empty(subset)=1
            endif
*
* to record weights and cutoff in summary file
            do k=2,5
              call reprow(k,prim,dual)
              wwwwwc(k)=dual
            enddo
            do k=1,3
               w(subset,k)=wwwwwc(k+1)
            enddo
            c(subset)=wwwwwc(5)
```

```
*
* to calculate the zr value for this subset
* and to record misclassification deviations
          zr1=0
          zr2=0
          do k=1,m1+m2
             t(k)=0.
          enddo
          do k=1,m1
            comb(k) = (w(subset,1) * a(k,1)) +
     &                (w(subset,2) * a(k,2)) +
     &                (w(subset,3) * a(k,3))
            if(comb(k) .gt. c(subset))then
              zr1=zr1+1
              t(k)=comb(k)-c(subset)
            endif
          enddo
          do k=m1+1,m1+m2
            comb(k) = (w(subset,1) * a(k,1)) +
     &                (w(subset,2) * a(k,2)) +
     &                (w(subset,3) * a(k,3))
            if(comb(k) .lt. c(subset))then
              zr2=zr2+1
              t(k)=c(subset)-comb(k)
            endif
          enddo
          zr(subset)=zr1+zr2*cost
*
* to set the adjustments array values
            adjust(subset,1)=adjust(predec,1)+2
            adjust(subset,2)=adjust(predec,2)
            if(adjust(predec,1) .gt. 3)then
              do i=4,adjust(predec,1)
                adjust(subset,i)=adjust(predec,i)
              enddo
            endif
            adjust(subset,adjust(predec,1)+1)=branch
            adjust(subset,adjust(predec,1)+2)=0
```

```
*
* to reset zu if necessary and write to summary file
            if(zu .gt. zr(subset))then
              zu=zr(subset)
              write(4,200)'******************************'
              write(4,200)'Part soln: B-A-B stage'
              write(4,200)'subset number = ',subset
              write(4,201)'zu = ',zu
              write(4,200)'cumulative iter= ',iterat
              write(4,200)'cumulative minor iter=',minit
              write(4,202)'w1=',w(subset,1),
     &                   ' w2=',w(subset,2),
     &                   '  c=',c(subset)
              write(4,302)'w3=',w(subset,3)
              write(4,200)'num of miscl in group 1= ',zr1
              write(4,200)'num of miscl in group 2= ',zr2
              write(4,201)'exp cost of miscl= ',zr(subset)
              write(4,200)'******************************'
*
* to determine holdout misclassifications
                    miscl1=0
                    miscl2=0
                    do i=1,5*m1
                       tdev=w(subset,1) * ah(i,1) +
     &                      w(subset,2) * ah(i,2) +
     &                      w(subset,3) * ah(i,3) -
     &                      c(subset)
                       if(tdev .gt. 0.0)then
                         miscl1=miscl1+1
                       endif
                    enddo
                    do i=(5*m1)+1,5*(m1+m2)
                     tdev = c(subset) -
     &                      w(subset,1) * ah(i,1) -
     &                      w(subset,2) * ah(i,2) -
     &                      w(subset,3) * ah(i,3)
                     if(tdev .gt. 0.0)then
                          miscl2=miscl2+1
                     endif
                    enddo
                    write(4,201)'Holdout miscl1=',miscl1
                    write(4,201)'Holdout miscl2=',miscl2
                    write(4,201)'proportion miscl=',
     &                  (miscl1+miscl2)/(5*(m1+m2))
*
                    write(4,200)'**********************'
                    write(4,200)'**********************'
            endif
```

```
*
* to select the farthest observation for next branch
            observ=0
            hight=0.
            do k=1,m1+m2
*
* to eliminate obs already partitioned
              do i=2,adjust(subset,1)
                 if(k .eq. adjust(subset,i))then
                   go to 55
                 endif
                 i=i+1
              enddo
*
* to select farthest from obs not already partitioned
              if(t(k) .gt. hight)then
                hight=t(k)
                observ=k
              endif
   55         continue
            enddo
            adjust(subset,3)=observ
*
* to determine if solution is integer for ti values
*             zerone=1 if integer
*                   =0 if fractional
*             zl should always be less than zr
*             if zl+.0001≥zr then soln is integer
              zerone(subset)=0
              zltrun=zl(subset)*1000
              zrtrun=zr(subset)*1000
              if(zltrun+1 .ge. zrtrun)then
                zerone(subset)=1
              endif
*
* to set at no descendant subsets (permit later branching)
*           descen=1 if yes
*                 =0 if no
            descen(subset)=0
*
* to prevent further branching if all miscl
* already partitioned
            if(observ .eq. 0)then
              descen(subset)=1
            endif
*
   60       continue
```

```
*
******************************************
* to branch as misclassified             *
******************************************
*
* to increment counter
            subset=subset+1
*
* to reinitialize LINDO storage
            call init
*
* to retrieve the original DMELP model
            call retr(9)
*
* to set No SUB's indicator to have SUB's
            do k=1,m1+m2
               nosub(k)=0
            enddo
*
* to reconstruct predecessor subset
** to reset farthest observation
          farthe=adjust(predec,2)
          call insert(1,m1+m2+farthe,1.,1)
          nosub(1)=farthe
*
** to set current branch obs as miscl, ie related si=0
          k=2
          nosub(k)=branch
*
** to determine the new RHS adjusted for branch
          nrhs2=rhs2
          nrhs3=rhs3
          nrhs4=rhs4
          nrhs5=rhs5
*
* for group 1 observations
          if(branch .le. m1)then
            nrhs2=nrhs2-a(branch,1)
            nrhs3=nrhs3-a(branch,2)
            nrhs4=nrhs4-a(branch,3)
            nrhs5=nrhs5+1.
          else
*
* for group 2 observations
            nrhs2=nrhs2+cost*a(branch,1)
            nrhs3=nrhs3+cost*a(branch,2)
            nrhs4=nrhs4+cost*a(branch,3)
            nrhs5=nrhs5-cost
          endif
```

```
*
** to reset other fixed observations & calc new RHS's
      if(adjust(predec,1) .gt. 3)then
         do i=4,adjust(predec,1)
*
* for correct classifications
            if(adjust(predec,i+1) .eq. 0)then
               call insert(1,adjust(predec,i),0.,1)
            endif
*
* for misclassifications from group 1
            if(adjust(predec,i) .le. m1 .and.
     &         adjust(predec,i+1) .ne. 0)then
               k=k+1
               nosub(k)=adjust(predec,i)
               nrhs2=nrhs2-a(adjust(predec,i),1)
               nrhs3=nrhs3-a(adjust(predec,i),2)
               nrhs4=nrhs4-a(adjust(predec,i),3)
               nrhs5=nrhs5+1
            endif
*
* for misclassifications from group 2
            if(adjust(predec,i) .gt. m1 .and.
     &         adjust(predec,i+1) .ne. 0)then
               k=k+1
               nosub(k)=adjust(predec,i)
               nrhs2=nrhs2+cost*a(adjust(predec,i),1)
               nrhs3=nrhs3+cost*a(adjust(predec,i),2)
               nrhs4=nrhs4+cost*a(adjust(predec,i),3)
               nrhs5=nrhs5-cost
             endif
             i=i+1
             enddo
      endif
               call insert(2,0,nrhs2,1)
               call insert(3,0,nrhs3,1)
               call insert(4,0,nrhs4,1)
               call insert(5,0,nrhs5,1)
```

```
*
* to set simple upper bounds (SUB's)
            do i=1,m1+m2
              no=0
              do k=1,m1+m2
                 if(nosub(k) .eq. i)then
                   no=1
                 endif
              enddo
              if(no .eq. 0)then
                 call setsub(m1+m2+i,1.)
              endif
            enddo
*
* to retrieve partial solution
            flname=slpred(predec)
            call clrbas
            call llunget(3,1,0,flname)
            call rdbc(3)
*
* to run linear program
            call go(1000,kondn)
            call klose(3)
*
* record cumulative number of iterations
            iterat=iterat+mdlit
*
* to record infeasible (unbounded) z1 values
            if(kondn .ne. 4)then
              z1(subset)=1000000
              go to 70
            endif
*
* to record feasible z1 values
            call reprow(1,prim,dual)
            z1(subset)=prim
```

```
*
* to change zl for previously miscl obs
            if(branch .le. m1)then
              zl(subset)=zl(subset)+1
            else
              zl(subset)=zl(subset)+cost
            endif
            if(adjust(predec,1) .gt. 3)then
              do i=4,adjust(predec,1)
                if(adjust(predec,i+1) .eq. 1)then
                  if(adjust(predec,i) .le. m1)then
                    zl(subset)=zl(subset)+1
                  else
                    zl(subset)=zl(subset)+cost
                  endif
                endif
                i=i+1
              enddo
            endif
*
* to save the partial lindo solution
            if(zl(subset) .le. zu)then
              read(2,130) flname
              open(3,file=flname,status='new')
              call sdbc(3)
              call klose(3)
              close(3)
              slpred(subset)=flname
              empty(subset)=1
            endif
*
* to record new Y and V values
          do k=1,m1+m2+m1+m2
            call repvar(k,newyv(k),dual)
          enddo
```

```
*
* to calculate minor pivots
          minor=0
          do k=m1+m2+1,m1+m2+m1+m2
            no=0
            do j=1,m1+m2
             if(nosub(j) .eq. k-100)then
               no=1
             endif
            enddo
            if(no .eq. 0)then
             if(oldyv(k) .eq. 0. .and.
     &          newyv(k) .eq. 1.)then
               minor=minor+1
             endif
             if(oldyv(k) .eq. 1. .and.
     &          newyv(k) .eq. 0.)then
               minor=minor+1
             endif
            endif
          enddo
          minit=minit+minor
*
* to record weights and cutoff
            do k=2,5
              call reprow(k,prim,dual)
              wwwwwc(k)=dual
            enddo
            do k=1,3
               w(subset,k)=wwwwwc(k+1)
            enddo
            c(subset)=wwwwwc(5)
```

```
*
* to calculate the zr value for this subset
* and record misclassification deviations
        zr1=0
        zr2=0
        do k=1,m1+m2
           t(k)=0.
        enddo
        do k=1,m1
          comb(k) = (w(subset,1) * a(k,1)) +
     &              (w(subset,2) * a(k,2)) +
     &              (w(subset,3) * a(k,3))
          if(comb(k) .gt. c(subset))then
            zr1=zr1+1
            t(k)=comb(k)-c(subset)
          endif
        enddo
        do k=m1+1,m1+m2
          comb(k) = (w(subset,1) * a(k,1)) +
     &              (w(subset,2) * a(k,2)) +
     &              (w(subset,3) * a(k,3))
            if(comb(k) .lt. c(subset))then
              zr2=zr2+1
              t(k)=c(subset)-comb(k)
            endif
        enddo
        zr(subset)=zr1+zr2*cost
*
* to set values in adjustments array
          adjust(subset,1)=adjust(predec,1)+2
          adjust(subset,2)=adjust(predec,2)
          adjust(subset,adjust(predec,1)+1)=branch
          adjust(subset,adjust(predec,1)+2)=1
          if(adjust(predec,1) .gt. 3)then
            do i=4,adjust(predec,1)
              adjust(subset,i)=adjust(predec,i)
            enddo
          endif
```

```
*
* to reset zu if necessary
            if(zu .gt. zr(subset))then
              zu=zr(subset)
              write(4,200)'**************************'
              write(4,200)'Part soln: B-A-B stage'
              write(4,200)'subset number=',subset
              write(4,201)'zu = ',zu
              write(4,200)'cumulative iter=',iterat
              write(4,200)'cumulative minor iter=',minit
              write(4,202)'w1=',w(subset,1),
     &                    ' w2=',w(subset,2),
     &                    '  c=',c(subset)
              write(4,302)'w3=',w(subset,3)
              write(4,200)'num of miscl in group 1= ',zr1
              write(4,200)'num of miscl in group 2= ',zr2
              write(4,201)'exp cost of miscl= ',zr(subset)
              write(4,200)'**************************'
*
* to determine holdout misclassifications
                    miscl1=0
                    miscl2=0
                    do i=1,5*m1
                       tdev= w(subset,1) * ah(i,1) +
     &                       w(subset,2) * ah(i,2) +
     &                       w(subset,3) * ah(i,3) -
     &                       c(subset)
                       if(tdev .gt. 0.0)then
                         miscl1=miscl1+1
                       endif
                    enddo
                    do i=(5*m1)+1,5*(m1+m2)
                        tdev = c(subset) -
     &                         w(subset,1) * ah(i,1) -
     &                         w(subset,2) * ah(i,2)
     &                       - w(subset,3) * ah(i,3)
                        if(tdev .gt. 0.0)then
                           miscl2=miscl2+1
                        endif
                    enddo
                    write(4,201)'Holdout miscl1=',miscl1
                    write(4,201)'Holdout miscl2=',miscl2
                    write(4,201)'proportion miscl=',
     &                  (miscl1+miscl2)/(5*(m1+m2))
                    write(4,200)'***********************'
                    write(4,200)'***********************'
            endif
```

```
*
* to select the farthest observation for next branch
           observ=0
           hight=0.
           do k=1,m1+m2
*
* to eliminate obs already partitioned
             do i=2,adjust(subset,1)
                if(k .eq. adjust(subset,i))then
                  go to 65
                endif
                i=i+1
             enddo
*
* to select farthest obs not already partitioned
             if(t(k) .gt. hight)then
               hight=t(k)
               observ=k
             endif
   65        continue
           enddo
           adjust(subset,3)=observ
*
* to determine if solution is integer for ti values
*          zerone=1 if integer
*                =0 if fractional
           zerone(subset)=0
           zltrun=zl(subset)*1000
           zrtrun=zr(subset)*1000
             if(zltrun+1 .ge. zrtrun)then
               zerone(subset)=1
             endif
*
* to set at no descendant subsets
*          descen=1 if yes
*                =0 if no
           descen(subset)=0
*
   70      continue
*
* to go back and continue
         call init
         go to 85
 105     continue
```

```
*
******************************************
* to determine the optimal subset number   *
******************************************
          do i=1,subset-1
             if(zr(i) .eq. zu )then
               optsub=i
               go to 117
             else
               optsub=0
             endif
          enddo
 117      continue
*
       write(4,200)'*********************************'
       write(4,200)'*********************************'
       write(4,200)'the optimal subset is ',optsub
       write(4,200)'the cumulative iterations = ',iterat
       write(4,200)'cumulative minor iterations=',minit
       write(4,200)'the num of sub's solved= ',subset-1
       write(4,202)'w1=',w(optsub,1),' w2=',w(optsub,2),
     &            '  c=',c(optsub)
       write(4,302)'w3=',w(optsub,3)
       write(4,201)'exp cost of miscl is = ',zr(optsub)
       write(4,200)'*********************************'
                   miscl1=0
                   miscl2=0
                   do i=1,5*m1
                      tdev = w(optsub,1) * ah(i,1) +
     &                       w(optsub,2) * ah(i,2) +
     &                       w(optsub,3) * ah(i,3) -
     &                       c(optsub)
                      if(tdev .gt. 0.0)then
                        miscl1=miscl1+1
                      endif
                   enddo
                   do i=(5*m1)+1,5*(m1+m2)
                     tdev = c(optsub) -
     &                      w(optsub,1) * ah(i,1) -
     &                      w(optsub,2) * ah(i,2) -
     &                      w(optsub,3) * ah(i,3)
                       if(tdev .gt. 0.0)then
                          miscl2=miscl2+1
                       endif
                   enddo
                   write(4,201)'Holdout miscl1=',miscl1
                   write(4,201)'Holdout miscl2=',miscl2
                   write(4,201)'proportion miscl=',
     &                 (miscl1+miscl2)/(5*(m1+m2))
```

```
*
 500   continue
 20    continue
       CH='rm zzzz'
       call system(CH)
       CH='rm y*'
       call system(CH)
       call klose(2)
       call klose(3)
       call klose(4)
       call klose(7)
       call klose(8)
       call klose(9)
       close(2)
       close(3)
       close(4)
       close(7)
       close(8)
       close(9)
       go to 700
 750   continue
       stop
       end
```

APPENDIX B

RESULTS FOR EACH EXPERIMENT

APPENDIX B1	TRAINING SAMPLE MISCLASSIFICATION RATES
APPENDIX B2	HOLDOUT SAMPLE MISCLASSIFICATION RATES
APPENDIX B3	INCREASED SAMPLE SIZE RESULTS FOR CELLS 2 AND 9

APPENDIX B1

TRAINING SAMPLE MISCLASSIFICATION RATES BY EXPERIMENT

CELL 1: TRAINING SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.310	.300	.260	.260	.260	.260
2	.300	.300	.260	.240	.240	.240
3	.290	.310	.280	.260	.260	.260
4	.340	.330	.310	.310	.310	.290
5	.420	.380	.350	.350	.350	.320
6	.350	.350	.310	.290	.290	.280
7	.290	.250	.240	.240	.240	.240
8	.320	.330	.270	.260	.260	.260
9	.300	.290	.210	.210	.210	.200
10	.320	.270	.250	.220	.220	.220
11	.370	.360	.330	.330	.330	.310
12	.280	.290	.220	.220	.220	.220
13	.340	.320	.310	.270	.270	.260
14	.300	.300	.240	.240	.240	.220
15	.270	.240	.220	.220	.220	.210
16	.340	.330	.250	.250	.250	.250
17	.380	.360	.290	.280	.280	.280
18	.360	.360	.320	.320	.300	.300
19	.300	.270	.250	.250	.250	.240
20	.330	.340	.340	.330	.300	.300
Mean	.326	.314	.276	.268	.265	.258
Standard Deviation	.038	.039	.042	.042	.039	.035

CELL 2: TRAINING SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.260	.250	.220	.220	.220	.220
2	.380	.350	.320	.320	.320	.310
3	.320	.320	.310	.280	.280	.260
4	.270	.280	.250	.230	.230	.210
5	.290	.280	.250	.250	.250	.240
6	.340	.330	.300	.300	.300	.290
7	.330	.290	.260	.260	.260	.240
8	.340	.310	.270	.270	.270	.260
9	.330	.320	.270	.270	.270	.260
10	.330	.340	.280	.270	.270	.250
11	.300	.290	.270	.250	.250	.250
12	.280	.270	.240	.240	.240	.240
13	.250	.250	.220	.220	.220	.210
14	.330	.290	.290	.290	.290	.270
15	.300	.290	.270	.270	.270	.270
16	.330	.340	.290	.280	.280	.240
17	.350	.360	.320	.300	.300	.290
18	.230	.210	.170	.170	.170	.170
19	.230	.200	.200	.200	.200	.170
20	.380	.370	.300	.290	.270	.250
Mean	.309	.297	.265	.259	.258	.245
Standard Deviation	.044	.046	.040	.037	.037	.036

CELL 3: TRAINING SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.330	.300	.270	.270	.270	.260
2	.310	.260	.230	.230	.230	.220
3	.280	.240	.220	.220	.220	.220
4	.310	.280	.270	.230	.230	.230
5	.310	.270	.240	.240	.240	.230
6	.350	.330	.300	.300	.300	.290
7	.330	.310	.310	.280	.280	.260
8	.310	.310	.290	.290	.290	.280
9	.290	.260	.250	.250	.250	.230
10	.340	.340	.340	.280	.280	.280
11	.280	.280	.240	.240	.240	.230
12	.390	.330	.310	.310	.310	.290
13	.270	.260	.240	.230	.230	.230
14	.230	.220	.220	.190	.190	.190
15	.390	.430	.310	.310	.310	.310
16	.370	.360	.310	.310	.310	.290
17	.360	.330	.310	.300	.300	.300
18	.260	.250	.240	.230	.230	.220
19	.300	.280	.240	.240	.240	.230
20	.330	.320	.320	.280	.280	.280
Mean	.317	.298	.273	.262	.262	.254
Standard Deviation	.043	.049	.038	.037	.036	.034

CELL 4: TRAINING SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.300	.270	.250	.230	.230	.220
2	.330	.320	.280	.280	.280	.260
3	.310	.300	.270	.270	.250	.250
4	.340	.310	.260	.230	.230	.230
5	.280	.280	.230	.230	.230	.230
6	.300	.300	.280	.230	.230	.230
7	.250	.230	.230	.220	.210	.200
8	.360	.340	.290	.290	.290	.290
9	.270	.230	.220	.210	.190	.170
10	.320	.310	.270	.270	.270	.250
11	.300	.270	.220	.220	.220	.210
12	.290	.250	.230	.220	.220	.210
13	.200	.210	.200	.190	.190	.190
14	.340	.340	.290	.270	.270	.270
15	.400	.360	.330	.280	.280	.260
16	.270	.230	.230	.230	.230	.210
17	.320	.280	.270	.270	.270	.250
18	.330	.320	.320	.300	.290	.280
19	.320	.320	.230	.230	.230	.220
20	.310	.310	.300	.290	.290	.270
Mean	.307	.289	.260	.248	.245	.235
Standard Deviation	.042	.042	.036	.032	.033	.032

CELL 5: TRAINING SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.280	.260	.240	.240	.240	n/a
2	.290	.260	.220	.220	.220	n/a
3	.270	.270	.240	.240	.240	n/a
4	.290	.270	.210	.210	.210	n/a
5	.280	.260	.230	.230	.230	n/a
6	.350	.320	.280	.280	.280	n/a
7	.280	.270	.210	.210	.200	n/a
8	.310	.270	.230	.230	.230	n/a
9	.200	.190	.170	.170	.170	n/a
10	.280	.270	.250	.250	.250	n/a
11	.220	.210	.190	.180	.180	n/a
12	.300	.380	.220	.220	.220	n/a
13	.220	.200	.200	.180	.180	n/a
14	.270	.290	.240	.240	.240	n/a
15	.350	.320	.290	.290	.290	n/a
16	.320	.330	.310	.300	.300	n/a
17	.330	.300	.270	.240	.240	n/a
18	.310	.350	.240	.240	.240	n/a
19	.340	.350	.290	.290	.290	n/a
20	.310	.310	.230	.230	.230	n/a
Mean	.290	.279	.238	.234	.234	n/a
Standard Deviation	.041	.045	.036	.036	.036	n/a

CELL 6: TRAINING SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.310	.320	.260	.260	.260	n/a
2	.340	.330	.320	.320	.290	n/a
3	.370	.380	.310	.290	.290	n/a
4	.250	.240	.230	.230	.230	n/a
5	.340	.320	.290	.280	.280	n/a
6	.310	.350	.300	.300	.300	n/a
7	.400	.400	.280	.280	.280	n/a
8	.310	.330	.280	.250	.250	n/a
9	.290	.300	.270	.270	.270	n/a
10	.340	.310	.300	.300	.300	n/a
11	.300	.310	.240	.240	.240	n/a
12	.260	.280	.260	.240	.240	n/a
13	.250	.270	.240	.240	.240	n/a
14	.280	.270	.240	.240	.240	n/a
15	.330	.330	.310	.300	.300	n/a
16	.310	.360	.280	.280	.280	n/a
17	.340	.320	.240	.240	.240	n/a
18	.360	.320	.290	.290	.270	n/a
19	.260	.250	.210	.210	.210	n/a
20	.330	.300	.280	.280	.280	n/a
Mean	.314	.314	.272	.267	.264	n/a
Standard Deviation	.041	.040	.030	.029	.027	n/a

CELL 7: TRAINING SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.390	.370	.350	.350	.340	n/a
2	.260	.290	.260	.230	.230	n/a
3	.270	.290	.290	.270	.260	n/a
4	.300	.300	.250	.250	.230	n/a
5	.320	.280	.280	.280	.280	n/a
6	.300	.270	.250	.240	.230	n/a
7	.340	.320	.280	.280	.280	n/a
8	.230	.190	.180	.180	.180	n/a
9	.330	.310	.300	.280	.280	n/a
10	.330	.310	.310	.280	.260	n/a
11	.240	.220	.190	.190	.190	n/a
12	.310	.310	.260	.260	.260	n/a
13	.370	.360	.330	.310	.310	n/a
14	.310	.290	.280	.280	.280	n/a
15	.340	.340	.300	.290	.290	n/a
16	.360	.380	.300	.300	.300	n/a
17	.280	.280	.240	.240	.240	n/a
18	.390	.360	.300	.300	.300	n/a
19	.310	.320	.280	.280	.280	n/a
20	.330	.330	.280	.270	.270	n/a
Mean	.316	.306	.275	.268	.264	n/a
Standard Deviation	.045	.047	.041	.039	.039	n/a

CELL 8: TRAINING SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.350	.350	.320	.300	.300	n/a
2	.270	.290	.290	.290	.260	n/a
3	.440	.440	.300	.300	.300	n/a
4	.380	.370	.300	.300	.300	n/a
5	.240	.250	.190	.190	.190	n/a
6	.350	.340	.290	.290	.290	n/a
7	.290	.290	.230	.230	.230	n/a
8	.260	.270	.230	.220	.220	n/a
9	.330	.320	.240	.240	.240	n/a
10	.320	.320	.260	.260	.260	n/a
11	.340	.300	.260	.260	.260	n/a
12	.320	.310	.300	.300	.300	n/a
13	.250	.240	.200	.190	.190	n/a
14	.260	.290	.250	.200	.200	n/a
15	.270	.280	.250	.240	.240	n/a
16	.240	.240	.240	.240	.240	n/a
17	.320	.350	.300	.290	.290	n/a
18	.260	.280	.210	.210	.210	n/a
19	.260	.290	.230	.230	.230	n/a
20	.190	.210	.180	.180	.180	n/a
Mean	.297	.302	.254	.248	.246	n/a
Standard Deviation	.058	.052	.041	.042	.041	n/a

CELL 9: TRAINING SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.280	.310	.240	.230	.230	n/a
2	.290	.280	.240	.240	.240	n/a
3	.370	.390	.310	.290	.290	n/a
4	.250	.290	.220	.220	.220	n/a
5	.210	.260	.220	.220	.200	n/a
6	.340	.290	.260	.240	.240	n/a
7	.260	.270	.260	.260	.250	n/a
8	.330	.320	.270	.240	.240	n/a
9	.270	.290	.210	.210	.200	n/a
10	.370	.340	.300	.300	.280	n/a
11	.310	.280	.280	.280	.280	n/a
12	.220	.260	.230	.210	.210	n/a
13	.230	.220	.180	.180	.180	n/a
14	.390	.380	.290	.290	.290	n/a
15	.230	.230	.180	.180	.160	n/a
16	.370	.320	.310	.310	.290	n/a
17	.370	.380	.330	.330	.330	n/a
18	.350	.350	.310	.270	.270	n/a
19	.280	.300	.210	.210	.210	n/a
20	.260	.230	.230	.210	.210	n/a
Mean	.299	.300	.254	.246	.241	n/a
Standard Deviation	.058	.050	.045	.043	.044	n/a

CELL 10: TRAINING SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.300	.350	.280	.260	.260	.250
2	.260	.250	.240	.230	.230	.230
3	.340	.330	.310	.310	.290	.250
4	.350	.360	.270	.270	.270	.260
5	.370	.380	.310	.300	.300	.290
6	.250	.270	.230	.230	.230	.210
7	.190	.220	.160	.160	.160	.150
8	.300	.250	.250	.250	.250	.230
9	.320	.330	.280	.280	.280	.260
10	.320	.320	.220	.220	.220	.210
11	.270	.280	.190	.190	.190	.190
12	.290	.310	.260	.230	.230	.230
13	.400	.390	.310	.310	.310	.290
14	.340	.350	.270	.260	.260	.250
15	.290	.290	.290	.240	.240	.240
16	.250	.250	.220	.220	.220	.200
17	.250	.270	.220	.220	.220	.190
18	.290	.320	.280	.260	.260	.250
19	.260	.250	.240	.240	.220	.200
20	.280	.290	.250	.250	.250	.240
Mean	.296	.303	.253	.246	.244	.231
Standard Deviation	.049	.048	.041	.038	.036	.035

CELL 11: TRAINING SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.360	.380	.310	.270	.270	n/a
2	.280	.280	.270	.230	.230	n/a
3	.340	.320	.250	.250	.250	n/a
4	.250	.240	.220	.220	.220	n/a
5	.300	.310	.270	.270	.260	n/a
6	.290	.300	.250	.250	.250	n/a
7	.320	.310	.250	.250	.250	n/a
8	.240	.230	.230	.230	.230	n/a
9	.250	.310	.200	.200	.200	n/a
10	.200	.220	.190	.190	.190	n/a
11	.260	.250	.250	.250	.250	n/a
12	.350	.360	.350	.330	.310	n/a
13	.210	.190	.190	.190	.190	n/a
14	.270	.250	.240	.240	.240	n/a
15	.300	.280	.280	.260	.260	n/a
16	.260	.250	.210	.210	.210	n/a
17	.250	.290	.230	.230	.230	n/a
18	.320	.330	.310	.260	.260	n/a
19	.330	.380	.300	.300	.270	n/a
20	.190	.190	.160	.150	.150	n/a
Mean	.279	.284	.248	.239	.236	n/a
Standard Deviation	.049	.056	.047	.040	.036	n/a

CELL 12: TRAINING SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.330	.250	.230	.190	.190	n/a
2	.320	.180	.180	.180	.180	n/a
3	.310	.280	.260	.220	.220	n/a
4	.290	.300	.280	.250	.250	n/a
5	.270	.260	.190	.180	.180	n/a
6	.240	.240	.220	.220	.220	n/a
7	.250	.270	.220	.210	.210	n/a
8	.290	.280	.250	.250	.250	n/a
9	.320	.320	.280	.280	.280	n/a
10	.250	.280	.270	.260	.250	n/a
11	.200	.220	.170	.170	.170	n/a
12	.270	.250	.250	.250	.180	n/a
13	.310	.330	.230	.230	.230	n/a
14	.240	.250	.220	.220	.220	n/a
15	.210	.230	.210	.180	.180	n/a
16	.350	.340	.230	.230	.230	n/a
17	.310	.310	.280	.280	.280	n/a
18	.280	.310	.250	.230	.230	n/a
19	.340	.350	.270	.270	.250	n/a
20	.330	.340	.290	.250	.250	n/a
Mean	.286	.280	.239	.227	.222	n/a
Standard Deviation	.043	.045	.035	.034	.033	n/a

APPENDIX B2

HOLDOUT SAMPLE MISCLASSIFICATION RATES BY EXPERIMENT

CELL 1: HOLDOUT SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.368	.376	.364	.364	.364	.370
2	.352	.358	.370	.358	.358	.358
3	.368	.380	.368	.386	.386	.382
4	.334	.336	.368	.368	.368	.364
5	.354	.336	.326	.326	.326	.340
6	.310	.312	.310	.316	.358	.326
7	.314	.320	.320	.320	.320	.304
8	.338	.344	.336	.316	.316	.316
9	.320	.308	.344	.344	.344	.334
10	.346	.336	.308	.332	.332	.328
11	.300	.306	.318	.318	.318	.314
12	.330	.340	.342	.342	.342	.318
13	.338	.328	.418	.328	.328	.348
14	.328	.330	.334	.334	.334	.342
15	.342	.352	.352	.352	.352	.348
16	.368	.366	.352	.352	.352	.396
17	.370	.376	.354	.456	.456	.352
18	.390	.382	.384	.384	.414	.374
19	.382	.392	.384	.384	.384	.390
20	.388	.386	.386	.388	.384	.402
Mean	.347	.348	.352	.354	.357	.350
Standard Deviation	.026	.027	.029	.035	.036	.029

CELL 2: HOLDOUT SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.344	.346	.356	.356	.356	.356
2	.328	.322	.348	.348	.348	.324
3	.334	.340	.334	.356	.356	.340
4	.322	.332	.358	.344	.344	.358
5	.334	.338	.334	.334	.334	.346
6	.312	.326	.336	.336	.338	.348
7	.308	.322	.336	.336	.336	.326
8	.294	.300	.354	.354	.354	.328
9	.318	.322	.346	.346	.346	.338
10	.312	.320	.322	.330	.330	.316
11	.310	.308	.300	.326	.324	.326
12	.340	.334	.336	.336	.336	.336
13	.312	.312	.340	.340	.340	.310
14	.290	.286	.286	.286	.286	.346
15	.316	.316	.350	.350	.350	.350
16	.314	.316	.384	.412	.412	.444
17	.318	.326	.376	.324	.324	.286
18	.320	.306	.306	.306	.306	.306
19	.340	.352	.352	.352	.352	.336
20	.322	.316	.380	.318	.364	.384
Mean	.320	.322	.342	.340	.342	.340
Standard Deviation	.014	.016	.025	.025	.023	.032

CELL 3: HOLDOUT SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.330	.332	.344	.344	.344	.384
2	.332	.340	.380	.380	.380	.372
3	.336	.338	.344	.344	.344	.352
4	.326	.330	.352	.388	.388	.388
5	.366	.368	.376	.376	.376	.348
6	.324	.328	.338	.338	.338	.354
7	.326	.344	.344	.364	.364	.366
8	.346	.344	.372	.372	.372	.342
9	.346	.350	.378	.378	.378	.344
10	.360	.366	.366	.356	.356	.356
11	.350	.336	.342	.342	.342	.348
12	.342	.340	.340	.340	.340	.340
13	.334	.336	.348	.336	.336	.336
14	.336	.364	.364	.348	.348	.348
15	.340	.332	.330	.330	.330	.330
16	.320	.328	.348	.348	.348	.478
17	.336	.336	.334	.340	.340	.340
18	.360	.366	.370	.362	.374	.372
19	.346	.352	.368	.368	.368	.386
20	.352	.346	.346	.370	.370	.358
Mean	.340	.344	.354	.356	.357	.362
Standard Deviation	.013	.013	.016	.017	.018	.032

CELL 4: HOLDOUT SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.334	.314	.360	.352	.352	.368
2	.314	.318	.362	.362	.362	.350
3	.346	.352	.304	.304	.322	.336
4	.314	.310	.316	.322	.322	.322
5	.310	.330	.332	.332	.322	.332
6	.296	.302	.300	.332	.332	.332
7	.306	.336	.336	.330	.294	.324
8	.290	.268	.280	.280	.280	.280
9	.320	.304	.336	.306	.296	.310
10	.312	.320	.310	.310	.310	.298
11	.292	.296	.308	.308	.308	.304
12	.316	.330	.320	.378	.378	.316
13	.342	.340	.338	.322	.322	.322
14	.330	.324	.324	.366	.366	.342
15	.334	.342	.404	.340	.340	.354
16	.344	.358	.358	.358	.358	.336
17	.324	.330	.368	.368	.368	.338
18	.320	.322	.322	.350	.330	.352
19	.310	.318	.326	.326	.326	.324
20	.342	.342	.372	.338	.342	.334
Mean	.320	.323	.334	.334	.332	.329
Standard Deviation	.017	.021	.030	.025	.027	.021

CELL 5: HOLDOUT SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.336	.330	.334	.334	.334	n/a
2	.380	.390	.370	.370	.370	n/a
3	.332	.342	.334	.334	.334	n/a
4	.380	.378	.378	.378	.378	n/a
5	.308	.326	.312	.312	.312	n/a
6	.358	.354	.358	.358	.358	n/a
7	.360	.376	.370	.370	.392	n/a
8	.336	.362	.358	.358	.358	n/a
9	.354	.358	.332	.332	.332	n/a
10	.334	.348	.336	.336	.336	n/a
11	.334	.328	.324	.324	.324	n/a
12	.308	.326	.300	.300	.300	n/a
13	.300	.302	.302	.312	.312	n/a
14	.390	.400	.386	.386	.386	n/a
15	.386	.370	.336	.336	.336	n/a
16	.368	.346	.348	.330	.330	n/a
17	.356	.358	.324	.322	.322	n/a
18	.354	.320	.370	.370	.370	n/a
19	.350	.362	.296	.296	.296	n/a
20	.362	.356	.354	.354	.354	n/a
Mean	.349	.352	.341	.341	.342	n/a
Standard Deviation	.026	.025	.027	.027	.028	n/a

CELL 6: HOLDOUT SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.350	.354	.322	.322	.322	n/a
2	.330	.336	.288	.288	.330	n/a
3	.306	.306	.316	.334	.334	n/a
4	.328	.330	.334	.334	.334	n/a
5	.334	.346	.336	.338	.338	n/a
6	.366	.364	.392	.392	.392	n/a
7	.338	.336	.308	.308	.308	n/a
8	.298	.286	.302	.328	.328	n/a
9	.408	.428	.366	.366	.366	n/a
10	.338	.358	.356	.356	.356	n/a
11	.388	.402	.368	.368	.368	n/a
12	.396	.380	.374	.368	.368	n/a
13	.354	.360	.364	.364	.364	n/a
14	.356	.354	.362	.362	.362	n/a
15	.328	.330	.324	.358	.358	n/a
16	.316	.330	.334	.334	.334	n/a
17	.370	.368	.360	.360	.360	n/a
18	.340	.346	.344	.344	.354	n/a
19	.358	.354	.356	.356	.356	n/a
20	.326	.320	.338	.338	.338	n/a
Mean	.346	.349	.342	.346	.348	n/a
Standard Deviation	.029	.032	.026	.024	.020	n/a

CELL 7: HOLDOUT SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.340	.346	.394	.394	.402	n/a
2	.304	.302	.326	.308	.308	n/a
3	.348	.346	.346	.338	.338	n/a
4	.314	.310	.286	.286	.346	n/a
5	.330	.342	.342	.342	.342	n/a
6	.318	.322	.316	.310	.324	n/a
7	.364	.380	.348	.348	.348	n/a
8	.326	.328	.336	.336	.336	n/a
9	.342	.328	.328	.312	.312	n/a
10	.322	.328	.328	.348	.326	n/a
11	.292	.318	.332	.332	.332	n/a
12	.314	.318	.302	.302	.302	n/a
13	.306	.318	.460	.360	.360	n/a
14	.338	.320	.314	.314	.314	n/a
15	.332	.328	.334	.308	.308	n/a
16	.344	.354	.348	.348	.348	n/a
17	.298	.294	.304	.304	.304	n/a
18	.346	.340	.370	.370	.370	n/a
19	.338	.356	.364	.364	.364	n/a
20	.350	.354	.338	.360	.360	n/a
Mean	.328	.331	.343	.334	.337	n/a
Standard Deviation	.019	.021	.038	.027	.026	n/a

CELL 8: HOLDOUT SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.318	.336	.344	.352	.352	n/a
2	.302	.336	.336	.336	.302	n/a
3	.354	.356	.392	.392	.392	n/a
4	.360	.344	.310	.310	.310	n/a
5	.328	.346	.330	.330	.330	n/a
6	.326	.330	.338	.338	.338	n/a
7	.352	.354	.366	.366	.366	n/a
8	.328	.350	.320	.316	.316	n/a
9	.322	.334	.322	.322	.322	n/a
10	.312	.322	.320	.320	.320	n/a
11	.312	.314	.360	.360	.360	n/a
12	.344	.362	.360	.360	.360	n/a
13	.322	.320	.316	.360	.360	n/a
14	.326	.300	.356	.348	.348	n/a
15	.322	.328	.326	.332	.332	n/a
16	.340	.344	.344	.344	.344	n/a
17	.344	.356	.360	.334	.334	n/a
18	.322	.350	.344	.344	.344	n/a
19	.308	.344	.316	.316	.316	n/a
20	.328	.316	.340	.340	.340	n/a
Mean	.329	.337	.340	.341	.339	n/a
Standard Deviation	.016	.017	.021	.020	.022	n/a

CELL 9: HOLDOUT SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.290	.308	.290	.320	.320	n/a
2	.306	.326	.318	.318	.318	n/a
3	.354	.368	.292	.302	.302	n/a
4	.296	.302	.322	.322	.322	n/a
5	.330	.354	.340	.340	.322	n/a
6	.300	.330	.332	.344	.344	n/a
7	.328	.360	.312	.312	.342	n/a
8	.304	.336	.292	.348	.348	n/a
9	.320	.342	.306	.306	.344	n/a
10	.358	.388	.290	.290	.336	n/a
11	.292	.302	.302	.302	.302	n/a
12	.332	.326	.324	.316	.316	n/a
13	.346	.372	.352	.352	.352	n/a
14	.310	.320	.302	.302	.302	n/a
15	.318	.330	.284	.284	.288	n/a
16	.302	.350	.316	.316	.336	n/a
17	.420	.422	.446	.446	.446	n/a
18	.342	.384	.356	.304	.304	n/a
19	.316	.330	.322	.322	.322	n/a
20	.272	.274	.274	.288	.288	n/a
Mean	.322	.341	.319	.321	.327	n/a
Standard Deviation	.032	.035	.037	.035	.034	n/a

CELL 10: HOLDOUT SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.296	.334	.306	.372	.372	.362
2	.334	.362	.348	.330	.330	.330
3	.290	.318	.280	.280	.342	.320
4	.280	.322	.294	.294	.294	.260
5	.332	.350	.290	.316	.316	.366
6	.306	.330	.304	.304	.304	.290
7	.354	.380	.356	.356	.356	.352
8	.350	.380	.380	.380	.380	.348
9	.318	.370	.304	.304	.304	.376
10	.338	.380	.316	.316	.316	.312
11	.278	.286	.288	.288	.288	.294
12	.328	.344	.342	.324	.324	.330
13	.344	.362	.312	.312	.312	.328
14	.340	.352	.348	.362	.362	.326
15	.306	.320	.320	.326	.326	.316
16	.308	.300	.302	.302	.302	.288
17	.318	.330	.368	.368	.368	.340
18	.304	.328	.312	.294	.294	.294
19	.300	.320	.288	.288	.366	.300
20	.320	.330	.306	.306	.306	.342
Mean	.317	.340	.318	.321	.327	.324
Standard Deviation	.023	.027	.029	.031	.030	.030

CELL 11: HOLDOUT SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.378	.404	.406	.356	.356	n/a
2	.288	.298	.350	.342	.342	n/a
3	.328	.354	.326	.326	.326	n/a
4	.310	.322	.318	.318	.318	n/a
5	.338	.384	.358	.358	.362	n/a
6	.300	.316	.298	.298	.298	n/a
7	.284	.286	.328	.328	.328	n/a
8	.336	.338	.338	.338	.338	n/a
9	.284	.320	.294	.294	.294	n/a
10	.314	.318	.294	.294	.294	n/a
11	.314	.320	.320	.320	.320	n/a
12	.340	.370	.312	.298	.304	n/a
13	.274	.278	.278	.278	.278	n/a
14	.316	.344	.310	.310	.310	n/a
15	.302	.330	.330	.280	.280	n/a
16	.318	.328	.324	.324	.324	n/a
17	.330	.352	.280	.280	.280	n/a
18	.294	.314	.306	.290	.290	n/a
19	.340	.346	.312	.312	.336	n/a
20	.300	.310	.300	.302	.302	n/a
Mean	.314	.332	.319	.312	.314	n/a
Standard Deviation	.025	.031	.029	.024	.025	n/a

CELL 12: HOLDOUT SAMPLE MISCLASSIFICATION RATES

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS	OPT
1	.302	.324	.300	.306	.306	n/a
2	.318	.320	.320	.320	.320	n/a
3	.244	.246	.246	.258	.258	n/a
4	.258	.296	.296	.282	.282	n/a
5	.292	.296	.318	.360	.360	n/a
6	.306	.310	.284	.284	.284	n/a
7	.316	.332	.318	.322	.322	n/a
8	.340	.352	.344	.344	.344	n/a
9	.368	.382	.474	.474	.474	n/a
10	.312	.332	.320	.340	.314	n/a
11	.290	.324	.300	.300	.300	n/a
12	.318	.332	.332	.332	.328	n/a
13	.332	.340	.310	.310	.310	n/a
14	.312	.332	.338	.338	.338	n/a
15	.298	.324	.308	.294	.294	n/a
16	.332	.336	.352	.352	.352	n/a
17	.274	.276	.276	.276	.276	n/a
18	.296	.292	.292	.290	.290	n/a
19	.288	.304	.312	.312	.334	n/a
20	.298	.334	.318	.328	.328	n/a
Mean	.305	.319	.318	.321	.320	n/a
Standard Deviation	.028	.029	.044	.045	.045	n/a

APPENDIX B3

INCREASED SAMPLE SIZE RESULTS FOR CELLS 2 AND 9

CELL 2: HOLDOUT SAMPLE MISCLASSIFICATION RATES
(Training sample size $M_1+M_2=200$)
(Holdout sample size = 1000)

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS
1	.314	.309	.329	.329	.329
2	.324	.308	.306	.306	.314
3	.319	.324	.341	.341	.341
4	.350	.335	.337	.337	.337
5	.324	.319	.343	.335	.335
6	.323	.321	.316	.328	.328
7	.364	.377	.411	.411	.411
8	.332	.329	.330	.330	.330
9	.314	.313	.310	.310	.310
10	.321	.313	.335	.321	.321
11	.320	.326	.347	.340	.340
12	.328	.326	.332	.332	.332
13	.346	.352	.356	.365	.338
14	.346	.353	.347	.401	.401
15	.347	.332	.360	.360	.350
16	.359	.364	.352	.352	.352
17	.346	.347	.347	.347	.347
18	.355	.349	.346	.346	.374
19	.320	.321	.356	.356	.356
20	.316	.319	.342	.342	.342
Mean	.333	.332	.343	.344	.344
Standard Deviation	.016	.019	.022	.026	.026

CELL 2: HOLDOUT SAMPLE MISCLASSIFICATION RATES
(Training sample size $M_1+M_2=500$)
(Holdout sample size = 2500)

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS
1	.328	.328	.336	.336	.332
2	.339	.341	.338	.338	.338
3	.335	.331	.331	.331	.352
4	.338	.334	.335	.337	.337
5	.344	.343	.346	.359	.359
Mean	.336	.335	.337	.340	.344
Standard Deviation	.006	.006	.006	.011	.011

CELL 9: HOLDOUT SAMPLE MISCLASSIFICATION RATES
(Training sample size $M_1+M_2=200$)
(Holdout sample size = 1000)

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS
1	.336	.355	.338	.338	.338
2	.315	.348	.320	.320	.341
3	.324	.335	.315	.311	.311
4	.348	.352	.313	.313	.313
5	.302	.329	.288	.288	.287
6	.298	.319	.300	.309	.310
7	.349	.359	.295	.295	.324
8	.340	.356	.313	.319	.319
9	.300	.333	.308	.308	.308
10	.371	.349	.326	.326	.341
11	.306	.332	.275	.299	.297
12	.326	.326	.287	.287	.287
13	.331	.334	.314	.330	.330
14	.307	.325	.302	.332	.332
15	.314	.317	.301	.301	.301
16	.321	.348	.293	.293	.293
17	.301	.325	.298	.298	.298
18	.320	.326	.300	.300	.300
19	.326	.347	.314	.314	.314
20	.319	.335	.305	.305	.305
Mean	.323	.338	.305	.309	.312
Standard Deviation	.019	.013	.015	.015	.017

CELL 9: HOLDOUT SAMPLE MISCLASSIFICATION RATES
(Training sample size $M_1+M_2=500$)
(Holdout sample size = 2500)

EXPERIMENT NUMBER	LDA	GKD	SS	TS	NS
1	.324	.349	.319	.319	.319
2	.316	.333	.306	.304	.312
3	.306	.336	.285	.285	.285
4	.315	.328	.291	.291	.291
5	.302	.310	.279	.279	.279
Mean	.316	.331	.296	.295	.297
Standard Deviation	.015	.014	.016	.016	.017

BIBLIOGRAPHY

Abad, P.L. and Banks, W.J., (1991), "New LP based heuristics for the classification problem", received final acceptance for publication by European Journal of Operational Research.

Aldrich, J.H. and Nelson, F.D., (1984), Linear Probability, Logit and Probit Models, Sage Publications, London.

Altman, E.I. and Eisenbeis, R.A., (1978), "Financial applications of discriminant analysis: a clarification", Journal of Financial and quantitative Analysis, Vol.13, pp.185-195.

Bajgier, S.M. and Hill, A.V., (1982), "An experimental comparison of statistical and linear programming approaches to the discriminant problem", Decision Sciences, Vol. 13, No. 4, pp.604-618.

Banks, W.J., (1988), "Classification as a mathematical programming model", Unpublished doctoral disertation proposal, McMaster University, April 15.

Banks, W.J. and Abad, P.A., (1991), "An efficient optimal solution algorithm for the classification problem", received final acceptance for publication by Decision Sciences.

Banks, W.J., and Abad, P.L., (1991a), "On the performance of a linear programming heuristic applied on a quadratic transformation in the classification problem", resubmitted for publication to Decision Sciences.

Bobrowski, L., (1986), "Linear discrimination with symmetrical models", Pattern Recognition, Vol. 19(1), pp.101-109.

Capon, N., (1982), "Credit scoring systems: a critical analysis", Journal of Marketing, Vol. 46, pp.82-91.

Chand, D.R., & Kapur, S.S., (1970), "An algorithm for convex polytopes", Journal of the Association for Computing Machinery, Vol. 17, pp.78-86.

Cooper, R.G., (1979), "The dimensions of industrial new product success and failure", Journal of Marketing, Vol. 43, pp.93-103.

Dillon, W., and Goldstein, M., (1984), Multivariate analysis: methods and applications, Wiley, New York.

Duea, R.W.Jr., (1985), Goal Programming Models For The Two-Group Discriminant Problem: An Experimental Comparison of Three Normalizations, Doctoral Thesis University of Colorado, Boulder Colorado.

Dyer, M.E., (1983), "The complexity of vertex enumeration methods", Mathematics of Operations Research, Vol. 8, No. 3, pp.381-402.

Eisenbeis, R.A., (1977), "Pitfalls in the application of discriminant analysis in business, finance and economics", Journal of Finance, Vol. 32, No. 3, pp.875-899.

Eisenbeis, R.A., (1978), "Problems in applying discriminant analysis in credit scoring models", Journal of Banking and Finance, Vol. 2, pp.205-219.

Erenguc, S.S. & Koehler, G.J., (1990), "Survey of mathematical programming models and experimental results for linear discriminant analysis", Managerial and Decision Economics, Vol. 11, pp.215-225.

Fisher, R.A., (1936), "The use of multiple measurements in taxonomic problems", Annals of Eugenics, Vol. 7, pp.179-188.

Fleishman, A.I., (1978), "A method for simulating non-normal distributions", Psychometrika, Vol. 43, No. 4, pp.521-532.

Freed, N. and Glover, F., (1981), "Simple but powerful goal programming models for discriminant problems", European Journal of Operational Research, Vol. 7, pp.44-60.

Freed, N. and Glover, F., (1981a), "A linear programming approach to the discriminant problem", Decision Sciences, Vol. 12, pp.68-74.

Freed, N. and Glover, F., (1982), "Linear programming and statistical discrimination- the LP side", Decision Sciences, Vol. 13, No. 1, pp.172-175.

Freed , N. and Glover, F., (1986), "Evaluating alternative linear programming models to solve the two-group discriminant problem", Decision Sciences, Vol. 17, No. 2, pp.151-162.

Freed, N. and Glover, F., (1986a), "Resolving certain difficulties and improving the classification power of LP discriminant analysis formulations", Decision Sciences, Vol. 17, pp.589-595.

Gehrlein, W.V., (1986), "General mathematical programming formulations for the statistical classification problem", Operations Research Letters, Vol. 5, pp.229-304.

Glorfeld, L.W., and Gaither, N., (1982), "On linear programming in discriminant analysis", Decision Sciences, Vol. 13, No. 1, pp.167-171.

Glorfeld, L.W., and Olsen, D.L., (1982), "Using the L1 metric for robust analysis of the two group discriminant problem", 1982 Procedings, Decision Sciences Institute, Atlanta, G.A.

Glover, F., (1990), "Improved linear programming models for discriminant analysis", Decision Sciences, Vol. 21, pp.771-785.

Glover, F., Keene, S. and Duea, B., (1988), "A new class of models for the discriminant problem", Decision Sciences, Vol. 19, pp.269-280.

Goldstein, M. & Dillon, W.R., (1978), Discrete Discriminant Analysis, John Wiley & Sons, New York.

Hand, D.J., (1981), Discrimination and Classification, Wiley, New York.

Hillier, F.S., and Lieberman, G.J., (1980), Introduction to Operations Research, Third Edition, Holden-Day Inc., San Francisco.

International Mathematical and Statistical Library, (1987), IMSL Problem Solving Software Systems, Houston, Texas.

Joachimsthaler, E.A., and Stam, A., (1988), "Four approaches to the classification problem in discriminant analysis: an experimental comparison", Decision Sciences, Vol. 19, pp.322-333

Joachimsthaler, E.A. & Stam, A., (1990), "Mathematical programming approaches for the classification problem in two-group discriminant analysis", Multivariate Behavioral Research, vol. 25(4), pp.427-454.

Joy, O.M. and Tollefson, J.O., (1975), "On the financial applications of discriminant analysis", Journal of Financial and Quantitative Analysis, Vol. 10, pp.723-739.

Joy, O.M. and Tollefson, J.O., (1978), "Some clarifying comments on discriminant analysis", Journal of Financial and Quantitative Analysis, Vol.13, pp.197-200.

Kallay, M., (1986), "Convex Hulls Made Easy", Information Processing Letters, Vol. 22, pp.161.

Kaplan, R.S. and Urwitz, G., (1979), "Statistical models of bond ratings: A methodological inquiry", Journal of Business, Vol. 52, pp.231-261.

Kirby, M.J.L., Love, H.R., & Swarup, K., (1972), "Extreme point mathematical programming", Management Science, Vol. 18, No. 9, pp.540-549.

Klecka, W.R., (1980), Discriminant Analysis, Sage Publications, London.

Koehler, G.J., (1989), "Characterizations of unacceptable solutions in LP discriminant analysis", Decision Sciences, Vol. 20, pp.239-257.

Koehler, G.J. and Erenguc, S., (1990), "Minimizing misclassifications in linear discriminant analysis", Decision Sciences, Vol.21, No. 1, pp.63-85.

Kolesar, P., and Showers, J.L., (1985), "A robust credit screening model using categorical data", Management Science, Vol. 31, No. 2, pp.123-133.

Lachenbruch, P.A., (1975), Discriminant Analysis, MacMillan Publishing Co., New York.

Lastovicka, J.L., Murry, J.P.Jr.,Joachimthaler, E.A., Bhalla, G., and Scheurich, J., (1987), "A lifestyle topology to model young male drinking and driving", Journal of Consumer Research, Vol. 14, pp.257-263.

Lee, D.T. and Preparata,F.P., (1984), "Computational geometry - a survey", IEEE Transactions on Computers, Vol. c-33, No. 12, pp.1072-1101.

Liittschwager, J.M., and Wang, C., (1978), "Integer programming solution of a classification problem", Management Science, Vol. 24, No. 14, pp.1515-1525.

Mahmood, M.A., and Lawrence, E.C., (1987), "A performance analysis of parametric and nonparametric discriminant approaches to business decision making", Decision Sciences, Vol. 18, No. 2, pp.308-326.

Markowski, E.P., and Markowski, C.A., (1985), "Some difficulties and improvements in applying linear programming formulations to the discriminant problem", Decision Sciences, Vol. 16, No. 3, pp.237-247.

Markowski, C.A., and Markowski, E.P., (1987), "An experimental comparison of several approaches to the discriminant problem with both qualitative and quantitative variables", European Journal of Operational Research, 28, pp.74-78.

Nath, R. and Jones, T.W., (1990), "A variable selection criterion in the linear programming approaches to discriminant analysis", Decision Sciences, Vol. 19, pp.554-563

Neter, J., Wasserman, W., & Kutner, M.H., (1985), Applied Linear Statistical Models, Second Edition, Irwin, Homewood Illinois.

Norusis, M.J., (1985), Advanced Statistics Guide SPSSX, McGraw-Hill, New York.

Rubin, P.A., (1990), "A comparison of linear programming and parametric approaches to the two-group discriminant problem", Decision Sciences, Vol.21, pp.373-386.

Rubin, P.A., (1990a), "Heuristic solution procedures for a mixed-integer programming discriminant model", Managerial and Decision Economics, Vol. 11, pp.255-266.

Schrage, L., (1986), Linear, Integer and Quadratic Programming with LINDO, Third Edition, The Scientific Press, Palo Alto, California

Scott, E., (1978), "On the financial application of discriminant analysis: comment", Journal of Financial and Quantitative Analysis, Vol. 13, pp201-205.

Showers, J.L. and Chakrin, L.M., (1981), "Reducing uncollectible revenue from residential telephone customers", Interfaces, Vol. 11, No. 6, pp.21-34.

Smith, C., (1947), "Some examples of discrimination", Annals of Eugenics, Vol.13, pp.272-282.

Smith, F.W., (1968), "Pattern classifier design by linear programming", IEEE Transactions on Computers, Vol. c-17, No. 4, pp.367-372.

Smith, F.W., (1969), "Design of multicategory pattern classifiers with two-category classifier design procedures", IEEE Transactions on Computers, Vol. c-18, No. 6, pp.548-551.

Stam, A., (1990), "Extensions of mathematical programming based classification rules: a multicriteria approach", European Journal of Operational Research, Vol. 48, pp.351-361.

Stam, A. and Joachimsthaler, E.A., (1990), "A comparison of a robust mixed-integer approach to existing methods for establishing classification rules for the discriminant problem", European Journal of Operational Research, 46, pp.113-122.

Stam, A. & Jones, D.G., (1990), "Classification performance of mathematical programming techniques in discriminant analysis: results for small and medium sample sizes", Managerial and Decision Economics, Vol.11, pp.243-253.

Stone, M. & Rasp, J., (1991), "Tradeoffs in the choice between logit and OLS for accounting choice studies", The Accounting Review, Vol. 66, No. 1, pp.170-187.

Tiku, M.L., Tan, W.Y. and Balakrishnan, N., (1986), Robust Interence, Marcel Dekker Inc., New York.

Vale, C.D., and Maurelli, V.A., (1983), "Simulating multivariate nonnormal distributions", Psychometrika, Vol. 48, No. 3, pp.465-471.

Wang, C., (1982), "Derivation of a maximum rank sum statistic and application to discriminant analysis", TIMS/Studies in Management Science, Vol. 19, pp.153-164.

Wiginton, J.C., (1980), "A note on the comparison of logit and discriminant models of consumer credit behavior", Journal of Financial and Quantitative Analysis, Vol. 15, No. 3, pp.757-768.

Zionts, S., (1974), Linear and Integer Programming, Prentice-Hall Inc., New Jersey