CLASSIFICATION MODELS FOR DISEASE DIAGNOSIS AND OUTCOME ANALYSIS

A Dissertation
Presented to
The Academic Faculty

by

Tsung-Lin Wu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology
August 2011

CLASSIFICATION MODELS FOR DISEASE DIAGNOSIS AND OUTCOME ANALYSIS

Approved by:

Dr. Eva K. Lee, Advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Yajun Mei
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Ellis L. Johnson
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Michael Krauthammer, MD
Department of Pathology
Yale University

Dr. Renato D.C. Monteiro
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: July 6, 2011

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my advisor, Dr. Eva Lee, for her supervision, guidance, and financial support for these years. Without her, the accomplishment of this dissertation would have been impossible.

I want to express my appreciation to Dr. Ellis Johnson, Dr. Renato Monteiro, Dr. Yajun Mei, and Dr. Michael Krauthammer, MD for serving on my dissertation committee. In particular, I am thankful to their valuable feedback on my work.

I want to acknowledge the National Science Foundation and the National Institutes of Health for their funding support.

I would like to acknowledge many ISyE faculty who teach me professional knowledge, ISyE staff who give me administrative support, and current and former fellow students who help me in both of my academic and personal life. All of them play important roles during my doctoral study.

I also want to thank all my friends in Atlanta for their company in these years. They make my life delightful, and I am particularly pleased that I have gained some lifelong friendships.

Finally, I would like to thank my parents, my sister, and family members for their endless love and support.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . . **iii**

LIST OF TABLES . . . . . **vii**

LIST OF FIGURES . . . . . **xi**

SUMMARY . . . . . **xiii**

I CLASSIFICATION **1**

1.1 Introduction 1

1.2 Classification via Mathematical Programming 3

1.2.1 Linear Programming Classification Models 3

1.2.2 Mixed Integer Programming Classification Models 15

1.2.3 Nonlinear Programming Classification Models 23

1.2.4 Support Vector Machine 27

1.3 Anderson's Model and DAMIP 31

1.3.1 Anderson's Model 31

1.3.2 DAMIP 32

II FEATURE SELECTION **37**

2.1 Introduction 37

2.2 Feature Relevance and Redundancy 39

2.2.1 Feature Relevance 39

2.2.2 Feature Redundancy 40

2.2.3 Remarks 41

2.3 Feature Ranking 42

2.4 Feature Subset Selection 44

2.5 Feature Subset Selection via Mathematical Programming 49

2.6 Particle Swarm Optimization 54

2.6.1 Introduction to PSO 54

2.6.2 Binary PSO 59

2.6.3 Feature Selection Using PSO 60

III PSO/DAMIP FRAMEWORK 65

3.1 Modified PSO 65

3.1.1 Algorithm 65

3.1.2 Parameter Selection 67

3.2 Solving Two-Group DAMIP by Exact Algorithms 71

3.3 Solving DAMIP without Misclassification Constraints by Greedy Algorithm 85

3.3.1 Introduction 85

3.3.2 Computational Study—Three-Group Case 87

3.3.3 Computational Study—More-Than-Three-Group Case 98

3.4 Trials on Solving DAMIP with Cuts 118

3.4.1 Combinatorial Benders' Cuts 118

3.4.2 Projected Chvatal-Gomory Cuts 120

IV APPLICATIONS 122

4.1 Alzheimer's Disease 122

4.1.1 Background 122

4.1.2 Results 126

4.2 Cardiovascular Disease 133

4.2.1 Background 133

4.2.2 Results 137

4.3 Sulfur Amino Acid Intake 146

4.3.1 Background 146

4.3.2 Results 151

4.4 CpG Islands 154

4.4.1 Background 154

4.4.2 Data Description 155

4.4.3 Pattern Search, Feature Selection, and Classification 156

4.4.4 Results 163

REFERENCES **172**

LIST OF TABLES

3.3.1 Configurations in settings of the computational study 88

3.3.2 Group sizes in settings of the computational study 89

3.3.3 Results about reserved observations in the computational study . . . 92

3.3.4 List of one-round strategies 93

3.3.5 Good strategies in each setting 94

3.3.6 Probabilities of being good strategies in each scenario 96

3.3.7 Suggested strategies of the greedy algorithm given certain conditions of the data sets 96

3.3.8 Comparison between solutions by CPLEX and by the greedy algorithm 98

3.3.9 "Dermatology": Mahalanobis distances between groups 99

3.3.10 "Dermatology": Classification results by DAMIP solved by CPLEX . 100

3.3.11 "Dermatology": Classification results by DAMIP solved by greedy algorithm 100

3.3.12 "Dermatology": Classification results by Bayes 101

3.3.13 "Dermatology": Classification results by DALP 101

3.3.14 "Ecoli": Mahalanobis distances between groups 102

3.3.15 "Ecoli": Classification results by DAMIP solved by CPLEX 104

3.3.16 "Ecoli": Classification results by DAMIP solved by greedy algorithm 104

3.3.17 "Ecoli": Classification results by Bayes 104

3.3.18 "Ecoli": Classification results by DALP 105

3.3.19 "Heart Disease": Mahalanobis distances between groups 108

3.3.20 "Heart Disease": Classification results by DAMIP solved by greedy algorithm 108

3.3.21 "Heart Disease": Classification results by Bayes 108

3.3.22 "Heart Disease": Classification results by DALP 110

3.3.23 "Heart Disease": Classification results by DAMIP solved by greedy algorithm using the maximizing-minimum-group-accuracy objective . 110

3.3.24 "Nursery": Mahalanobis distances between groups 113

3.3.25 "Nursery": Classification results by DAMIP solved by greedy algorithm 114

3.3.26 "Nursery": Classification results by DAMIP solved by greedy algorithm using the maximizing-minimum-group-accuracy objective . . . 115

3.3.27 "Nursery": Classification results by Bayes 115

3.3.28 "Nursery": Classification results by DALP 115

4.1.1 Number of subjects of three groups from two trials in data from Emory 126

4.1.2 Classification results of Emory data. Five selected features: MMSE–cMMtotal, WordList–cWL2Butter, WordList–cWL2Queen, WordList–cWL2Ticket, GDS–GDS13. 127

4.1.3 Classification results of Emory data. Four selected features: MMSE–cMMtotal, MMSE–cMMz, WordList–cWL1Queen, GDS–GDS13; or five selected features: MMSE–cMMtotal, MMSE–cMMz, WordList–cWL2Butter, WordList–cWL1Queen, GDS–GDS13. 128

4.1.4 Classification results of Emory data. Five selected features: MMSE–cMMsRapple, WordList–cWL1Queen, WordList–cWL3Engine, GDS–GDS9, GDS–GDS13. 128

4.1.5 Classification results of Emory data. Five selected features: MMSE–cMMtotal, WordList–cWL3Queen, WordList–cWL2Engine, GDS–GDS13, GDS–GDS15. 129

4.1.6 Classification results of Emory data, training trial 1 and blind predicting trial 2 from 100 features. Five selected features: MMSE–cMMsCounty, MMSE–cMMsWorld, Clock–cClockHands1, WordList–cWL2Queen, WordList–cWRyShore. 129

4.1.7 Classification results of Emory data, training trial 1 and blind predicting trial 2 from 100 features. Five selected features: MMSE–cMMsCounty, MMSE–cMMsWorld, Clock–cClockHands1, WordList–cWL2Queen, WordList–cWRyArm. 130

4.1.8 Classification results of Emory data, training trial 2 and blind predicting trial 1 from 100 features. Five selected features: Age, MMSE–cMMsRapple, WordList–cWL2Queen, WordList–cWL2Engine, GDS–GDS13. 130

4.1.9 Classification results of Emory data from 9 score-type features. Two selected features: MMSE–cMMtotal, Word List–cWLcorTotal. 131

4.1.10 Classification results of LONI/ADNI data. Five selected features: CLOCKHAND, AVTOT5, AVTOT6, CATVEGESC, TRABERROM. 131

4.1.11 Classification results of LONI/ADNI data. Five selected features: AVTOT5, AVTOT6, CATVEGESC, TRABSCOR, TRABERROM. . . . 132

4.2.1 Three grouping ways for classification 136

4.2.2 Classification results on IMT, selecting five features: Age, BMI, Family CAD history, Eh GSH/GSSG, d-ROM 137

4.2.3 Classification results on IMT, selecting six features: Age, Gender, Hypertension, Family CAD history, Eh GSH/GSSG, d-ROM 138

4.2.4 Feature appearance for grouping by IMT of all 75% or better results . 139

4.2.5 Classification results on IMT, selecting (1) six features: Age, Gender, BMI, TC, Hypertension, Family CAD history, or (2) six features: Age, Gender, BMI, LDL, HDL, TC 140

4.2.6 Classification results on FMD, selecting five features: Gender, BMI, LDL, Hypertension, Family CAD history 141

4.2.7 Classification results on FMD, selecting six features: Gender, Hypertension, GSSG, CySS, CySSG, Diastolic BP 141

4.2.8 Feature appearance for grouping by FMD of all 70% or better results 142

4.2.9 Classification results on IMT and FMD, selecting (1) four features: Age, HDL, Hypertension, Family CAD history, (2) five features: Age, HDL, Hypertension, Family CAD history, Fasting insulin, or (3) six features: Age, HDL, Hypertension, Family CAD history, Framingham risk score, CySS 143

4.2.10 Feature appearance for grouping by IMT and FMD of all 80% or better results 145

4.3.1 First classification results on SAA 151

4.3.2 Second classification results on SAA 153

4.4.1 Categorization of CpG islands 155

4.4.2 Letters and their meanings 157

4.4.3 Complements of letters 157

4.4.4 Number of patterns under fixed numbers of match and wild letters . . 164

4.4.5 Number of patterns under specific criteria 165

4.4.6 Number of patterns in pattern pools for CpG islands 166

4.4.7 Number of patterns in pattern pools for extended 167

4.4.8 Number of patterns in pattern pools for outer 167

4.4.9 Classification results on CpG islands, selecting nine features: (ANGGCHA, TDGCCNT), (BGSAA, TTSCV), (CCCBGTK, MACVGGG), (AACCBBA, TVVGGTT), (AAGTVAV, BTBACTT), (AGMGTTR, YAACKCT), (CAHGWTG, CAWCDTG), (CGCCCGCGC, GCGCGGGCG), (GTCGCDD, HHGCGAC) 168

4.4.10 Classification results on CpG islands, selecting ten features: (AGCYAGS, SCTRGCT), (CGGCGGASG, CSTCCGCCG), (AAGTMAV, BTKACTT), (AHYTACC, GGTARDT), (TANGTNA, TNACNTA), (CAGAWTD, HAWTCTG), (HCKGTGA, TCACMGD), (BATCSAA, TTSGATV), (CASWAGG, CCTWSTG), (AKTDGAA, TTCHAMT) 169

4.4.11 Classification results on Extended, selecting ten features: (CAHTAGK, MCTADTG), (GVCTKTA, TAMAGBC), (AGGTDTV, BAHACCT), (CAMTAGB, VCTAKTG), (CSCACCCCC, GGGGGTGSG), (ACGTAVM, KBTACGT), (ABTCCYA, TRGGAVT), (CGGHANA, TNTDCCG), (BAGGTKC, GMACCTV), (BTACAGY, RCTGTAV) . . . 169

4.4.12 Classification results on Extended, selecting ten features: (GGCABTD, HAVTGCC), (AVGGCWA, TWGCCBT), (DGCTGCAA, TTGCAGCH), (ACACAGVG, CBCTGTGT), (CAMTAGB, VCTAKTG), (CSCACCCCC, GGGGGTGSG), (AACTRRG, CYYAGTT), (CGGHAHA, TDTDCCG), (GGCTGGAA, TTCCAGCC), (GGGAGAAA, TTTCTCCC) 170

4.4.13 Classification results on Outer, selecting ten features: (GAWSGAC, GTCSWTC), (AVCTGGCC, GGCCAGBT), (CDGTYG, CRACHG), (RCCGANA, TNTCGGY), (AGATNGS, SCNATCT), (AHGHTAG, CTADCDT), (CAHTAGK, MCTADTG), (CTTWRAC, GTYWAAG), (CDAACCD, HGGTTHG), (KATCCAM, KTGGATM) 170

4.4.14 Classification results on Outer, selecting ten features: (GAWSGAC, GTCSWTC), (CDGTYG, CRACHG), (RCCGANA, TNTCGGY), (ATAMGCH, DGCKTAT), (ATGMTAG, CTAKCAT), (AGATNGS, SCNATCT), (CAHTAGK, MCTADTG), (CTTWRAC, GTYWAAG), (CDAACCD, HGGTTHG), (KATCCAM, KTGGATM) 171

LIST OF FIGURES

2.4.1 Feature subset selection for classification. 44

2.4.2 Four steps of feature subset selection [90]. 45

2.6.1 Neighborhood topology of PSO. (a) *lbest*; (b) von Neumann. 57

3.1.1 Behavior of PSO in an example: (a) number of particle moving, (b) number of objective calculation, (c) best accuracy, and (d) number of particle improving. 70

3.2.1 Two-group problem on the $\frac{f_2(\boldsymbol{x}_i)}{f_1(\boldsymbol{x}_i)}$ axis. (a) Representation of observations. (b) The lines are partitioned at $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$ into correctly-classified regions (C) and misclassified regions (M). (c) The lines are partitioned at $\frac{\pi_1}{\lambda_{21}}$ and $\frac{\lambda_{12}}{\pi_2}$ into correctly-classified regions (C), misclassified regions (M), and reserved judgement regions (R). 76

3.2.2 Representation of observations allowed to be misclassified. 77

3.2.3 Cases in which reserved judgement region is not or is needed. (a) $a_1 < a_2$. Reserved judgement region is not needed. (b) $a_1 \geq a_2$. Reserved judgement region is needed. 78

3.2.4 The interval (l_L, l_R) in which optimal $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$ is located. (a) No misclassification constraints. (b) A constraint on group 1. (c) A constraint on group 2. (d) Constraints on both groups with $a_1 < a_2$. 80

3.2.5 Positions to check the optimal value of $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$ within (l_L, l_R). 80

3.2.6 Two-group problem on the $\pi_2\frac{f_2(\boldsymbol{x}_i)}{f_1(\boldsymbol{x}_i)} - \pi_1$ axis given that $\lambda_{21} = 0$. The lines are partitioned at λ_{12} into correctly-classified regions (C) and misclassified regions (M). 82

3.3.1 Configurations in settings of the computational study. 90

3.3.2 "Ecoli": Histogram of the objective values of all one-round strategies for the greedy algorithm 103

3.3.3 "Ecoli": 10-fold CV accuracy on all possible subsets of the features . 106

3.3.4 "Ecoli": Minimum group accuracy (10-fold CV) on all possible subsets of the features 107

3.3.5 "Heart Disease": Histogram of the objective values of all one-round strategies for the greedy algorithm 109

3.3.6 "Heart Disease": 10-fold CV accuracy on all possible subsets of the features 111

3.3.7 “Heart Disease”: Minimum group accuracy (10-fold CV) on all possible subsets of the features 112

3.3.8 “Ecoli”: Histogram of the objective values of all one-round strategies for the greedy algorithm 113

3.3.9 “Nursery”: 10-fold CV accuracy on all possible subsets of the features 116

3.3.10 “Nursery”: Minimum group accuracy (10-fold CV) on all possible subsets of the features 117

4.3.1 Selected features from the first result on the spectra. (a) -SAA samples, (b) +SAA samples. 152

4.3.2 Selected features from the second result on the spectra. (a) -SAA samples, (b) +SAA samples. 152

4.4.1 Example of base pairs on DNA. 155

4.4.2 Example of patterns and their reverse complements on DNA. 156

SUMMARY

In this dissertation we study the feature selection and classification problems and apply our methods to real-world medical and biological data sets for disease diagnosis.

Classification is an important problem in disease diagnosis to distinguish patients from normal population. DAMIP (discriminant analysis – mixed integer program) was shown to be a good classification model, which can directly handle multigroup problems, enforce misclassification limits, and provide reserved judgement region. However, DAMIP is NP-hard and presents computational challenges. Feature selection is important in classification to improve the prediction performance, prevent over-fitting, or facilitate data understanding. However, this combinatorial problem becomes intractable when the number of features is large.

In this dissertation, we propose a modified particle swarm optimization (PSO), a heuristic method, to solve the feature selection problem, and we study its parameter selection in our applications. We derive theories and exact algorithms to solve the two-group DAMIP in polynomial time. We also propose a heuristic algorithm to solve the multigroup DAMIP. Computational studies on simulated data and data from UCI machine learning repository show that the proposed algorithm performs very well. The polynomial solution time of the heuristic method allows us to solve DAMIP repeatedly within the feature selection procedure.

We apply the PSO/DAMIP classification framework to several real-life medical

and biological prediction problems. (1) Alzheimer's disease: We use data from several neuropsychological tests to discriminate subjects of Alzheimer's disease, subjects of mild cognitive impairment, and control groups. (2) Cardiovascular disease: We use traditional risk factors and novel oxidative stress biomarkers to predict subjects who are at high or low risk of cardiovascular disease, in which the risk is measured by the thickness of the carotid intima-media or/and the flow-mediated vasodilation. (3) Sulfur amino acid (SAA) intake: We use ^{1}H NMR spectral data of human plasma to classify plasma samples obtained with low SAA intake or high SAA intake. This shows that our method helps for metabolomics study. (4) CpG islands for lung cancer: We identify a large number of sequence patterns (in the order of millions), search candidate patterns from DNA sequences in CpG islands, and look for patterns which can discriminate methylation-prone and methylation-resistant (or in addition, methylation-sporadic) sequences, which relate to early lung cancer prediction.

CHAPTER I

CLASSIFICATION

This chapter introduces the classification problem, summarizes the mathematical-programming-based classification methods, and discusses Anderson's model and its related mixed integer programming model.

1.1 Introduction

The goal of classification is to predict the group of an observation from its features. We use the famous iris data set [36, 37] as an example to describe the classification problem. In this data set there are three types of irises: setosa, versicolour, and virginica. Regardless of the type, each iris sample is measured by (1) sepal length in cm (centimeter), (2) sepal width in cm, (3) petal length in cm, and (4) petal width in cm. Here is part of the data.

Iris-setosa	5.1	3.5	1.4	0.2
Iris-setosa	4.9	3.0	1.4	0.2
	⋮			
Iris-versicolor	7.0	3.2	4.7	1.4
Iris-versicolor	6.4	3.2	4.5	1.5
	⋮			
Iris-virginica	6.3	3.3	6.0	2.5
Iris-virginica	5.8	2.7	5.1	1.9
	⋮			

Each iris sample is an *observation*, the type of the irises is the *group*, and the measures of the iris sample are the *features*. Given observations in which the corresponding

groups are known, the goal is to find a function to "predict" the group from the features, and we want the predicted group to match the real group as likely as possible. The data used to train the predictive function is called the *training data*. We want the function to be predictive not only for the training data but also for data which are not used for training, or *testing data*.

We introduce the notations used in the dissertation. Suppose in the data we have n observations from K groups with m features. Let $\mathcal{G} = \{1, 2, \cdots, K\}$ be the set of indices of the groups, $\mathcal{O} = \{1, 2, \cdots, n\}$ be the set of indices of the observations, and $\mathcal{F} = \{1, 2, \cdots, m\}$ be the set of indices of the features. Also, let $\mathcal{O}_k$, $k \in \mathcal{G}$ and $\mathcal{O}_k \subseteq \mathcal{O}$, be the set of indices of observations which belong to group k. Moreover, let $\mathcal{F}_j$, $j \in \mathcal{F}$, be the domain of the jth feature, which could be the space of real, integer, or binary values. The ith observation, $i \in \mathcal{O}$, is represented as $(y_i, \boldsymbol{x}_i) = (y_i, x_{i1}, \cdots, x_{im}) \in \mathcal{G} \times \mathcal{F}_1 \times \cdots \times \mathcal{F}_m$, where y_i is the group of observation i and $(x_{i1}, \cdots, x_{im})$ is the feature vector of observation i. In the classification problem, we want to find a function $f : (\mathcal{F}_1 \times \cdots \times \mathcal{F}_m) \to \mathcal{G}$ so that we can obtain the predicted group from the features. This function is sometimes called the *decision rule*. Different classification models or methods refer to different forms of decision rules.

The classification methods can be parametric or nonparametric. In parametric methods, data are assumed to follow some parametric distribution, while in nonparametric methods, no distribution assumption is made. Examples of parametric methods include linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA); examples of nonparametric methods include support vector machine and k-nearest-neighbor algorithm.

1.2 Classification via Mathematical Programming

Mathematical-programming-based classification methods emerge in the 1960's, gain popularity in the 1980's, and have grown drastically ever since. Most mathematical programming approaches are nonparametric—cited as an advantage when contaminated data sets are analyzed [144]. Most of the literature on mathematical programming methods is concerned with finding hyperplanes in the feature space to separate data from different groups, in which mathematical programming is used to determine the coefficients of the hyperplanes.

1.2.1 Linear Programming Classification Models

The use of linear programming (LP) to determine the coefficients of linear discriminant functions has been widely studied [38, 57, 64, 96]. The methods determine the coefficients for different objectives, including minimizing the sum of distances to the separating hyperplane of the observations, minimizing the maximum distance to the hyperplane of the observations, maximizing some measures of goodness of fit, and so on.

1.2.1.1 Two-group Classification

One of the earliest LP classification models is proposed by Mangasarian [96], which constructs a hyperplane to separate data from two groups. Separation by a nonlinear surface using LP is also proposed when the surface parameters appear linearly. Two sets of points may be inseparable by one hyperplane or surface through a single-step LP approach, but they can be strictly separated by more hyperplanes or surfaces via a multi-step LP approach [97]. In [97] real problems with up to 117 data points, 10 features, and 3 groups are solved. The 3-group separation is achieved by separating group 1 from groups 2 and 3, and then group 2 from group 3.

Studies of LP models for the discriminant problem in the early 1980's are carried out by Hand [60], Freed and Glover [38, 39], and Bajgier and Hill [5]. Three LP models for the two-group classification problem, including minimizing the sum of deviations (MSD), minimizing the maximum deviation (MMD), and minimizing the sum of interior distances (MSID) are proposed. Freed and Glover [40] provide computational studies of these models where the test conditions involve normal and non-normal populations.

MSD (Minimizing the sum of deviations)

$$
\begin{aligned}
\min \quad & \sum_{i \in \mathcal{O}} d_i & \\
\text{s.t.} \quad & w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j - d_i \leq 0 & \forall\ i \in \mathcal{O}_1 \\
& w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j + d_i \geq 0 & \forall\ i \in \mathcal{O}_2 \\
& w_j \text{ urs} & \forall\ j \in \{0\} \cup \mathcal{F} \\
& d_i \geq 0 & \forall\ i \in \mathcal{O}
\end{aligned}
$$

Note that *urs* is the abbreviation of *unrestricted in sign*.

MMD (Minimizing the maximum deviation)

$$
\begin{aligned}
\min \quad & d \\
\text{s.t.} \quad & w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j - d \leq 0 && \forall\ i \in \mathcal{O}_1 \\
& w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j + d \geq 0 && \forall\ i \in \mathcal{O}_2 \\
& w_j \text{ urs} && \forall\ j \in \{0\} \cup \mathcal{F} \\
& d \geq 0
\end{aligned}
$$

MSID (Minimizing the sum of interior distances)

$$
\begin{aligned}
\min \quad & pd - \sum_{i \in \mathcal{O}} e_i \\
\text{s.t.} \quad & w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j - d + e_i \leq 0 && \forall\ i \in \mathcal{O}_1 \\
& w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j + d - e_i \geq 0 && \forall\ i \in \mathcal{O}_2 \\
& w_j \text{ urs} && \forall\ j \in \{0\} \cup \mathcal{F} \\
& d \geq 0 \\
& e_i \geq 0 && \forall\ i \in \mathcal{O}
\end{aligned}
$$

where p is a weight constant.

The objective function of the MSD model is the L_1-norm distance while the objective function of MMD is the L_∞-norm distance. They are special cases of L_p-norm classification [64, 145].

In some models the constant term of the hyperplane is a fixed number instead of a decision variable. The model MSD^0 shown below is an example in which the cut-off

score b replaces w_0 in the formulation. The same replacement can apply to other formulations.

MSD^0 (Minimizing the sum of deviations with constant cut-off score)

$$
\begin{aligned}
\min \quad & \sum_{i \in \mathcal{O}} d_i & \\
\text{s.t.} \quad & \sum_{j \in \mathcal{F}} x_{ij} w_j - d_i \leq b & \forall\ i \in \mathcal{O}_1 \\
& \sum_{j \in \mathcal{F}} x_{ij} w_j + d_i \geq b & \forall\ i \in \mathcal{O}_2 \\
& w_j \text{ urs} & \forall\ j \in \mathcal{F} \\
& d_i \geq 0 & \forall\ i \in \mathcal{O}
\end{aligned}
$$

A gap can be introduced between the two regions determined by the separating hyperplane to prevent degenerate solutions. Take MSD as an example, the separation constraints become

$$
\begin{aligned}
& w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j - d_i \leq -\epsilon & \forall\ i \in \mathcal{O}_1 \\
& w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j + d_i \geq \epsilon & \forall\ i \in \mathcal{O}_2.
\end{aligned}
$$

The small number ϵ can be normalized to 1.

Besides introducing a gap, another normalization approach is introducing normalization constraints such as $\sum_{j=0}^{m} w_j = 1$ or $\sum_{j=1}^{m} w_j = 1$ into the LP models to avoid unbounded or trivial solutions.

Glover et al. [56] generalize the MSID model and give the hybrid model.

Hybrid model

$$
\begin{aligned}
\min \quad & pd + \sum_{i \in \mathcal{O}} p_i d_i - qe - \sum_{i \in \mathcal{O}} q_i e_i & \\
\text{s.t.} \quad & w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j - d - d_i + e + e_i = 0 & \forall\ i \in \mathcal{O}_1 \\
& w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j + d + d_i - e - e_i = 0 & \forall\ i \in \mathcal{O}_2 \\
& w_j \text{ urs} & \forall\ j \in \{0\} \cup \mathcal{F} \\
& d, e \geq 0 & \\
& d_i, e_i \geq 0 & \forall\ i \in \mathcal{O}
\end{aligned}
$$

where p, p_i, q, q_i are the cost for different deviations. Including different combinations of deviation terms in the objective function then leads to variant models.

Joachimsthaler and Stam [64] review and summarize LP formulations applied to two-group classification problems in discriminant analysis, including MSD, MMD, MSID, and the hybrid model. They summarize the performance of the LP methods together with the traditional classification methods such as Fisher's linear discriminant function (LDF) [36], Smith's quadratic discriminant function (QDF) [142], and a logistic discriminant method. In their review, MSD sometimes but not uniformly improves classification accuracy, compared with traditional methods. On the other hand, MMD is found to be inferior to MSD. Erenguc and Koehler [30] present a unified survey of LP models and their experimental results, in which the LP models include several versions of MSD, MMD, MSID, and hybrid models. Rubin [133] provides experimental results of comparing these LP models with LDF and QDF. He concludes that QDF performed best when the data follow normal distributions and that QDF can be the benchmark when seeking situations for advantageous LP methods. In summary, the above review papers [30, 64, 133] describe previous work on LP classification models and their comparison with traditional methods. However, it is

difficult to make definitive statements about conditions under which an LP model is superior to the others, as stated in [144].

Stam and Ungar [148] introduce a software package RAGNU, a utility program in conjunction with the LINDO optimization software, for solving two-group classification problems using LP-based methods. LP formulations such as MSD, MMD, MSID, hybrid models and their variants are contained in the package.

There are some difficulties in LP-based formulations—some models can result in unbounded, trivial, or unacceptable solutions [109, 41], but possible remedies are proposed. Koehler [73, 74, 75] and Xiao [162, 163] characterize the conditions of unacceptable solutions in two-group LP discriminant models, including MSD, MMD, MISD, the hybrid model, and their variants. Glover [55] proposes the normalization constraint, $\sum_{j\in\mathcal{F}}(-|\mathcal{O}_2|\sum_{i\in\mathcal{O}_1}x_{ij}+|\mathcal{O}_1|\sum_{i\in\mathcal{O}_2}x_{ij})w_j=1$, which is more effective and reliable. Rubin [134] examines the separation failure for two-group models and suggests to apply the models twice, reversing the group designations at the second time. Xiao and Feng [164] propose a regularization method to avoid multiple solutions in LP discriminant analysis by adding the term $\epsilon\sum_{j\in\mathcal{F}}w_j^2$ in the objective functions.

Bennett and Mangasarian [9] propose the model of minimizing the average of the deviations, which is called robust linear programming.

RLP (Robust linear programming)

$$
\begin{aligned}
\min \quad & \frac{1}{|\mathcal{O}_1|}\sum_{i\in\mathcal{O}_1} d_i + \frac{1}{|\mathcal{O}_2|}\sum_{i\in\mathcal{O}_2} d_i & \\
\text{s.t.} \quad & w_0 + \sum_{j\in\mathcal{F}} x_{ij}w_j - d_i \le -1 & \forall\ i \in \mathcal{O}_1 \\
& w_0 + \sum_{j\in\mathcal{F}} x_{ij}w_j + d_i \ge 1 & \forall\ i \in \mathcal{O}_2 \\
& w_j \text{ urs} & \forall\ j \in \{0\}\cup\mathcal{F} \\
& d_i \ge 0 & \forall\ i \in \mathcal{O}
\end{aligned}
$$

It is shown that this model gives the null solution $w_1 = \cdots = w_m = 0$ if and only if $\frac{1}{|\mathcal{O}_1|}\sum_{i\in\mathcal{O}_1} x_{ij} = \frac{1}{|\mathcal{O}_2|}\sum_{i\in\mathcal{O}_2} x_{ij}$ for all j, in which case the solution $w_1 = \cdots = w_m = 0$ is guaranteed to be not unique. Data of different diseases are tested by the proposed classification methods, as in most of Mangasarian's papers.

Mangasarian et al. [108] describe two applications of LP models in the field of breast cancer research, one in diagnosis and the other in prognosis. The first application is to discriminate benign from malignant breast lumps and the second one is to predict when breast cancer is likely to recur. Both of them work successfully in clinical practice. The RLP model [9] together with the multisurface method tree algorithm (MSMT) [8] is used in the diagnostic system.

Duarte Silva and Stam [140] include the second-order (i.e., quadratic and cross-product) terms of the feature values in the LP-based models such as MSD and hybrid models and compare them with linear models, LDF, and QDF. The results of the simulation experiments show that the methods which include second-order terms perform much better than first-order methods, given that the data substantially violate the multivariate normality assumption. Wanarat and Pavur [160] investigate the effect

of the inclusion of the second-order terms in the MSD, MIP, and hybrid models when sample size is small to moderate. However, the simulation study shows that second-order terms may not always improve the performance of a first-order LP model even with data configurations that are more appropriately classified by QDF. Another result of the simulation study is that inclusion of the cross-product terms may hurt the model's accuracy, while omission of these terms causes the model to be not invariant with respect to a nonsingular transformation of the data.

Pavur [121] studies the effect of the position of the contaminated normal data in the two-group classification problem. The methods for comparison in their study include MSD, MM (described in Section 1.2.2), LDF, QDF, and nearest neighbor models. The nontraditional methods such as LP models have potential for outperforming the standard parametric procedures when non-normality is present, but this study shows that no single model is consistently superior in all cases.

Asparoukhov and Stam [4] propose LP and MIP models to solve the two-group classification problem where the features are binary. In this case the training data can be partitioned into multinomial cells, allowing for a substantial reduction in the number of variables and constraints. The proposed models not only have the usual geometric interpretation, but also possess a strong probabilistic foundation. Let s be the index of the cells, n_{1s}, n_{2s} be the number of data points in cell s from groups 1 and 2, respectively, and $(b_{s1}, \cdots, b_{sm})$ be the binary digits representing cell s. The model shown below is the LP model of minimizing the sum of deviations for two-group classification with binary features.

Cell convensional MSD

$$
\begin{aligned}
\min \quad & \sum_{s:\ n_{1s}+n_{2s}>0} (n_{1s}d_{1s} + n_{2s}d_{2s}) & \\
\text{s.t.} \quad & w_0 + \sum_{j\in\mathcal{F}} b_{sj}w_j - d_{1s} \le 0 & \forall\ s:\ n_{1s} > 0 \\
& w_0 + \sum_{j\in\mathcal{F}} b_{sj}w_j + d_{2s} > 0 & \forall\ s:\ n_{2s} > 0 \\
& w_j \text{ urs} & \forall\ j \in \{0\} \cup \mathcal{F} \\
& d_{1s}, d_{2s} \ge 0 & \forall\ s
\end{aligned}
$$

Binary features are usually found in medical diagnoses data. In this study three real data sets about disease discrimination are tested: developing postoperative pulmonary embolism or not, having dissecting aneurysm or other diseases, and suffering from posttraumatic epilepsy or not. In these data sets the MIP model for binary features (BMIP), which will be described later, performs better than other LP models or traditional methods.

1.2.1.2 Multigroup Classification

Freed and Glover [39] extend the LP classification models from two-group to multigroup problems. One formulation which uses a single discriminant function is given below.

$$
\begin{aligned}
\min \quad & \sum_{k=1}^{K-1} c_k \alpha_k & \\
\text{s.t.} \quad & \sum_{j \in \mathcal{F}} x_{ij} w_j \leq U_k & \forall\ i \in \mathcal{O}_k,\ k \in \mathcal{G} \\
& \sum_{j \in \mathcal{F}} x_{ij} w_j \geq L_k & \forall\ i \in \mathcal{O}_k,\ k \in \mathcal{G} \\
& U_k + \epsilon \leq L_{k+1} + \alpha_k & k = 1, \cdots, K-1 \\
& w_j \text{ urs} & \forall\ j \in \mathcal{F} \\
& U_k, L_k \text{ urs} & \forall\ k \in \mathcal{G} \\
& \alpha_k \text{ urs} & k = 1, \cdots, K-1
\end{aligned}
$$

where the number ϵ could be normalized to be 1, and c_k is the misclassification cost. However, single function classification is not as flexible and general as multiple function classification. Another extension from the two-group case to multigroup in [39] is to solve two-group LP models for all pairs of groups and determine classification rules based on these solutions. However, in some cases the group assignment is not clear and the resulting classification scheme may be sub-optimal [144].

For the multigroup discrimination problem, Bennett and Mangasarian [10] define the piecewise-linear separability of data from K groups as the following: The data from K groups are piecewise-linear separable if and only if there exist $(w_0^k, w_1^k, \cdots, w_m^k) \in \mathbb{R}^{m+1},\ \forall k \in \mathcal{G}$, such that $w_0^h + \sum_{j \in \mathcal{F}} x_{ij} w_j^h \geq w_0^k + \sum_{j \in \mathcal{F}} x_{ij} w_j^k + 1,\ \forall i \in \mathcal{O}_h, h, k \in \mathcal{G}, k \neq h$. The following LP will generate a piecewise-linear separation for the K groups if one exists, otherwise it will generate an error-minimizing separation.

$$
\begin{aligned}
\min \quad & \sum_{h\in\mathcal{G}} \sum_{k\in\mathcal{G},k\neq h} \frac{1}{|\mathcal{O}_h|} \sum_{i\in\mathcal{O}_h} d_i^{hk} \\
\text{s.t.} \quad & d_i^{hk} \geq -(w_0^h + \sum_{j\in\mathcal{F}} x_{ij} w_j^h) + (w_0^k + \sum_{j\in\mathcal{F}} x_{ij} w_j^k) + 1 \quad \forall\ i \in \mathcal{O}_h,\ h,k \in \mathcal{G},\ k \neq h \\
& w_j^k \text{ urs} \qquad\qquad \forall\ j \in \{0\} \cup \mathcal{F},\ k \in \mathcal{G} \\
& d_i^{hk} \geq 0 \qquad\qquad \forall\ i \in \mathcal{O}_h,\ h,k \in \mathcal{G},\ k \neq h
\end{aligned}
$$

The method is tested in three data sets. It performs pretty well in two of the data sets which are totally (or almost totally) piecewise-linear separable. The classification result is not good in the third data set, which is inherently more difficult. However, by combining the multisurface method tree algorithm (MSMT) [8], the performance improves.

Gochet et al. [57] introduce an LP model for the general multigroup classification problem. The method separates the data with several hyperplanes by sequentially solving LP's. The vectors $\boldsymbol{w}^k = (w_0^k, w_1^k, \cdots, w_m^k)$, $k \in \mathcal{G}$, are estimated for the classification decision rule. The rule is to classify an observation i into group s where $s = \arg\max_k \{w_0^k + \sum_{j\in\mathcal{F}} x_{ij} w_j^k\}$.

Given that observation i is from group h, denote the goodness of fit for observation i with respect to group k by

$G_{hk}^i(\boldsymbol{w}^h, \boldsymbol{w}^k) = [(w_0^h + \sum_{j\in\mathcal{F}} x_{ij} w_j^h) - (w_0^k + \sum_{j\in\mathcal{F}} x_{ij} w_j^k)]^+$, where $[a]^+ = \max\{0, a\}$.

Likewise, denote the badness of fit for observation i with respect to group k by

$B_{hk}^i(\boldsymbol{w}^h, \boldsymbol{w}^k) = [(w_0^h + \sum_{j\in\mathcal{F}} x_{ij} w_j^h) - (w_0^k + \sum_{j\in\mathcal{F}} x_{ij} w_j^k)]^-$, where $[a]^- = -\min\{0, a\}$.

The total goodness of fit and total badness of fit are then defined as

$G(\boldsymbol{w}) = G(\boldsymbol{w}^1, \cdots, \boldsymbol{w}^K) = \sum_{h\in\mathcal{G}} \sum_{k\in\mathcal{G}, k\neq h} \sum_{i\in\mathcal{O}_h} G^i_{hk}(\boldsymbol{w}^h, \boldsymbol{w}^k)$

$B(\boldsymbol{w}) = B(\boldsymbol{w}^1, \cdots, \boldsymbol{w}^K) = \sum_{h\in\mathcal{G}} \sum_{k\in\mathcal{G}, k\neq h} \sum_{i\in\mathcal{O}_h} B^i_{hk}(\boldsymbol{w}^h, \boldsymbol{w}^k)$

The LP is to minimize the total badness of fit, subject to a normalization equation, in which $q > 0$.

$$\begin{aligned} \min \quad & B(\boldsymbol{w}) \\ \text{s.t.} \quad & G(\boldsymbol{w}) - B(\boldsymbol{w}) = q \\ & \boldsymbol{w} \text{ urs} \end{aligned}$$

By expanding $G(\boldsymbol{w})$ and $B(\boldsymbol{w})$ and substituting $G^i_{hk}(\boldsymbol{w}^h, \boldsymbol{w}^k)$ and $B^i_{hk}(\boldsymbol{w}^h, \boldsymbol{w}^k)$ by γ^i_{hk} and β^i_{hk} respectively, the LP becomes

$$\begin{aligned} \min \quad & \sum_{h\in\mathcal{G}} \sum_{k\in\mathcal{G}, k\neq h} \sum_{i\in\mathcal{O}_h} \beta^i_{hk} \\ \text{s.t.} \quad & (w^h_0 + \sum_{j\in\mathcal{F}} x_{ij} w^h_j) - (w^k_0 + \sum_{j\in\mathcal{F}} x_{ij} w^k_j) = \gamma^i_{hk} - \beta^i_{hk} \quad \forall\ i \in \mathcal{O}_h,\ h, k \in \mathcal{G},\ k \neq h \\ & \sum_{h\in\mathcal{G}} \sum_{k\in\mathcal{G}, k\neq h} \sum_{i\in\mathcal{O}_h} (\gamma^i_{hk} - \beta^i_{hk}) = q \\ & w^k_j \text{ urs} \qquad \forall\ j \in \{0\} \cup \mathcal{F},\ k \in \mathcal{G} \\ & \gamma^i_{hk}, \beta^i_{hk} \geq 0 \qquad \forall\ i \in \mathcal{O}_h,\ h, k \in \mathcal{G},\ k \neq h \end{aligned}$$

The classification results for two real data sets show that this model can compete with LDF and k-nearest neighbor method.

1.2.2 Mixed Integer Programming Classification Models

While linear programming offers a polynomial-time computational guarantee, mixed integer programming (MIP) allows more flexibility in modeling misclassified observations and/or misclassification costs.

1.2.2.1 Two-group Classification

In the two-group classification problem, binary variables can be used in the formulation to track and minimize the exact number of misclassifications. Such an objective function is also considered as the L_0-norm criterion [144].

MM (Minimizing the number of misclassifications)

$$
\begin{aligned}
\min \quad & \sum_{i \in \mathcal{O}} z_i & \\
\text{s.t.} \quad & w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j \leq M z_i & \forall\ i \in \mathcal{O}_1 \\
& w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j \geq -M z_i & \forall\ i \in \mathcal{O}_2 \\
& w_j \text{ urs} & \forall\ j \in \{0\} \cup \mathcal{F} \\
& z_i \in \{0, 1\} & \forall\ i \in \mathcal{O}
\end{aligned}
$$

$(w_0, w_1, \cdots, w_m)$ is required to be a nonzero vector to prevent the trivial solution.

In this MIP formulation the objective function could include the deviation terms, such as those in the hybrid models, as well as the number of misclassifications [5]; or it could represent expected cost of misclassification [6, 1, 141, 135]. There are some variant versions of the basic model.

Stam and Joachimsthaler [146] study the classification performance of MM and compare it with MSD, LDF, and QDF. In some cases the MM model performs better, but in some cases it does not. MIP formulations are in the review studies of Joachimsthaler and Stam [64] and Erenguc and Koehler [30], and contained in the software developed by Stam and Ungar [148]. Computational experiments show that the MIP model performs better when the group overlap is higher [64, 146], although it is still not easy to reach general conclusions [144].

Since the MIP model is $\mathcal{NP}$-hard, exact algorithms and heuristics are proposed to solve it efficiently. Koehler and Erenguc [76] develop a procedure to solve MM in which the condition of nonzero w is replaced by the requirement of at least one violation of the constraints $w_0 + \sum_j x_{ij} w_j \leq 0$ for $i \in G_1$ or $w_0 + \sum_j x_{ij} w_j \geq 0$ for $i \in G_2$. Banks and Abad [6] solve the MIP of minimizing the expected cost of misclassification by an LP-based algorithm. Abad and Banks [1] develop three heuristic procedures to the problem of minimizing the expected cost of misclassification. They also include the interaction terms of the features in the data and apply the heuristics [7]. Duarte Silva and Stam [141] introduce a divide and conquer algorithm for the classification problem of minimizing the misclassification cost by solving MIP and LP subproblems. Rubin [135] solves the same problem by using a decomposition approach and tests this procedure on some data sets, including two breast cancer data sets. Yanev and Balev [165] propose exact and heuristic algorithms for solving MM, which are based on some specific properties of the vertices of a polyhedral set neatly connected with the model.

For the two-group classification problem where the features are binary, Asparoukhov and Stam [4] propose LP and MIP models which partition the data into multinomial cells and result in fewer number of variables and constraints. Let s be the index of the cells, n_{1s}, n_{2s} be the number of data points in cell s from groups 1 and 2, respectively,

and $(b_{s1}, \cdots, b_{sm})$ be the binary digits representing cell s. Below is the MIP model for binary features (BMIP), which performs better than other LP models or traditional methods in three real data sets [4].

BMIP

$$
\begin{aligned}
\min \quad & \sum_{s:\ n_{1s}+n_{2s}>0} \{|n_{1s}-n_{2s}|z_s + \min(n_{1s}, n_{2s})\} & \\
\text{s.t.} \quad & w_0 + \sum_{j\in\mathcal{F}} b_{sj} w_j \le M z_s & \forall\ s:\ n_{1s} \ge n_{2s};\ n_{1s} > 0 \\
& w_0 + \sum_{j\in\mathcal{F}} b_{sj} w_j > -M z_s & \forall\ s:\ n_{1s} < n_{2s} \\
& w_j \text{ urs} & \forall\ j \in \{0\} \cup \mathcal{F} \\
& z_s \in \{0,1\} & \forall\ s:\ n_{1s} + n_{2s} > 0
\end{aligned}
$$

Pavur et al. [123] include different secondary goals in the model MM and compare their misclassification rates. A new secondary goal is proposed, which maximizes the difference between the means of the discriminant scores of the two groups, represented by the decision variable δ. In this model the term $-\delta$ is added to the minimization objective function as a secondary goal with a constant multiplier while the constraint $\sum_{j\in\mathcal{F}} \bar{x}_j^{(2)} w_j - \sum_{j\in\mathcal{F}} \bar{x}_j^{(1)} w_j \ge \delta$ is included, where $\bar{x}_j^{(k)} = \frac{1}{|\mathcal{O}_k|} \sum_{i\in\mathcal{O}_k} x_{ij},\ \forall j \in \mathcal{F},\ k = 1, 2$. The results of simulation study show that an MIP model with the proposed secondary goal has better performance than other studied models.

Glen [49] proposes IP techniques for normalization in the two-group discriminant analysis models. One technique is to add the constraint $\sum_{j\in\mathcal{F}} |w_j| = 1$. In the proposed model, w_j, $j \in \mathcal{F}$ is represented by $w_j = w_j^+ - w_j^-$, where $w_j^+, w_j^- \ge 0$, and binary variables δ_j and γ_j are defined such that $\delta_j = 1 \Leftrightarrow w_j^+ \ge \epsilon$ and $\gamma_j = 1 \Leftrightarrow w_j^- \ge \epsilon$. The IP normalization technique is applied to MSD and MMD, and the MSD version

is presented below.

MSD – with IP normalization

$$
\begin{aligned}
\min \quad & \sum_{i \in \mathcal{O}} d_i \\
\text{s.t.} \quad & w_0 + \sum_{j \in \mathcal{F}} x_{ij}(w_j^+ - w_j^-) - d_i \leq 0 && \forall\ i \in \mathcal{O}_1 \\
& w_0 + \sum_{j \in \mathcal{F}} x_{ij}(w_j^+ - w_j^-) + d_i \geq 0 && \forall\ i \in \mathcal{O}_2 \\
& \sum_{j \in \mathcal{F}} (w_j^+ + w_j^-) = 1 \\
& w_j^+ - \epsilon \delta_j \geq 0 && \forall\ j \in \mathcal{F} \\
& w_j^+ - \delta_j \leq 0 && \forall\ j \in \mathcal{F} \\
& w_j^- - \epsilon \gamma_j \geq 0 && \forall\ j \in \mathcal{F} \\
& w_j^- - \gamma_j \leq 0 && \forall\ j \in \mathcal{F} \\
& \delta_j + \gamma_j \leq 1 && \forall\ j \in \mathcal{F} \\
& w_0 \text{ urs} \\
& w_j^+, w_j^- \geq 0 && \forall\ j \in \mathcal{F} \\
& d_i \geq 0 && \forall\ i \in \mathcal{O} \\
& \delta_j, \gamma_j \in \{0, 1\} && \forall\ j \in \mathcal{F}
\end{aligned}
$$

The variable coefficients of the discriminant function generated by the models are invariant under origin shifts. The proposed models are validated using two data sets from [56, 109]. The models are also extended for feature selection by adding the constraint $\sum_{j \in \mathcal{F}} (\delta_j + \gamma_j) = p$, which allows only a constant number, p, of features to be used for classification.

Other than the objectives MSD and MMD, the normalization technique (i.e., $\sum_{j\in\mathcal{F}}|w_j|=1$) and the feature selection technique (i.e., $\sum_{j\in\mathcal{F}}(\delta_j+\gamma_j)=p$) are also applied to the objective which maximizes classification accuracy (MCA), or, equivalently, minimizes the number of misclassifications [50].

MCA (Maximizing classification accuracy)

$$
\begin{aligned}
\max \quad & \sum_{i\in\mathcal{O}} z_i \\
\text{s.t.} \quad & w_0+\sum_{j\in\mathcal{F}} x_{ij}(w_j^+-w_j^-)+(M+\Delta)z_i \le M && \forall\, i\in\mathcal{O}_1 \\
& w_0+\sum_{j\in\mathcal{F}} x_{ij}(w_j^+-w_j^-)-(M+\Delta)z_i \ge -M && \forall\, i\in\mathcal{O}_2 \\
& \sum_{j\in\mathcal{F}}(w_j^++w_j^-)=1 \\
& w_j^+-\epsilon\delta_j\ge 0 && \forall\, j\in\mathcal{F} \\
& w_j^+-\delta_j\le 0 && \forall\, j\in\mathcal{F} \\
& w_j^--\epsilon\gamma_j\ge 0 && \forall\, j\in\mathcal{F} \\
& w_j^--\gamma_j\le 0 && \forall\, j\in\mathcal{F} \\
& \delta_j+\gamma_j\le 1 && \forall\, j\in\mathcal{F} \\
& \sum_{j\in\mathcal{F}}(\delta_j+\gamma_j)=p \\
& w_0 \text{ urs} \\
& w_j^+, w_j^-\ge 0 && \forall\, j\in\mathcal{F} \\
& z_i\in\{0,1\} && \forall\, i\in\mathcal{O} \\
& \delta_j,\gamma_j\in\{0,1\} && \forall\, j\in\mathcal{F}
\end{aligned}
$$

where Δ is a small positive number.

Furthermore, with this normalization technique and feature selection technique,

two-stage approaches are proposed in [50, 51]. Glen [54] also compares standard (i.e., one stage) MP-based classification methods with two-stage MP-based methods proposed by Stam and Ragsdale [147] and Sueyoshi [151, 152]. The idea of two-stage methods is to firstly identify the observations which are difficult to be classified and deal with these observations in the second stage. Integer variables are used for part of the observations in the second stage but not for all observations, so the MIP's are easier to solve. The results of the comparison indicate that "a single technique will not produce good linear classifiers under all data conditions."

Instead of linear discriminant functions obtained by MP-based models, Glen [53] proposes piecewise-linear models with objectives MCA and MSD. The computational results show that the MCA piecewise-linear model performs better than the standard MCA model.

Glen [52] developes MIP models which determine the thresholds for forming dichotomous variables as well as the discriminant function coefficient w_j's. For each continuous feature to be formed as a dichotomous feature, the model finds the threshold among possible thresholds while determining the separating hyperplane and optimizing the objective function such as minimizing the sum of deviations or minimizing the number of misclassifications. Computational results of a real data set and some simulated data sets show that the MSD model with dichotomous categorical variable formation can improve classification performance. The reason for the potential for performance improvement is that the generated linear discriminant function is a nonlinear function of the original variables.

1.2.2.2 Multigroup Classification

Gehrlein [47] proposes MIP formulations of minimizing the total number of misclassifications in the multigroup classification problem. He gives both a single function classification scheme and a multiple function classification scheme, as follows.

GSFC (General single function classification – minimizing the number of misclassifications)

$$
\begin{aligned}
\min \quad & \sum_{i \in \mathcal{O}} z_i & & \\
\text{s.t.} \quad & w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j - M z_i \leq U_k & & \forall\ i \in \mathcal{O}_k,\ k \in \mathcal{G} \\
& w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j + M z_i \geq L_k & & \forall\ i \in \mathcal{O}_k,\ k \in \mathcal{G} \\
& U_k - L_k \geq \delta' & & \forall\ k \in \mathcal{G} \\
& L_g - U_k + M y_{gk} \geq \delta & & \forall\ g, k \in \mathcal{G},\ g \neq k \\
& L_k - U_g + M y_{kg} \geq \delta & & \forall\ g, k \in \mathcal{G},\ g \neq k \\
& y_{gk} + y_{kg} = 1 & & \forall\ g, k \in \mathcal{G},\ g \neq k \\
& w_j \text{ urs} & & \forall\ j \in \{0\} \cup \mathcal{F} \\
& U_k, L_k \text{ urs} & & \forall\ k \in \mathcal{G} \\
& z_i \in \{0, 1\} & & \forall\ i \in \mathcal{O} \\
& y_{gk} \in \{0, 1\} & & \forall\ g, k \in \mathcal{G},\ g \neq k
\end{aligned}
$$

where U_k, L_k denote the upper and lower endpoints of the interval assigned to group k, and $y_{gk} = 1$ if the interval associated with group g precedes that with group k and $y_{gk} = 0$ otherwise. The constant δ' is the minimum width of an interval of a group and the constant δ is the minimum gap between adjacent intervals.

GMFC (General multiple function classification – minimizing the number of misclassifications)

$$
\begin{aligned}
\min \quad & \sum_{i \in \mathcal{O}} z_i \\
\text{s.t.} \quad & (w_0^h + \sum_{j \in \mathcal{F}} x_{ij} w_j^h) - (w_0^k + \sum_{j \in \mathcal{F}} x_{ij} w_j^k) + M z_i \geq \epsilon \quad \forall\ i \in \mathcal{O}_h,\ h, k \in \mathcal{G},\ k \neq h \\
& w_j^k \text{ urs} \qquad \forall\ j \in \{0\} \cup \mathcal{F},\ k \in \mathcal{G} \\
& z_i \in \{0, 1\} \qquad \forall\ i \in \mathcal{O}
\end{aligned}
$$

Both models work successfully on the iris data set provided by Fisher [36].

Pavur [120] solves the multigroup classification problem by sequentially solving GSFC in one dimension each time. Linear discriminant functions are generated by successively solving GSFC with the added constraints that all linear discriminants are uncorrelated to each other. According to simulation results, this procedure substantially improves the GSFC model and sometimes outperforms GMFC, LDF, or QDF.

To solve the three-group classification problem more efficiently, Loucopoulos and Pavur [93] make a slight modification on GSFC and propose the model MIP3G, which also minimizes the number of misclassifications. Compared with GSFC, MIP3G is also a single function classification model, but it reduces the possible group orderings from six to three in the formulation and thus becomes more efficient. Loucopoulos and Pavur [94] report the results of a simulation experiment on the performance of GMFC, MIG3G, LDF, and QDF for the three-group classification problem with small training samples. Second-order terms are also considered in the experiment. Simulation results show that GMFC and MIP3G can outperform the parametric procedures in some non-normal data sets and that the inclusion of second-order terms

can improve the performance of MIP3G in some data sets. Pavur and Loucopoulos [122] investigate the effect of the gap size in the MIP3G model for the three-group classification problem. A simulation study illustrates that for fairly separable data, or data with small sample sizes, a nonzero-gap model can improve the performance. A possible reason for this result is that the zero-gap model may over-fit the data.

1.2.3 Nonlinear Programming Classification Models

In the literature, nonlinear programming is mainly applied to two-group classification problems, therefore we focus on the two-group problems in this section.

Stam and Joachimsthaler [145] propose a class of nonlinear programming methods to solve the two-group classification problem under the L_p-norm objective criterion. This is an extension of MSD and MMD, for which the objectives are the L_1-norm and L_∞-norm, respectively.

Minimize the general L_p-norm distance

$$\begin{aligned}
\min \quad & \left(\sum_{i \in \mathcal{O}} d_i^p \right)^{1/p} & \\
\text{s.t.} \quad & \sum_{j \in \mathcal{F}} x_{ij} w_j - d_i \leq b & \forall\ i \in \mathcal{O}_1 \\
& \sum_{j \in \mathcal{F}} x_{ij} w_j + d_i \geq b & \forall\ i \in \mathcal{O}_2 \\
& w_j \text{ urs} & \forall\ j \in \mathcal{F} \\
& d_i \geq 0 & \forall\ i \in \mathcal{O}
\end{aligned}$$

Based on the computational results, the authors recommend to apply this model by using $1 \leq p \leq 3$ and $p = \infty$.

Mangasarian et al. [107] propose a nonconvex model for the two-group classification problem:

$$
\begin{aligned}
\min \quad & d_1 + d_2 \\
\text{s.t.} \quad & \sum_{j \in \mathcal{F}} x_{ij} w_j - d_1 \leq 0 && \forall\ i \in \mathcal{O}_1 \\
& \sum_{j \in \mathcal{F}} x_{ij} w_j + d_2 \geq 0 && \forall\ i \in \mathcal{O}_2 \\
& \max_{j \in \mathcal{F}} |w_j| = 1 \\
& w_j \text{ urs} && \forall\ j \in \mathcal{F} \\
& d_1, d_2 \text{ urs}
\end{aligned}
$$

This model can be solved in polynomial-time by solving $2m$ linear programs, which generate a sequence of parallel planes, resulting in a piecewise-linear nonconvex discriminant function. The model works successfully in clinical practice for the diagnosis of breast cancer.

Mangasarian [98] also formulates the problem of minimizing the number of misclassifications as a linear program with equilibrium constraints (LPEC) instead of the MIP model MM described in Section 1.2.2.

MM-LPEC (Minimizing the number of misclassifications – Linear program with equilibrium constraints)

$$
\begin{aligned}
\min \quad & \sum_{i \in \mathcal{O}} z_i & \\
\text{s.t.} \quad & w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j - d_i \leq -1 & \forall\ i \in \mathcal{O}_1 \\
& z_i (w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j - d_i + 1) = 0 & \forall\ i \in \mathcal{O}_1 \\
& w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j + d_i \geq 1 & \forall\ i \in \mathcal{O}_2 \\
& z_i (w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j + d_i - 1) = 0 & \forall\ i \in \mathcal{O}_2 \\
& d_i (1 - z_i) = 0 & \forall\ i \in \mathcal{O} \\
& 0 \leq z_i \leq 1 & \forall\ i \in \mathcal{O} \\
& d_i \geq 0 & \forall\ i \in \mathcal{O} \\
& w_j \text{ urs} & \forall\ j \in \{0\} \cup \mathcal{F}
\end{aligned}
$$

The general LPEC can be converted to an exact penalty problem with a quadratic objective and linear constraints. A stepless Frank-Wolfe-type algorithm is proposed for the penalty problem, terminating at a stationary point or a global solution. This method is called the parametric misclassification minimization (PMM) procedure, and numerical testing is included in [99].

To illustrate the next model, we first define the step function $s : \mathbb{R} \to \{0, 1\}$ as

$$
s(u) = \begin{cases} 1 & \text{if } u > 0 \\ 0 & \text{if } u \leq 0 \end{cases}
$$

The problem of minimizing the number of misclassifications is equivalent to

$$
\begin{aligned}
\min \quad & \sum_{i \in \mathcal{O}} s(d_i) & \\
\text{s.t.} \quad & w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j - d_i \leq -1 & \forall\ i \in \mathcal{O}_1 \\
& w_0 + \sum_{j \in \mathcal{F}} x_{ij} w_j + d_i \geq 1 & \forall\ i \in \mathcal{O}_2 \\
& w_j \text{ urs} & \forall\ j \in \{0\} \cup \mathcal{F} \\
& d_i \geq 0 & \forall\ i \in \mathcal{O}
\end{aligned}
$$

Mangasarian [99] proposes a simple concave approximation of the step function for nonnegative variables: $t(u, \alpha) = 1 - e^{-\alpha u}$, where $\alpha > 0, u \geq 0$. Let $\alpha > 0$ and approximate $s(d_i)$ by $t(d_i, \alpha)$. The problem then reduces to minimizing a smooth concave function bounded below on a nonempty polyhedron, which has a minimum at a vertex of the feasible region. A finite successive linearization algorithm (SLA) is proposed, terminating at a stationary point or a global solution. Numerical tests of SLA are done and compared with the PMM procedure described above. The results show that the much simpler SLA obtains a separation that is almost as good as PMM in considerably less computing time.

Chen and Mangasarian [22] define a hybrid misclassification minimization problem, which is more computationally tractable than the $\mathcal{NP}$-hard misclassification minimization problem, and a related algorithm. The basic idea of the hybrid approach is to obtain iteratively w_0 and $(w_1, \cdots, w_m)$ of the separating hyperplane: (1) For a fixed w_0, solve RLP (Bennett and Mangasarian [9]) to determine $(w_1, \cdots, w_m)$, and (2) for this $(w_1, \cdots, w_m)$, solve the one-dimensional misclassification minimization problem to determine w_0. Compared with RLP and PMM procedure, the hybrid method performs better and is much faster than PMM.

Mangasarian [100] proposes the model of minimizing the sum of arbitrary-norm distances of misclassified points to the separating hyperplane. For a general norm $||\cdot||$ on $\mathbb{R}^m$, the dual norm $||\cdot||'$ on $\mathbb{R}^m$ is defined as $||\boldsymbol{x}||' = \max_{||\boldsymbol{y}||=1} \boldsymbol{x}^T\boldsymbol{y}$. Define $[a]^+ = \max\{0, a\}$ and let $\boldsymbol{w} = (w_1, \cdots, w_m)$, the formulation is:

$$\begin{aligned} \min \quad & \sum_{i\in\mathcal{O}_1}[w_0 + \sum_{j\in\mathcal{F}} x_{ij}w_j]^+ + \sum_{i\in\mathcal{O}_2}[-w_0 - \sum_{j\in\mathcal{F}} x_{ij}w_j]^+ \\ \text{s.t.} \quad & ||\boldsymbol{w}||' = 1 \\ & w_0, \boldsymbol{w} \text{ urs} \end{aligned}$$

The problem is to minimize a convex function on a unit sphere. A related decision problem to this minimization problem is shown to be $\mathcal{NP}$-complete, except for $p = 1$. For a general p-norm, the minimization problem can be transformed via an exact penalty formulation to minimizing the sum of a convex function and a bilinear function on a convex set.

1.2.4 Support Vector Machine

A support vector machine is a type of mathematical programming approach (Vapnik [159]) originally for two-group classification problems. It has been widely studied and has become popular in many application fields in recent years. The introductory description of support vector machines (SVM) given here is summarized from the tutorial by Burges [21].

In this section, the domain of y_i is redefined to be consistent with SVM studies in published literature. That is, for the ith observation $(y_i, \boldsymbol{x}_i) = (y_i, x_{i1}, \cdots, x_{im})$, we have $y_i \in \{-1, +1\}$ instead of $y_i \in \mathcal{G} = \{1, 2\}$. Besides, let $\boldsymbol{w} = (w_1, \cdots, w_m)$.

In the two-group separable case, SVM finds a separating hyperplane $\boldsymbol{x}^T\boldsymbol{w} - b = 0$ in $\mathbb{R}^m$ by maximizing the margin between two groups, $2/||\boldsymbol{w}||$, or equivalently, minimizing $||\boldsymbol{w}||^2$.

SVM – separable case

$$\begin{aligned} \min \quad & \boldsymbol{w}^T\boldsymbol{w} \\ \text{s.t.} \quad & \boldsymbol{x}_i^T\boldsymbol{w} + b \geq +1 && \forall\ i:\ y_i = +1 \\ & \boldsymbol{x}_i^T\boldsymbol{w} + b \leq -1 && \forall\ i:\ y_i = -1 \\ & \boldsymbol{w}, b \text{ urs} \end{aligned}$$

This problem can be solved by solving its Wolfe dual problem:

$$\begin{aligned} \max \quad & \sum_{i\in\mathcal{O}} \alpha_i - \frac{1}{2}\sum_{i\in\mathcal{O}}\sum_{j\in\mathcal{O}} \alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j \\ \text{s.t.} \quad & \sum_{i\in\mathcal{O}} \alpha_i y_i = 0 \\ & \alpha_i \geq 0 && \forall\ i \in \mathcal{O} \end{aligned}$$

Here α_i is the Lagrange multiplier for observation i. The points $(x_{i1}, \cdots, x_{im})$ which satisfy $\alpha_i > 0$ are called the support vectors. The primal solution $\boldsymbol{w}$ is given by

$$\boldsymbol{w} = \sum_{i\in\mathcal{O}} \alpha_i y_i \boldsymbol{x}_i. \tag{1.2.1}$$

b can be computed by solving $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 = 0$ for any i with $\alpha_i > 0$.

In the non-separable case, slack variables ξ_i's are introduced to handle the errors. Let C be the penalty for the errors. The problem becomes

SVM – non-separable case

$$
\begin{aligned}
\min \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\left(\sum_{i\in\mathcal{O}}\xi_i\right)^k & \\
\text{s.t.} \quad & \boldsymbol{x}_i^T\boldsymbol{w} + b \geq +1 - \xi_i & \forall\ i:\ y_i = +1 \\
& \boldsymbol{x}_i^T\boldsymbol{w} + b \leq -1 + \xi_i & \forall\ i:\ y_i = -1 \\
& \boldsymbol{w}, b \text{ urs} & \\
& \xi_i \geq 0 & \forall\ i \in \mathcal{O}
\end{aligned}
$$

When k is chosen to be 1, neither the ξ_i's nor their Lagrange multipliers appear in the Wolfe dual problem:

$$
\begin{aligned}
\max \quad & \sum_{i\in\mathcal{O}}\alpha_i - \frac{1}{2}\sum_{i\in\mathcal{O}}\sum_{j\in\mathcal{O}}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j & \\
\text{s.t.} \quad & \sum_{i\in\mathcal{O}}\alpha_i y_i = 0 & \\
& 0 \leq \alpha_i \leq C & \forall\ i \in \mathcal{O}
\end{aligned}
$$

The data points can be separated nonlinearly by mapping the data into some higher dimensional space and applying linear SVM to the mapped data. Instead of knowing explicitly the mapping Φ, SVM needs only the dot products of two transformed data points $\Phi(\boldsymbol{x}_i)\cdot\Phi(\boldsymbol{x}_j)$. The kernel function K is introduced such that $K(\boldsymbol{x}_i,\boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i)\cdot\Phi(\boldsymbol{x}_j)$. Replacing $\boldsymbol{x}_i^T\boldsymbol{x}_j$ by $K(\boldsymbol{x}_i,\boldsymbol{x}_j)$ in the above problem, the separation becomes nonlinear while the problem to solve remains a quadratic program. To predict the group of a new data point $\boldsymbol{x}$ after training, the sign of the function $f(\boldsymbol{x})$ is computed to determine the group of $\boldsymbol{x}$:

$$
f(\boldsymbol{x}) = \sum_{i=1}^{N_s}\alpha_i y_i \Phi(\boldsymbol{s}_i)\cdot\Phi(\boldsymbol{x}) + b = \sum_{i=1}^{N_s}\alpha_i y_i K(\boldsymbol{s}_i,\boldsymbol{x}) + b
$$

where $\boldsymbol{s}_i$'s are the support vectors and N_s is the number of support vectors. Again the explicit form of $\Phi(\boldsymbol{x})$ is avoided.

Mangasarian provides a general mathematical programming framework for SVM, called generalized support vector machine or GSVM [101, 105]. Special cases can be derived from GSVM, including the standard SVM.

Many SVM-type methods have been developed by Mangasarian and other authors to solve huge-sized classification problems more efficiently. These methods include successive overrelaxation for SVM [104], proximal SVM [42, 44], smooth SVM [87], reduced SVM [86], Lagrangian SVM [106], incremental SVMs [43], and other methods [16, 103]. Mangasarian summarizes some of the developments in [102]. Examples of applications of SVM include breast cancer studies [88, 89] and genome research [95].

Hsu and Lin [61] compare different methods for multigroup classification using support vector machines. Three methods studied are based on applying two-group SVM several times: one-against-one, one-against-all, and directed acyclic graph (DAG) SVM. The other two methods studied are methods considering all groups at once with decomposition implementation. The experiment results show that the one-against-one and DAG methods are more suitable for practical use than the other methods. Lee et al. [85] propose a generic approach to multigroup problems with some theoretical properties, and the proposed method is well applied to microarray data for cancer classification and satellite radiance profiles for cloud classification.

1.3 Anderson's Model and DAMIP

This section introduces Anderson's model, a classification model which incorporates misclassification-limit constraints, and the model DAMIP (discriminant analysis – mixed integer program), whose solution gives the optimal decision rule of Anderson's model.

1.3.1 Anderson's Model

Anderson [3] proposes a classification model which maximizes the probability of correct classification subject to some limits of misclassification probability. This model is parametric—assuming data of each group follow certain distribution. Let π_k be the prior probability of group k and $f_k(\boldsymbol{x})$ be the value of the conditional probability density function for the data point $\boldsymbol{x} \in \mathbb{R}^m$ of group k, $k \in \mathcal{G}$. Also let $\alpha_{hk} \in (0,1), h, k \in \mathcal{G}, h \neq k$ be the predetermined limits of the misclassification probability that data of group h are misclassified to group k. The proposed model is to seek for a partition $\{R_0, R_1, \cdots, R_K\}$ of $\mathbb{R}^m$, where R_k, $k \in \mathcal{G} = \{1, \cdots, K\}$, is the region classified to group k and R_0 is the reserved judgement region, in which the group-assignment decision of data points is reserved.

Anderson's model

$$\begin{aligned} \max \quad & \sum_{k \in \mathcal{G}} \pi_k \int_{R_k} f_k(\boldsymbol{x}) d\boldsymbol{x} \\ \text{s.t.} \quad & \int_{R_k} f_h(\boldsymbol{x}) d\boldsymbol{x} \leq \alpha_{hk} \qquad \forall\ h, k \in \mathcal{G},\ h \neq k. \end{aligned}$$

Anderson shows that there exist nonnegative constants $\lambda_{hk},\ h, k \in \mathcal{G},\ h \neq k$, such

that the optimal decision rule is given by

$$R_k = \{\boldsymbol{x} \in \mathbb{R}^m : \ L_k(\boldsymbol{x}) = \max_{h \in \{0\} \cup \mathcal{G}} L_h(\boldsymbol{x})\}, \ k \in \{0\} \cup \mathcal{G}, \tag{1.3.1}$$

where

$$\begin{aligned} L_0(\boldsymbol{x}) &= 0 \\ L_k(\boldsymbol{x}) &= \pi_k f_k(\boldsymbol{x}) - \sum_{h \in \mathcal{G}, h \neq k} \lambda_{hk} f_h(\boldsymbol{x}), \quad k \in \mathcal{G}. \end{aligned} \tag{1.3.2}$$

This rule is called Anderson's rule.

However, the optimal λ's are difficult to find.

1.3.2 DAMIP

Gallagher et al. [45, 46] first propose mixed integer programming formulations, named DAMIP, for obtaining the optimal values of λ's in Anderson's rule. A nonlinear and a linear version of DAMIP are presented below. The binary variable u_{ki} indicates whether observation i is classified to group k or not. Recall that $y_i \in \mathcal{G}$ represents the group of observation i, the objective function (1.3.3) maximizes the total number of correctly-classified observations. Constraints (1.3.4) define $L_k(\boldsymbol{x})$ of Equation (1.3.2) in Anderson's rule, constraints (1.3.5) and (1.3.6) guarantee the correct value of u_{ki} based on (1.3.1), and constraints (1.3.7) model the misclassification limits. The linear version of DAMIP uses constraints (1.3.9)-(1.3.12) to model constraints (1.3.5) of the nonlinear version, in which the variable t_i achieves the value of $\max\{0, L_{ki} : k \in \mathcal{G}\}$. This (linear) version of DAMIP is based on [17], which is almost equivalent to nonlinear DAMIP except that DAMIP introduces a small value ϵ in its formulation to increase the stability of the classification rule derived by DAMIP, as seen in constraint (1.3.10) and (1.3.12).

Nonlinear DAMIP

$$\max \quad \sum_{i \in \mathcal{O}} u_{y_i i} \tag{1.3.3}$$

$$\text{s.t.} \quad L_{ki} = \pi_k f_k(\boldsymbol{x}_i) - \sum_{h \in \mathcal{G}, h \neq k} f_h(\boldsymbol{x}_i) \lambda_{hk} \qquad \forall\ i \in \mathcal{O},\ k \in \mathcal{G} \tag{1.3.4}$$

$$u_{ki} = \begin{cases} 1 & \text{if } k = \arg\max\{0, L_{hi} : h \in \mathcal{G}\} \\ 0 & \text{otherwise} \end{cases} \qquad \forall\ i \in \mathcal{O},\ k \in \{0\} \cup \mathcal{G} \tag{1.3.5}$$

$$\sum_{k \in \{0\} \cup \mathcal{G}} u_{ki} = 1 \qquad \forall\ i \in \mathcal{O} \tag{1.3.6}$$

$$\sum_{i:\ i \in \mathcal{O}_h} u_{ki} \leq \lfloor \alpha_{hk} n_h \rfloor \qquad \forall\ h, k \in \mathcal{G},\ h \neq k \tag{1.3.7}$$

$$u_{ki} \in \{0, 1\} \qquad \forall\ i \in \mathcal{O},\ k \in \{0\} \cup \mathcal{G}$$

$$L_{ki} \text{ urs} \qquad \forall\ i \in \mathcal{O},\ k \in \mathcal{G}$$

$$\lambda_{hk} \geq 0 \qquad \forall\ h, k \in \mathcal{G},\ h \neq k$$

DAMIP

$$\max \quad \sum_{i \in \mathcal{O}} u_{y_i i} \tag{1.3.8}$$

$$\text{s.t.} \quad L_{ki} = \pi_k f_k(\boldsymbol{x}_i) - \sum_{h \in \mathcal{G}, h \neq k} f_h(\boldsymbol{x}_i)\lambda_{hk} \qquad \forall\ i \in \mathcal{O},\ k \in \mathcal{G}$$

$$t_i - L_{ki} \leq M(1 - u_{ki}) \qquad \forall\ i \in \mathcal{O},\ k \in \mathcal{G} \tag{1.3.9}$$

$$t_i - L_{ki} \geq \epsilon(1 - u_{ki}) \qquad \forall\ i \in \mathcal{O},\ k \in \mathcal{G} \tag{1.3.10}$$

$$t_i \leq M(1 - u_{0i}) \qquad \forall\ i \in \mathcal{O} \tag{1.3.11}$$

$$t_i \geq \epsilon u_{ki} \qquad \forall\ i \in \mathcal{O},\ k \in \mathcal{G} \tag{1.3.12}$$

$$\sum_{k \in \{0\} \cup \mathcal{G}} u_{ki} = 1 \qquad \forall\ i \in \mathcal{O}$$

$$\sum_{i:\ i \in \mathcal{O}_h} u_{ki} \leq \lfloor \alpha_{hk} n_h \rfloor \qquad \forall\ h, k \in \mathcal{G},\ h \neq k$$

$$u_{ki} \in \{0, 1\} \qquad \forall\ i \in \mathcal{O},\ k \in \{0\} \cup \mathcal{G}$$

$$L_{ki} \text{ urs} \qquad \forall\ i \in \mathcal{O},\ k \in \mathcal{G}$$

$$t_i \geq 0 \qquad \forall\ i \in \mathcal{O}$$

$$\lambda_{hk} \geq 0 \qquad \forall\ h, k \in \mathcal{G},\ h \neq k$$

Brooks [17] and Brooks and Lee [19] show that DAMIP is polynomially solvable for $K = 2$ but is $\mathcal{NP}$-complete for a general K. Computational strategies in the branch and bound algorithm for solving DAMIP are provided, including cutting planes obtained by utilizing the conflict graph, a branching strategy incorporating information from the conflict graph and the implied cuts, and a heuristic used to generate integer feasible solutions. Computational results show that the strategies significantly improve the computational time.

Brooks and Lee [18] study the consistency property of DAMIP. Given some observations (i.e., samples of certain distributions), the classifier using the λ's obtained by solving DAMIP on these observations is strongly consistent to Anderson's rule—when the sample size goes to infinity, (1) the probability of correct allocation by DAMIP converges to that by Anderson's rule with probability one, and (2) the probabilities of misclassification of group h to group k are less than or equal to the predetermined limits with probability one. Furthermore, the consistency is universal—this applies to all possible distributions.

When there is one (or more) group whose size is relatively smaller than the others, the original objective function which maximizes the total number of correctly-classified observations (or equivalently, the overall accuracy) might give a solution with good overall accuracy but poor group accuracy for the small-sized group. In this case, we can improve the group accuracies by changing the objective function into maximizing the average group accuracy or maximizing the minimum group accuracy. Recall that $\mathcal{O}_k,\ k \in \mathcal{G}$ represents the set of indices of observations of group k. To maximize the average group accuracy, we replace the objective function (1.3.8) by (1.3.13). To maximize the minimum group accuracy, we use the objective function (1.3.14) with additional constraints (1.3.15) and (1.3.16).

$$\max \quad \sum_{k \in \mathcal{G}} \left(\frac{1}{|\mathcal{O}_k|} \sum_{i \in \mathcal{O}_k} u_{ki} \right) \tag{1.3.13}$$

$$\max \quad v \tag{1.3.14}$$

$$v \leq \frac{1}{|\mathcal{O}_k|} \sum_{i \in \mathcal{O}_k} u_{ki} \qquad \forall\ k \in \mathcal{G} \tag{1.3.15}$$

$$v \text{ urs} \tag{1.3.16}$$

Lee et al. [84] propose a linear programming approach, named DALP (discriminant analysis – linear program), as a heuristic method to get the λ's in Anderson's rule.

DALP

$$
\begin{aligned}
\min \quad & \sum_{i \in \mathcal{O}} (c_1 w_i + c_2 s_i) \\
\text{s.t.} \quad & L_{ki} = \pi_k \hat{p}_k(\boldsymbol{x}_i) - \sum_{h \in \mathcal{G}, h \neq k} \hat{p}_h(\boldsymbol{x}_i) \lambda_{hk} && \forall\ i \in \mathcal{O},\ k \in \mathcal{G} \\
& L_{y_i i} - L_{ki} + w_i \geq 0 && \forall\ i \in \mathcal{O},\ k \in \mathcal{G},\ k \neq y_i \\
& L_{y_i i} + w_i \geq 0 && \forall\ i \in \mathcal{O} \\
& -L_{ki} + s_i \geq 0 && \forall\ i \in \mathcal{O},\ k \in \mathcal{G} \\
& L_{ki} \text{ urs} && \forall\ i \in \mathcal{O},\ k \in \mathcal{G} \\
& w_i \geq 0 && \forall\ i \in \mathcal{O} \\
& s_i \geq 0 && \forall\ i \in \mathcal{O} \\
& \lambda_{hk} \geq 0 && \forall\ h, k \in \mathcal{G},\ h \neq k
\end{aligned}
$$

where c_1 and c_2 are constants controlling the emphasis on correctly classifying observations or placing them in the reserved judgement region and $\hat{p}_k(\boldsymbol{x}_i)$ is the normalized conditional probability density value defined by $\hat{p}_k(\boldsymbol{x}_i) = \frac{f_k(\boldsymbol{x}_i)}{\sum_{h \in \mathcal{G}} f_h(\boldsymbol{x}_i)}$.

The DAMIP/DALP approaches have been successfully applied to various multi-group disease diagnosis and biological/medical prediction problems [82, 83, 31, 32, 81].

CHAPTER II

FEATURE SELECTION

This chapter introduces the feature selection problem, discusses the concepts of feature relevance and redundancy, summarizes feature ranking methods, feature subset selection methods, mathematical-programming-based feature subset selection methods, and describes particle swarm optimization, a heuristic method used for feature subset selection.

2.1 Introduction

Feature selection is to select a subset of the original features in certain problems, including regression, clustering, and classification. The goal of feature selection can be (1) improving the prediction performance; (2) preventing over-fitting; (3) providing faster predictors; and (4) facilitating data understanding. The focus of this chapter is on feature selection for classification. In this section we provide two kinds of categorization of feature selection methods.

First we categorize feature selection methods into feature ranking and feature subset selection by the output of the methods. In *feature ranking*, we weigh and rank individual features; the output is the rankings of the features. Top ranking features can be used for classification, but feature ranking is not necessarily used for classification directly. We can apply feature ranking for initially reducing the number of features or simply for understanding the features. In *feature subset selection*, we search for a "good" subset of features based on certain objective function; the output is a subset of features, which is used for classification. The objective functions could

involve prediction accuracy, the number of selected features, or other criteria. We can also use cross-validation prediction accuracy as the objective function. Note that this categorization refers to the output of feature selection. In other words, in feature ranking we may evaluate subset of features to help for ranking individual features; in feature subset selection we may utilize individual feature ranking to help for choosing feature subsets.

The other categorization of feature selection methods is *filter methods* versus *wrapper methods* [65, 77, 79, 12, 27, 90]. Filter methods are independent of the classification methods while wrapper methods are dependent on the classification methods. That is, in filter methods, feature selection serves as a preprocessing step before classification; in wrapper methods, feature selection involves classification.

Some review papers have been proposed to survey and introduce feature selection. Dash and Liu [27] and Liu and Yu [90] identify four key steps and propose a categorizing framework for feature subset selection. See Section 2.4 for more details. Guyon and Elisseeff's introduction [58] covers feature ranking, feature subset selection, and related topics.

In this dissertation we do not consider feature extraction or construction, which constructs new features from the raw ones. Principal component analysis is an example of feature extraction methods. Instead, we deal with only the original features. Besides, in this dissertation we use the term "feature selection" instead of "variable selection." These two terms are sometimes interchangeably used. In [58] the authors call the raw data "variables" and the constructed ones "features." We need not distinguish them in this dissertation, and we use the term "feature selection" universally.

2.2 Feature Relevance and Redundancy

This section introduces the concept of feature relevance and feature redundancy and provides remarks on different feature selection methods and the selected features.

2.2.1 Feature Relevance

We define feature relevance according to [65, 77, 166]. Let the random vector $(Y, X_1, \cdots, X_m) \in \mathcal{G} \times \mathcal{F}_1 \times \cdots \times \mathcal{F}_m$ represent the data point, where Y denotes the group and $(X_1, \cdots, X_m)$ denotes the vector of features. Observed value of a data point is denoted by $(y, x_1, \cdots, x_m)$. Let S_j be the set of all features except X_j, i.e., $S_j = \{X_1, \cdots, X_{j-1}, X_{j+1}, \cdots, X_m\}$, and let s_j be its observed value. Also, let p be the probability measure on the space $\mathcal{G} \times \mathcal{F}_1 \times \cdots \times \mathcal{F}_m$.

A feature X_j is *strongly relevant* if and only if there exists some x_j, y, and s_j for which $p(X_j = x_j, S_j = s_j) > 0$ such that

$$p(Y = y | X_j = x_j, S_j = s_j) \neq p(Y = y | S_j = s_j).$$

A feature X_j is *weakly relevant* if and only if it is not strongly relevant, and there exists a subset of features S'_j of S_j for which there exists some x_j, y, and s'_j with $p(X_j = x_j, S'_j = s'_j) > 0$ such that

$$p(Y = y | X_j = x_j, S'_j = s'_j) \neq p(Y = y | S'_j = s'_j).$$

A feature is *relevant* if it is either strongly relevant or weakly relevant; otherwise it is *irrelevant*.

The above definitions are based on an optimal Bayes classifier. X_j is strong relevant if the removal of X_j alone deteriorates the performance of an optimal Bayes classifier; X_j is weak relevant if it is not strong relevant and there exists a subset of

features, S'_j, such that a Bayes classifier performs worse on S'_j than on $S'_j \cup \{X_j\}$; otherwise, X_j is irrelevant.

The definition of feature relevance helps us to identify the general goal: find all strongly relevant features, a useful subset of weakly relevant features, but no irrelevant features [65]. However, in practice, in a problem with a particular objective function and a particular classification method, a relevant feature, even strongly relevant, does not imply that it is in the optimal feature subset; an irrelevant feature does not imply that it should not be in the optimal feature subset. Incorporating the classifier when searching for a good subset, wrapper methods for feature subset selection can perform better, particularly in prediction accuracy [77].

2.2.2 Feature Redundancy

Intuitively, feature redundancy relates to correlation. That is, if two features are highly correlated, one of them could be redundant. To deal with redundancy which involves more than two features, Yu and Liu [166] introduce the definition of feature redundancy among relevant features based on a feature's Markov blanket defined by Koller and Sahami [78].

Let F be the set of all features. Given a feature X_j, let $S'_j \subset F$ and $X_j \notin S'_j$. S'_j is said to be a *Markov blanket for* X_j if and only if

$$p(F - S'_j - \{X_j\}, Y | S'_j, X_j) = p(F - S'_j - \{X_j\}, Y | S'_j).$$

Let F' be a subset of features. A feature is *redundant* in F' if and only if it is weakly relevant and has a Markov blanket within F'.

The definition partitions the set of weakly relevant features into two parts: (1)

weakly relevant and redundant features; (2) weakly relevant but non-redundant features. Note that the partition may not be unique.

In the previous subsection, the definition of feature relevance identifies the general goal: finding all strongly relevant features and a useful subset of weakly relevant features. The definition of feature redundancy further characterizes that non-redundant features are the useful weakly relevant ones. However, similar to the situation of feature relevancy, optimal subset for a particular problem may include features which are redundant to each other.

Guyon and Elisseeff [58] discuss feature redundancy in the meaning of correlation. Through examples, they show that including high correlated features can significantly help to separate data of different groups, a self-useless feature can significantly improve the classification performance when taken with others, and two self-useless features can provide good separation together.

2.2.3 Remarks

The discussion of feature relevance and redundancy and the examples in the literature suggest that

1. In a problem with a particular objective function and a particular classification method, features chosen from wrapper methods for subset selection perform better than features obtained from top ranking features by ranking methods.

2. Features chosen from subset selection may not include all strongly relevant features, presumed to be important, and may include redundant or irrelevant features.

2.3 Feature Ranking

In feature ranking, we weigh and rank individual features. That is, we compute a score S_j for each feature j and sort the scores. We can categorize feature ranking methods into univariate or multivariate. If the computing of S_j involves the data from feature j but no data from other features, the ranking is called univariate, otherwise it is multivariate.

A simple feature ranking method for the two-group problem is the Pearson's correlation coefficient between feature j and group,

$$S_j = \frac{\sum_{i\in\mathcal{O}}(x_{ij} - \bar{x}_j)(y_i - \bar{y})}{\sqrt{\sum_{i\in\mathcal{O}}(x_{ij} - \bar{x}_j)^2 \sum_{i\in\mathcal{O}}(y_i - \bar{y})^2}},$$

where $\bar{x}_j$ is the average value of feature j and $\bar{y}$ is the average value of group, both over all observations.

Some feature ranking methods calculate a score for each feature by performing statistical tests; for example, t-test on the values of feature j between two groups. Some feature ranking methods are based on mutual information; for example, the minimal-redundancy-maximal-relevance criterion [125]. A simple wrapper method for feature ranking is to classify the data using each single feature by a classification method and regard the classification result as the score of that feature.

Guyon et al. [59] propose a feature selection method using support vector machine to select genes which separate normal and cancer subjects. Initially putting all features in the list, this method iteratively trains the SVM with remaining features in the list to get the weight vector $\boldsymbol{w}$ in Equation (1.2.1) (see Section 1.2.4) and removes the feature with the smallest value of w_j^2. The order of the removal of features gives the feature ranking. This method is a wrapper method for feature ranking in

the two-group problem. Similarly, Rakotomamonjy [130] investigates three feature ranking criteria which are also based on SVM.

2.4 Feature Subset Selection

Feature subset selection can be regarded as a combinatorial optimization problem, in which we find an optimal subset of the original feature set according to an objective function. The size of the selected subset can be predetermined, unrestricted, or involved in the objective function. The selected features are then used for classification. Figure 2.4.1 shows the process of feature subset selection for classification.

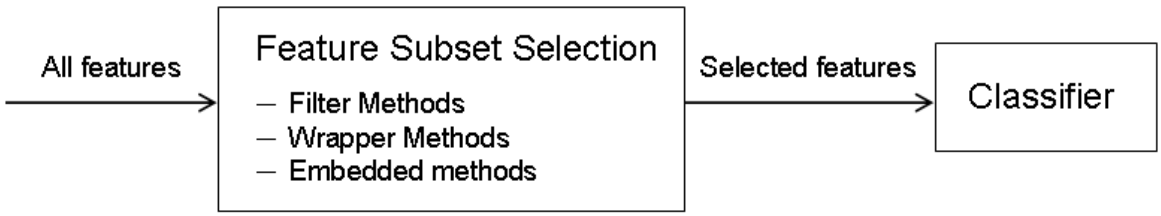

Figure 2.4.1: Feature subset selection for classification.

Feature subset selection methods are categorized into filter methods, wrapper methods [79, 65, 77], and embedded methods [12, 58], in which the first two are mentioned in Section 2.1. In filter methods, the feature selection process does not involve the classifier; feature selection filters features and passes them to the classifier. In wrapper methods, the feature selection process uses the classifier as a black-box to guide the selection; feature selection is regarded as a wrapper around the classifier. In embedded methods, the feature selection process is performed while the classifier is trained; feature selection is embedded within the classifier.

Dash and Liu [27] and Liu and Yu [90] provide a framework to categorize feature subset selection methods. They identify four steps in a feature subset selection method: subset generation, subset evaluation, stopping criterion, and result validation, shown in Figure 2.4.2. In fact, the first three steps describe the procedure of

solving the feature subset selection optimization problem, either heuristic or exact. Furthermore, the authors categorize feature selection methods according to search strategies, evaluation criteria, and data mining tasks, providing a way to distinguish existing methods. In their framework, data mining tasks include classification and clustering.

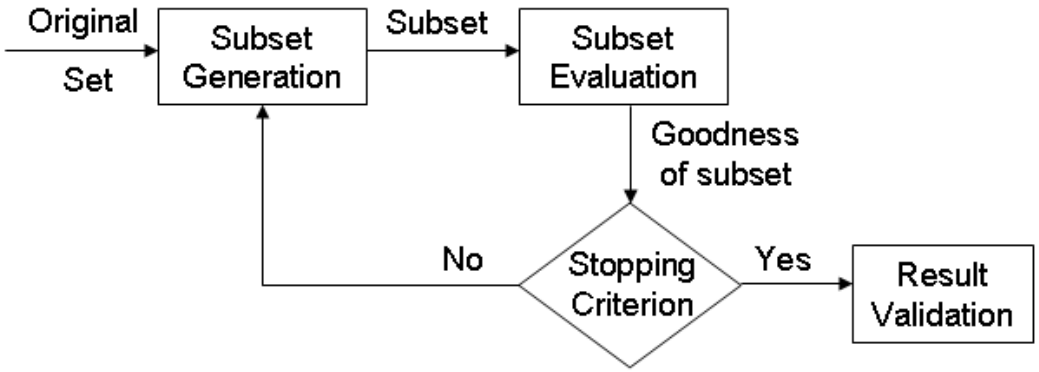

Figure 2.4.2: Four steps of feature subset selection [90].

In Liu and Yu's framework, search strategies include complete search, sequential search, and random search, which are exactly the solution methods to the feature subset selection optimization problem. As for exact solution method, Narendra and Fukunaga [112] propose a branch and bound algorithm for selecting the best k features out of m original ones. To utilize the power of branch and bound, the objective function needs to be monotone, i.e., the performance of a subset A should not be worse than any proper subset of A. The assumption of monotonic property is not very restrictive; however, if the objective function involves cross-validation technique to overcome over-fitting, monotonic property will not be satisfied.

Traditional sequential search methods include sequential forward selection (SFS)

and sequential backward selection (SBS). SFS starts with no features and greedily selects features; SBS starts with all features and greedily drops features. Stearns [149] proposes the plus-l-minus-r search method to prevent the drawback of SFS (SBS) that once a feature is selected (dropped), it can no longer be dropped (selected). Pudil et al. [129] propose floating search methods, which make the sequential search more flexible, and Somol et al. [143] propose a more complex version of the floating search methods. Siedlecki and Sklansky [139] apply genetic algorithms to the feature selection problems, which is an example of random search in the framework.

In Liu and Yu's framework, evaluation criteria are categorized into filter methods, wrapper methods, and hybrid methods, which relate to the objective function of the feature subset selection optimization problem. As described before, filters are independent of the classifiers while wrappers are dependent on the classifiers. In hybrid methods, both kinds of evaluation criteria (i.e., independent and dependent ones) are used.

An example of the wrapper evaluation criterion is called LS bound, derived from leave-one-out procedure of LS-SVM (least squares SVM) to evaluate gene selection by Zhou and Mao [167]. An example of the filter evaluation criterion is the criterion proposed by Bruzzone and Serpico [20] for the classification of remote sensing images acquired by passive sensors. The criterion is based on an upper bound to the Bayes error under the assumption that each group follows the Gaussian distribution with equal covariance. The proposed criterion is

$$\sum_{i=1}^{K}\sum_{j>i}^{K}\{(p(\omega_i)+p(\omega_j))Q(\frac{\sqrt{d_{ij}}}{2})\},$$

where $p(\omega_i)$ is the prior probability of group ω_i, $Q(x)=\frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-\xi^2/2}d\xi$, and d_{ij} is the Mahalanobis distance between group ω_i and ω_j. It is shown to outperform some

criteria used in remote sensing, such as criteria based on the Bhattacharyya distance, the Jeffreys–Matusita (J–M) distance, and scatter matrices.

Lasso (least absolute shrinkage and selection operator) [155] and LARS (least angle regression) [28] are examples of embedded feature selection methods. They are designed for regression problems, but a regression model can be regarded as a binary classifier when the dependent variable y_i's in the data have value 1 or -1 instead of continuous values and the predicted group of a new observation is determined by the sign of the regressed y from x_j's.

Tibshirani [155] proposes the Lasso model. Assuming that x_{ij}'s are standardized so that $\sum_{i\in\mathcal{O}} x_{ij}/|\mathcal{O}| = 0$ and $\sum_{i\in\mathcal{O}} x_{ij}^2/|\mathcal{O}| = 1$ for all $j \in \mathcal{F}$, the Lasso estimate $w_j,\ j \in \{0\} \cup \mathcal{F}$, is given by solving the following quadratic program, where t is a tuning parameter.

Lasso

$$\begin{aligned} \min \quad & \sum_{i\in\mathcal{O}} \left(y_i - w_0 - \sum_{j\in\mathcal{F}} x_{ij} w_j \right)^2 \\ \text{s.t.} \quad & \sum_{j\in\mathcal{F}} |w_j| \le t \\ & w_j \text{ urs} \qquad\qquad \forall\ j \in \{0\} \cup \mathcal{F} \end{aligned}$$

This model tends to produce some w_j's which are exactly 0, so by tuning the parameter t, we also do feature selection in the same time when we train the binary classifier.

LARS proposed by Efron et al. [28] is a procedure to calculate the values of w_j's

in a regression model, one w_j in each step, based on the equiangular directions. This procedure is regarded as a feature selection algorithm, which relates to the classical forward selection but is less greedy. LARS is computationally efficient, and the Lasso solution can be obtained by a simple modification of the LARS procedure.

Keerthi [66] generalizes the LARS feature selection procedure to the SVM classifier with L_2 loss function. Computational study shows that this is an effective feature selection method.

2.5 Feature Subset Selection via Mathematical Programming

In this section we review some feature subset selection methods which utilize mathematical programming, and we point out the type of these methods according to the filter-wrapper-embedded-method categorization.

Bertolazzi et al. [11] propose a feature selection formulation for the two-group classification problem with binary data, i.e., the values of all features are binary. This method belongs to the filter methods. Originally the feature selection problem for two-group binary-feature data is formulated as a set covering problem:

$$\begin{aligned} \min \quad & \sum_{j \in \mathcal{F}} z_j & & \\ \text{s.t.} \quad & \sum_{j \in \mathcal{F}} a_{i_1 i_2, j} z_j \geq 1 & & \forall\ i_1 \in \mathcal{O}_1,\ i_2 \in \mathcal{O}_2 \qquad (2.5.1) \\ & z_j \in \{0, 1\} & & \forall\ j \in \mathcal{F} \end{aligned}$$

where $a_{i_1 i_2, j} = 1$ if and only if $x_{i_1, j} \neq x_{i_2, j}$, and $a_{i_1 i_2, j} = 0$ otherwise. The meaning of $a_{i_1 i_2, j} = 1$ is that observation i_1 and i_2 from different groups have distinct values in feature j. In this formulation, the decision variable z_j has value one if feature j is selected, and the objective function is to minimize the number of selected features. The constraint (2.5.1) says that, for each pair of observations from different groups, at least one feature must be selected such that this feature distinguishes, or covers, these two observations.

To increase the power of prediction on not only training data but also testing data, the authors propose to raise the right hand side of (2.5.1), resulting in (2.5.2),

$$\sum_{j \in \mathcal{F}} a_{i_1 i_2, j} z_j \geq \alpha \qquad \forall i_1 \in \mathcal{O}_1, i_2 \in \mathcal{O}_2 \qquad (2.5.2)$$

where α is an integer which measures the degree of information provided by the selected features for the generation of prediction rules. In this case the set covering problem becomes the generalized set covering problem.

Furthermore, to reduce the computational burden in this model, an approximated formulation is proposed, which reduces the number of constraints from quadratic to linear in the number of observations. Define $P_1(j)$ and $P_2(j)$, for each feature j, as the proportion of observations in $\mathcal{O}_1$ and $\mathcal{O}_2$ which have value 1 in feature j. The constraint (2.5.2) is replaced by (2.5.3),

$$\sum_{j \in \mathcal{F}} d_{ij} z_j \geq \alpha \qquad \forall i \in \mathcal{O} \tag{2.5.3}$$

where d_{ij} has value 1 or 0 based on the following table.

	$P_1(j) > P_2(j)$		$P_1(j) < P_2(j)$	
d_{ij}	$x_{ij} = 1$	$x_{ij} = 0$	$x_{ij} = 1$	$x_{ij} = 0$
$i \in \mathcal{O}_1$	1	0	0	1
$i \in \mathcal{O}_2$	0	1	1	0

The generalized set covering problem is then solved by the greedy randomized adaptive search procedure.

In the two-group classification problem, when a hyperplane is obtained to separate data of two groups under certain objectives such as minimizing the sum of deviations or minimizing the number of misclassifications, Glen [49, 50, 51, 52] introduces to these models the constraint that only p of the coefficients, $w_1, \cdots, w_m$, of the separating hyperplane $w_0 + \sum_{j=1}^{m} x_{ij} w_j = 0$ can be nonzero. The models then find the optimal hyperplane while restrict the number of selected features to p. Section 1.2.2.1 has more details on these classification models. These models belong to the embedded

feature subset selection methods.

Iannarilli and Rubin [62] propose mathematical programming models to select features in the multigroup classification problems. The methods are categorized as embedded methods. In these models, the binary variable z_j indicates the selection of feature j, and the objective function measures the pairwise intergroup margin between groups, which is based on the group conditional distributions of the reduced feature vectors and is related to the Bayes error. The original formulation is an integer nonlinear program:

$$\begin{aligned} \max \quad & \sum_{h,k \in \mathcal{G}, h<k} \max_{j \in \mathcal{F}} \{a_j^{hk} z_j\} \\ \text{s.t.} \quad & \sum_{j \in \mathcal{F}} z_j \leq r \\ & z_j \in \{0,1\} \qquad\qquad \forall j \in \mathcal{F}, \end{aligned}$$

where r is the upper bound of the number of selected features, and a_j^{hk} could be a weighted L_1 metric applied to the group means,

$$a_j^{hk} = f\left(c \frac{|\mu_{hj} - \mu_{kj}|}{\frac{\sigma_{hj}\sigma_{kj}}{\sigma_{hj}+\sigma_{kj}}} \right),$$

in which μ_{hj} and σ_{hj} are the training set conditional mean and standard deviation of feature j given group h, c is a positive parameter, and $f()$ is a bounded function, e.g., the sigmoidal function tanh(). There are other choices of a_j^{hk} as other metrics.

This nonlinear formulation is then transformed to an equivalent linear one so that it could be solved to optimality by standard solvers. With the auxiliary variable w_j^{hk}, the integer linear program is

$$
\begin{aligned}
\max \quad & \sum_{h,k\in\mathcal{G},h<k} \sum_{j\in\mathcal{F}} a_j^{hk} w_j^{hk} \\
\text{s.t.} \quad & \sum_{j\in\mathcal{F}} z_j \le r \\
& \sum_{j\in\mathcal{F}} w_j^{hk} \le 1 && \forall h,k \in \mathcal{G}, h<k \\
& w_j^{hk} \le z_j && \forall h,k \in \mathcal{G}, h<k,\ \forall j \in \mathcal{F} \\
& z_j \in \{0,1\} && \forall j \in \mathcal{F} \\
& w_j^{hk} \ge 0 && \forall h,k \in \mathcal{G}, h<k,\ \forall j \in \mathcal{F}
\end{aligned}
$$

The above nonlinear/linear formulation is regarded as the L_∞ model since, for each pair of groups, only one feature with the maximum a_j^{hk} accounts in the objective function. The authors propose the constrained L_p model in which not just one feature could account for each pair of groups. Given the user-specified lower bounds λ^{hk}'s on the intergroup margins, the constrained L_p model is

$$
\begin{aligned}
\max \quad & \sum_{j\in\mathcal{F}} \bar{a}_j z_j \\
\text{s.t.} \quad & \sum_{j\in\mathcal{F}} z_j \le r \\
& \sum_{j\in\mathcal{F}} a_j^{hk} z_j \ge \lambda^{hk} && \forall h,k \in \mathcal{G}, h<k \\
& z_j \in \{0,1\} && \forall j \in \mathcal{F},
\end{aligned}
$$

where

$$
\bar{a}_j = \frac{1}{\binom{K}{2}} \sum_{h,k\in\mathcal{G},h<k} a_j^{hk}
$$

denotes the average intergroup margin for feature j. The parameter λ^{hk} has to be

tuned through trial and error.

2.6 Particle Swarm Optimization

Particle swarm optimization (PSO) is a heuristic algorithm to solve an optimization problem. It is an evolutionary computation technique originally developed by Kennedy and Eberhart [68]. Candidate solutions, named positions of particles, are initialized randomly in the solution space. In each iteration of the algorithm, each particle moves to a new position based on a randomly-generated velocity, which is affected by the best position (i.e., the solution with the best objective value) achieved so far by this particle and the best position achieved so far by the particle in its neighborhood. In PSO the population is called the swarm and the objective function is called the fitness function. We will solve feature subset selection problem by PSO.

Elbeltagi et al. [29] compare five evolutionary-based optimization algorithms, including genetic algorithms, memetic algorithms, PSO, ant-colony systems, and shuffled frog leaping and conclude that PSO generally performs better than the other methods.

2.6.1 Introduction to PSO

This section introduces the PSO algorithm, neighborhood topologies, convergence properties, parameter selection, and other topics.

2.6.1.1 Algorithm

Let $\boldsymbol{x}_i$ be the position vector and $\boldsymbol{v}_i$ be the velocity vector of particle i. Let $\boldsymbol{p}_i$ be the best position vector of particle i in the history, i.e., the position possessing the best fitness value among all positions visited so far by particle i. The initialization and updating of $\boldsymbol{x}_i$'s and $\boldsymbol{v}_i$'s and the termination of PSO are as the following.

- Initialization:

  Randomly generate $\boldsymbol{x}_i$ and $\boldsymbol{v}_i$ within predetermined ranges for each particle i.

- Updating:

  In each iteration, $\boldsymbol{x}_i$ and $\boldsymbol{v}_i$ are updated by

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i\, \omega + (\boldsymbol{p}_i - \boldsymbol{x}_i)\, c_1\, rand() + (\boldsymbol{p}_{n^*(i)} - \boldsymbol{x}_i)\, c_2\, rand(), \tag{2.6.1}$$

$$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \boldsymbol{v}_i, \tag{2.6.2}$$

  where $n^*(i)$ is the index of the best particle (i.e., having the best fitness value in the history) in the neighborhood of the ith particle, $rand()$ denotes a random number (i.e., $rand() \sim U(0,1)$), and ω, c_1, and c_2 are parameters. Also, each component of $\boldsymbol{v}_i$ is restricted within the range $[-V_{\max}, V_{\max}]$ to prevent the speed from being out of control ($V_{\max}$ is a predetermined value).

- Termination:

  The PSO algorithm terminates when certain criteria are met. The criteria could involve the number of iterations (e.g. achieving an upper bound), the fitness value (e.g. having few improvement), or the position vectors (e.g. most particles not moving, particularly in discrete case).

ω, c_1, and c_2 are meaningful parameters in (2.6.1). The inertia weight ω is introduced by Shi and Eberhart [138], which improves the convergence of PSO when its value as well as values of other parameters are appropriately chosen. c_1 and c_2 are known as the acceleration coefficients of the cognitive part and the social part, respectively. In the social-psychological metaphor, the cognitive part represents self-learning of the particle while the social part represents learning from other particles.

Clerc and Kennedy [23] introduce the constriction coefficient χ, which ensures convergence of PSO without using $V_{\max}$ to limit the velocity. Under this version of

PSO, the velocity updating formula (2.6.1) becomes

$$\boldsymbol{v}_i \leftarrow \chi \cdot [\boldsymbol{v}_i + c_1 \cdot rand() \cdot (\boldsymbol{p}_i - \boldsymbol{x}_i) + c_2 \cdot rand() \cdot (\boldsymbol{p}_{n^*(i)} - \boldsymbol{x}_i)]. \tag{2.6.3}$$

Note that a PSO with constriction (i.e., (2.6.3) and (2.6.2)) is algebraically equivalent to a PSO with inertia (i.e., (2.6.1) and (2.6.2)), given that the inertia is fixed but not changeable during the PSO process.

Mendes et al. [110] propose the fully-informed version of PSO, in which the next position of each particle is affected by the best positions of "all" particles in its neighborhood instead of the best one. In this case the velocity updating formula (2.6.1) becomes

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i \, \omega + \sum_{j \in N(i)} (\boldsymbol{p}_j - \boldsymbol{x}_i) \, c_j \, rand(), \tag{2.6.4}$$

where $N(i)$ denotes the neighborhood of particle i. Their computational study shows that the fully-informed PSO which assigns equal weights to each neighbor of the particle performs better.

2.6.1.2 Neighborhood Topologies

Several neighborhood topologies for PSO are proposed in the literature [67, 70, 124], including *gbest* (global best), *lbest* (local best), von Neumann topology, and so on. The *gbest* topology treats the entire population as the neighborhood of the target particle. The *lbest* topology is described as a one-dimensional ring lattice, where all particles are aligned and form a ring, as shown in Figure 2.6.1(a). Different neighborhood sizes can be used in *lbest*. For example, *lbest* with size 3 represents that the neighborhood of the ith particle contains particles $i-1$, i, and $i+1$; if with size 5, particle $i-2$ and $i+2$ are further included. The von Neumann topology is described as a two-dimensional lattice, where in the 2d grid a particle has neighbors above,

below, right, and left. See Figure 2.6.1(b). Note that the neighbors of a particle are determined before the algorithm starts and are based on the indices of particles but not on the positions of particles or the distances between particles in the solution space. Besides, in any topology a particle itself can be either included in or excluded from its neighborhood; however, study shows that the inclusion/exclusion of the target particle has little impact on behavior [70, 128].

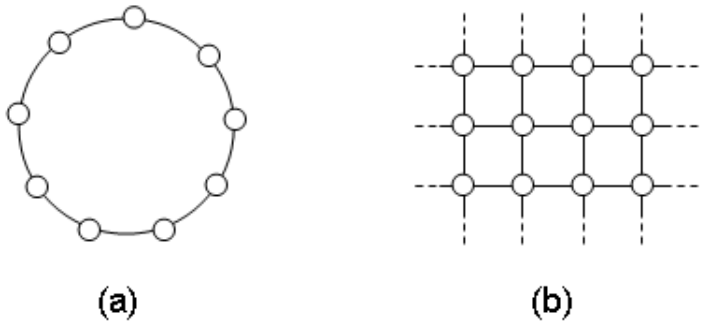

Figure 2.6.1: Neighborhood topology of PSO. (a) *lbest*; (b) von Neumann.

The effect of neighborhood topology on the performance of PSO is significant and is dependent on the fitness function. Kennedy and Mendes [70] recommend the von Neumann topology since it performs more superiorly and consistently in the experiment. Besides, PSO with higher connected population topology converges faster and tends to be better for unimodal problems, and vice versa [67, 70].

2.6.1.3 Convergence Properties and Parameter Selection

Two types of tools have been used to analyze the convergence properties of PSO. Clerc and Kennedy [23], Trelea [157], and van den Bergh and Engelbrecht [158] study the convergence properties of PSO using dynamic system theory, in which the analysis starts from a simplified deterministic model (one-particle system without randomness). On the other hand, Jiang et al. [63] study the convergence properties using

stochastic process theory. Regardless of the analysis methods, conditions about parameters ω, c_1, and c_2 under which the PSO system converges are given and guidelines of parameter selections are also provided. Note that the convergent position of each particle is the overall best position found so far, which says nothing about any local or global optimal solutions.

One good setting of the parameters for the version with inertia is $\omega = 0.7298$ and $c_1 = c_2 = 1.49618$. These numbers satisfy convergence conditions and are popular used in PSO literature [158, 128].

2.6.1.4 Others

Many applications, generalization, and variants of the basic PSO algorithm have been studied since PSO is first proposed. Poli et al. [128] give a comprehensive overview of PSO in the newly established journal *Swarm Intelligence*. Laskari et al. [80] apply PSO to solving integer nonlinear programming problems and compare its performance with that of branch and bound technique. Parsopoulos and Vrahatis [117] propose an approach for computing all global minimizers of an objective function, where techniques for objective function transformation are incorporated in the context of PSO. Parsopoulos and Vrahatis propose the unified particle swarm optimization scheme, which combines the global and local PSO variants. Although the neighborhood structure *gbest* is a generalization of *lbest*, the authors distinguish between them due to their exploitation and exploration properties, respectively. Parameter selection and adaptation in unified PSO are also studied [118]. Petalas et al. [126] propose the memetic particle swarm optimization scheme, which incorporates local search techniques to the standard PSO. The new scheme performs better than PSO in different types of test problems. Poli [127] analyzes the publications on the applications of

PSO. Fernandez Martinez and Garcia Gonzalo [33] generalize the PSO algorithm for any time step Δt:

$$\begin{aligned}\boldsymbol{v}_i(t+\Delta t) \leftarrow\ & \boldsymbol{v}_i(t)\ (1-(1-\omega)\Delta t) \\ & + (\boldsymbol{p}_i(t) - \boldsymbol{x}_i(t))\ c_1\ rand()\ \Delta t \\ & + (\boldsymbol{p}_{n^*(i)}(t) - \boldsymbol{x}_i(t))\ c_2\ rand()\ \Delta t, \qquad (2.6.5)\end{aligned}$$

$$\boldsymbol{x}_i(t+\Delta t) \leftarrow \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t+\Delta t)\Delta t. \qquad (2.6.6)$$

This generalization is based on a mechanical analogy: a damped mass-spring system. A family of PSO versions is also derived from this mechanical analogy [34]. Cooren et al. [25] study TRIBES, an adaptive version of PSO, which avoids manual parameter tuning. In TRIBES the users determine only the adaptation rules while the particles' behaviors and the topology changes automatically.

2.6.2 Binary PSO

Kennedy and Eberhart [69] modify the PSO algorithm to work on binary variables. The binary PSO also operates on continuous variables since a continuous value can be represented as a bit string given a prespecified precision. In Kennedy and Eberhart's binary PSO algorithm, all position vectors become binary vectors, but the velocity vectors remain continuous. The velocity is used to define the probabilities that a bit is one or zero. With the subscript j denoting the jth dimension of $\boldsymbol{x}_i$ and $\boldsymbol{v}_i$, the position updating formula becomes

$$x_{ij} = \begin{cases} 1 & \text{if } rand() < S(v_{ij}), \\ 0 & \text{otherwise,} \end{cases} \qquad (2.6.7)$$

where $S(v_{ij}) = \frac{1}{1+e^{-v_{ij}}}$ is the sigmoid function. The authors implement this method using population topology *lbest* which includes the target particle and has neighborhood size 3. Kennedy and Spears [71] compare this binary PSO with some versions

of genetic algorithm on solving multimodal problems. In their experiment, the binary PSO appeares to be robust.

Pampara et al. [114] propose the Angle Modulated PSO (AMPSO) which employs a trigonometric function to generate bit strings. Instead of evolving the binary vectors representing candidate solutions to the original problem, the standard PSO is applied to optimize a simpler 4-dimensional tuple (a, b, c, d) that are the parameters of the generating function

$$g(x) = \sin(2\pi(x - a) \times b \times \cos(A)) + d, \tag{2.6.8}$$

where

$$A = 2\pi \times c(x - a).$$

When the values of (a, b, c, d) are obtained in an iteration of PSO, they are substituted back into function (2.6.8). Suppose that m is the dimension of the binary vector to the original problem and that m evenly spaced intervals in the domain of x is predetermined. The binary vector is generated by the following procedure: sampling a point at each interval and evaluating $f(x)$ at that point; if $f(x)$ is positive, the variable corresponding to this interval is assigned to be 1, otherwise 0. Experimental results using von Neumann topology show that AMPSO performs better than the original binary PSO proposed by Kennedy and Eberhart [69].

2.6.3 Feature Selection Using PSO

In feature subset selection problem, the selection of features can be represented as a binary vector, so the binary versions of PSO algorithm described in Section 2.6.2 are candidate solution methods. Besides, several PSO variants are proposed for feature selection although some of them are applied to regression or other problems instead

of classification.

Agrafiotis and Cedeño [2] present a binary adaption of PSO for feature selection, applied in the construction of quantitative structure-activity relationship models based on neural networks. In this feature selection application, $x_{ij} = 1$ if the jth feature of the ith particle is selected and $x_{ij} = 0$ if not selected. In each iteration and for the ith particle, the number of features to be selected, k, is predefined. x_{ij} is obtained by applying formulas (2.6.1) and (2.6.2) and is confined in the interval $[0, 1]$. Then the features are selected by employing roulette wheel selection: assigning the jth feature a slice of a roulette wheel whose size is the probability q_{ij} obtained from equation (2.6.9); then spinning the wheel and selecting the feature under the wheel's marker until k distinct features are selected.

$$q_{ij} = \frac{x_{ij}^{\alpha}}{\sum_j x_{ij}^{\alpha}} \tag{2.6.9}$$

In equation (2.6.9) the parameter α represents the selection pressure. In the computational tests, α is set to be 2; *lbest* including the target particle with neighborhood size 5 is used for the population topology. The results show that this method compares favorably with simulated annealing.

Monteiro and Kosugi [111] propose a feature selection method to extract information from hyperspectral imagery data. This method uses two particle swarms simultaneously, one PSO for deciding the number of selected features and the other PSO for selecting features. The continuous version of PSO in one-dimension is used to search for the number of selected features, where values are discretized by rounding to the nearest integers. The binary PSO by Agrafiotis and Cedeño [2] described above is used to select features. Neural networks are utilized to construct regression models with features selected by PSO.

Shen et al. [137] propose a modified binary PSO for feature selection in multiple linear regression and partial least-squares modeling. The *gbest* neighborhood topology is used here. Let g be the index of the best particle in the whole population and recall that $\boldsymbol{p}_i$ is the best previous position of the ith particle. In this proposed algorithm the velocity v_{ij} is defined as a random number, and the position x_{ij} is updated by the rule:

$$\text{If } (0 < v_{ij} \leq a), \quad \text{then } x_{ij} \leftarrow x_{ij}, \tag{2.6.10}$$

$$\text{If } (a < v_{ij} \leq \frac{1}{2}(1+a)), \quad \text{then } x_{ij} \leftarrow p_{ij}, \tag{2.6.11}$$

$$\text{If } (\frac{1}{2}(1+a) < v_{ij} \leq 1), \quad \text{then } x_{ij} \leftarrow p_{gj}, \tag{2.6.12}$$

where a is a value in the range of $(0, 1)$ and initially set as 0.5. Using decreasing values of a and some percentage of particles not following previous bests, this method has satisfactory performance and convergence rate compared with genetic algorithms.

Wang and Yu [161] modify the method of Shen et al. by introducing mutation and apply to fault diagnosis in chemical process. SVM is used as the classifier and the fitness function includes the correct classification rate as well as the number of selected features.

Correa et al. [26] present a discrete PSO designed for feature selection. The velocity is defined by proportional likelihoods which are affected by previous bests. Each component of the velocity is multiplied by a random number, and the features corresponding to the larger components of the velocity are selected.

Liu et al. [91] propose a method for solving classification problems where radial basis function (RBF) neural network is used as the classifier and feature selection is included. In their method the standard PSO is employed to do feature selection and

neural network training simultaneously. Each particle consists of two parts: flags for feature selection and parameters of the neural network. The jth feature is selected if the corresponding flag is positive and not selected if its flag is nonpositive.

Ressom et al. [132, 131] combine SVM with PSO, applying to serum mass spectral profiles for biomarker discovery. The biomarkers are mass-per-charge values (i.e., m/z values), which form a continuous space. The continuous version of PSO with *gbest* population topology operates in this continuous space to select biomarkers which better distinguish cancer patients from healthy individuals.

Tang et al. [154] propose an evaluation criterion for feature selection, which originates from an exact calculation of the leave-one-out error of a least squares SVM, and present a searching scheme to combine with the proposed criterion. In the searching scheme, principle component analysis is applied to transform the original data, scaling factors are introduced into the kernel matrix for SVM, and standard PSO is used to optimize the evaluation criterion with respect to the scaling factor for the transformed data. This feature selection method has good performance and low computational cost to perform gene selection from DNA microarray data.

Samanta and Nataraj [136] implement PSO together with proximal support vector machines [42] for machinery fault detection. Two versions of PSO are implemented in this study. The first version is the binary PSO proposed by Kennedy and Eberhart [69], in which the position vector $\boldsymbol{x}_i$ is a binary vector indicating weather a feature is selected or not. The second version is the original real-valued PSO, in which the position vector represents the indices of selected features. In this implementation, the dimension of the position vector is a predetermined value equal to the number

of selected features, the values in the position vector are rounded to integers as indices, and a solution is excluded if a feature appears more than once in the solution. Simulation results show that the difference between these two versions of PSO is not significant.

CHAPTER III

PSO/DAMIP FRAMEWORK

This chapter introduces the PSO/DAMIP classification framework, which uses PSO (particle swarm optimization) for feature selection and DAMIP (discriminant analysis – mixed integer program) for classification. This is a wrapper method in which PSO iteratively searches for the subsets of the original features with a good ten-fold cross-validation classification accuracy obtained by solving DAMIP using the selected features. Topics include the modified PSO, exact algorithms for solving two-group DAMIP, heuristics for solving multigroup DAMIP without misclassification constraints, and trials on solving DAMIP with cuts.

3.1 Modified PSO

We directly modify the original PSO algorithm to solve the binary problem—the feature subset selection problem. In our framework, the number of selected features is determined based on the number of observations and the number of features; in our applications it is usually chosen from 3 to 20. The modified algorithm and the selection of PSO parameters are described in this section.

3.1.1 Algorithm

Recall that $\boldsymbol{x}_i$ is the position vector and $\boldsymbol{v}_i$ is the velocity vector of particle i. Also, $\boldsymbol{p}_i$ is the best position vector of particle i in the history. We fix the number of selected features to be k. The PSO algorithm is as the following.

- Initialization:

For each particle i, $\boldsymbol{x}_i$ is generated such that ones are in randomly-selected k components and zeros are in the remaining components; $\boldsymbol{v}_i$ is generated by $v_{ij} \sim U(-V_{init.max}, V_{init.max})$.

- Updating:

  In each iteration, $\boldsymbol{v}_i$ is updated by

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i\, \omega + (\boldsymbol{p}_i - \boldsymbol{x}_i)\, c_1\, rand() + (\boldsymbol{p}_{n^*(i)} - \boldsymbol{x}_i)\, c_2\, rand(). \tag{3.1.1}$$

  (Recall that $n^*(i)$ is the index of the best particle in the neighborhood of the ith particle, $rand() \sim U(0,1)$, and ω, c_1, and c_2 are parameters.)

  $\boldsymbol{x}_i$ gets k ones in the components whose corresponding components in $\boldsymbol{v}_i$ have the largest k valeus. The other components in $\boldsymbol{x}_i$ get zeros.

- Termination:

  The PSO algorithm terminates when (1) the maximum number of iterations is achieved, or (2) the percentage of the number of particle moving is less than a threshold.

We can also use the velocity updating formula (3.1.2) in the fully-informed version of PSO instead of (3.1.1):

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i\, \omega + \sum_{j \in N(i)} (\boldsymbol{p}_j - \boldsymbol{x}_i)\, c_j\, rand(), \tag{3.1.2}$$

where $N(i)$ denotes the neighborhood of particle i.

For the $\boldsymbol{x}_i$ updating, instead of determining the zeros and ones based on the values of $\boldsymbol{v}_i$, we can also use an alternative way: after updating $\boldsymbol{v}_i$ by (3.1.1) or (3.1.2), we first update $\boldsymbol{x}_i$ by

$$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \boldsymbol{v}_i, \tag{3.1.3}$$

then change the largest k components of $\boldsymbol{x}_i$ to ones and others to zeros.

All solutions visited during the PSO process are stored, so when a particle visits a position which is already recorded, the objective function value is obtained directly. In this way we avoid recalculating the ten-fold cross-validation classification accuracy, i.e., solving ten classification problems.

3.1.2 Parameter Selection

Based on the literature and our computational experience, we choose the settings and parameters of PSO listed below.

- Neighborhood topology:

  We use von Neumann topology since it performs better than *gbest* or *lbest* with smaller neighborhood sizes in our computational experience, which also matches the results in the literature [70]. Also, we find that the inclusion or exclusion of the target particle in the neighborhood (i.e., whether the neighborhood of particle i includes particle i itself) is not critical.

- $V_{init.max}$ in the initialization step:

  During the modified PSO algorithm the v_{ij}'s usually have values around or smaller than one, so we set $V_{init.max} = 1$. There is no significant difference if we set $V_{init.max}$ larger, such as 2 or 6.

- Velocity updating formula:

  We test on both formula (3.1.1) and (3.1.2) but we prefer the latter one, the fully-informed version. This version converges slower but finds better solutions. When using formula (3.1.2), we prefer not to include particle i in its neighborhood.

- ω, c_1, and c_2 in velocity updating:

  We use $\omega = 0.7298$ and $c_1 = c_2 = 1.49618$ in (3.1.1), which are popular used in the literature [158, 128]. When we use the fully-informed version formula (3.1.2), the c_j's are chosen as $c_j = \frac{1.49618+1.49618}{|N(i)|}$ for all $j \in N(i)$, which will satisfy the convergence conditions. Under von Neumann neighborhood topology, $|N(i)|$ equals 5 or 4, depending on the inclusion/exclusion of the target particle.

- $V_{\max}$ in velocity updating:

  As described in Section 2.6.1.1, we might restrict v_{ij}'s within the range $[-V_{\max}, V_{\max}]$. Our computational experience shows that there is no need to use this restriction.

- Position updating:

  There is no significant difference to determine the zeros and ones in $\boldsymbol{x}_i$ based on (1) the values of $\boldsymbol{v}_i$, or (2) the values of $\boldsymbol{x}_i$ after executing the formula (3.1.3).

- Termination:

  We use one or both of these criteria: (1) the maximum number of iterations is achieved, and (2) the percentage of the number of particle moving is less than a threshold. The suitable maximum number of iterations varies in different applications.

- Number of particles:

  Under von Neumann neighborhood topology, the number of particles might be chosen as 9, 12, 16, 25, 36, or other numbers, depending on how large the total number of features is.

We can temporarily run PSO with a large number of iterations, observe the behavior of PSO, and then decide a suitable upper bound of the iterations or other

suitable termination criteria. We can also do similar testing runs with different number of particles, neighborhood topologies, or other settings. After deciding suitable parameters and settings from the testing runs, we use them on the same data set with more independent runs or on other similar data sets. Figure 3.1.1 demonstrates the behavior of PSO in a testing run. In this example the data set has two groups, 212 observations, and 6375 features. We select 10 features in this testing run. The PSO uses 25 particles, fully-informed velocity updating, and von Neumann neighborhood topology in which the target particle is not included in the neighborhood. The figure shows the number of particle moving, the number of objective calculation, the best accuracy, and the number of particle improving in 1000 iterations. (Note that when a particle moves from its current position, we may or may not calculate the objective function, depending on whether the new position is already recorded; when a particle moves, the best accuracy of the particle in the history may or may not improve; and when a particle improves its best accuracy, the overall best accuracy may or may not change.) We then decide the suitable termination criteria according to the behavior together with the elapsed time of the testing run and how many runs or data sets we will implement next.

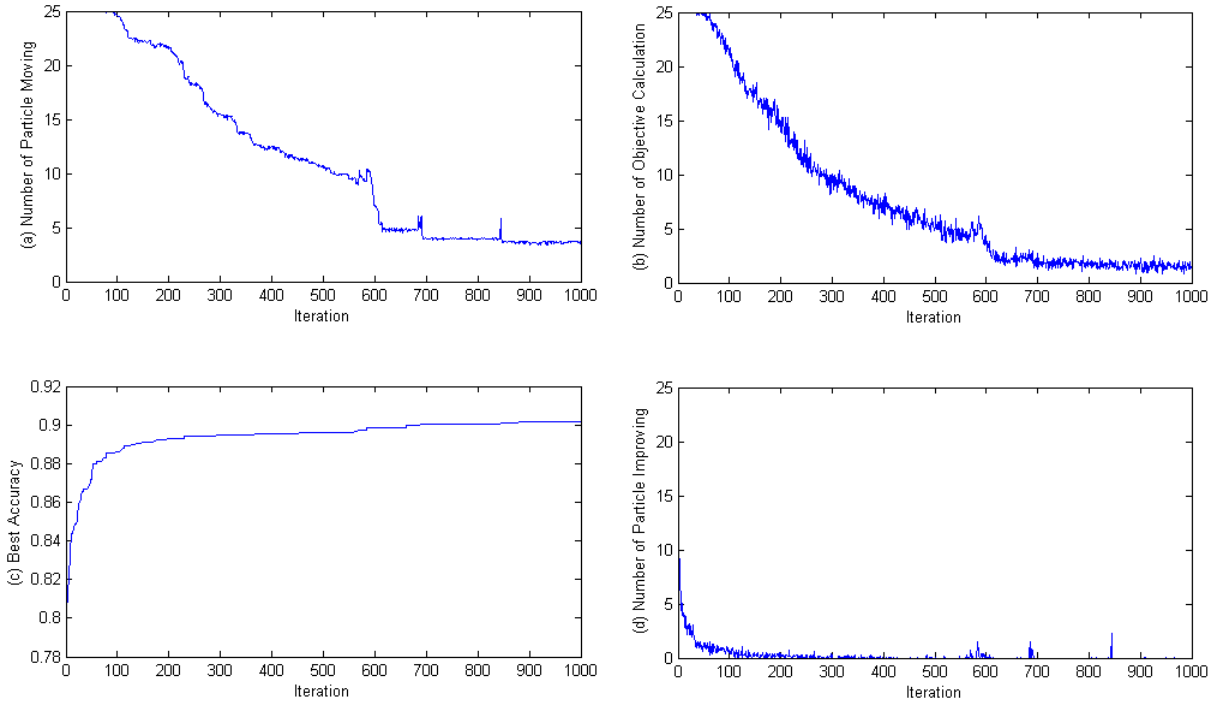

Figure 3.1.1: Behavior of PSO in an example: (a) number of particle moving, (b) number of objective calculation, (c) best accuracy, and (d) number of particle improving.

3.2 Solving Two-Group DAMIP by Exact Algorithms

We develop theories and exact algorithms to solve the two-group DAMIP problems. Here is the nonlinear two-group DAMIP formulation:

$$
\begin{aligned}
\max \quad & \sum_{i\in\mathcal{O}} u_{y_i i} \\
\text{s.t.} \quad & L_{1i} = \pi_1 f_1(\boldsymbol{x}_i) - f_2(\boldsymbol{x}_i)\lambda_{21} && \forall\ i \in \mathcal{O} \\
& L_{2i} = \pi_2 f_2(\boldsymbol{x}_i) - f_1(\boldsymbol{x}_i)\lambda_{12} && \forall\ i \in \mathcal{O} \\
& u_{ki} = \begin{cases} 1 & \text{if } k = \arg\max\{0, L_{1i}, L_{2i}\} \\ 0 & \text{otherwise} \end{cases} && \forall\ i \in \mathcal{O},\quad k \in \{0,1,2\} \\
& \sum_{k\in\{0,1,2\}} u_{ki} = 1 && \forall\ i \in \mathcal{O} \\
& \sum_{i_1:\ i_1\in\mathcal{O}_1} u_{2i_1} \le \lfloor \alpha_{12} n_1 \rfloor \\
& \sum_{i_2:\ i_2\in\mathcal{O}_2} u_{1i_2} \le \lfloor \alpha_{21} n_2 \rfloor \\
& u_{0i}, u_{1i}, u_{2i} \in \{0,1\},\quad L_{1i}, L_{2i} \text{ urs} && \forall\ i \in \mathcal{O} \\
& \lambda_{12}, \lambda_{21} \ge 0
\end{aligned}
$$

The following lemma gives basic observations of the problem and will be used for developing propositions and algorithms.

Lemma 3.2.1.

(i) Observation $i_1 \in \mathcal{O}_1$ is correctly classified if and only if

$$\frac{f_2(\boldsymbol{x}_{i_1})}{f_1(\boldsymbol{x}_{i_1})} < \frac{\pi_1 + \lambda_{12}}{\pi_2 + \lambda_{21}}$$

and

$$\pi_1 f_1(\boldsymbol{x}_{i_1}) - f_2(\boldsymbol{x}_{i_1})\lambda_{21} > 0.$$

(ii) Observation $i_2 \in \mathcal{O}_2$ is correctly classified if and only if

$$\frac{f_2(\boldsymbol{x}_{i_2})}{f_1(\boldsymbol{x}_{i_2})} > \frac{\pi_1 + \lambda_{12}}{\pi_2 + \lambda_{21}}$$

and

$$\pi_2 f_2(\boldsymbol{x}_{i_2}) - f_1(\boldsymbol{x}_{i_2})\lambda_{12} > 0.$$

(iii) Observation $i \in \mathcal{O}$ is classified in the reserved judgement region if and only if

$$\frac{\pi_1}{\lambda_{21}} < \frac{f_2(\boldsymbol{x}_i)}{f_1(\boldsymbol{x}_i)} < \frac{\lambda_{12}}{\pi_2}.$$

(iv) If λ_{12} and λ_{21} satisfy $\frac{\pi_1}{\lambda_{21}} < \frac{\lambda_{12}}{\pi_2}$, then

$$\frac{\pi_1}{\lambda_{21}} < \frac{\pi_1 + \lambda_{12}}{\pi_2 + \lambda_{21}} < \frac{\lambda_{12}}{\pi_2}.$$

Proof. (i) For $i_1 \in \mathcal{O}_1$, i_1 is correctly classified if and only if $L_{1i_1} > L_{2i_1}$ and $L_{1i_1} > 0$. Equivalently, $\pi_1 f_1(\boldsymbol{x}_{i_1}) - f_2(\boldsymbol{x}_{i_1})\lambda_{21} > \pi_2 f_2(\boldsymbol{x}_{i_1}) - f_1(\boldsymbol{x}_{i_1})\lambda_{12}$ and $\pi_1 f_1(\boldsymbol{x}_{i_1}) - f_2(\boldsymbol{x}_{i_1})\lambda_{21} > 0$. The former inequality is equivalent to $\frac{f_2(\boldsymbol{x}_{i_1})}{f_1(\boldsymbol{x}_{i_1})} < \frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$.

(ii) For $i_2 \in \mathcal{O}_2$, i_2 is correctly classified if and only if $L_{2i_2} > L_{1i_2}$ and $L_{2i_2} > 0$. Equivalently, $\pi_2 f_2(\boldsymbol{x}_{i_2}) - f_1(\boldsymbol{x}_{i_2})\lambda_{12} > \pi_1 f_1(\boldsymbol{x}_{i_2}) - f_2(\boldsymbol{x}_{i_2})\lambda_{21}$ and $\pi_2 f_2(\boldsymbol{x}_{i_2}) - f_1(\boldsymbol{x}_{i_2})\lambda_{12} > 0$. The former inequality is equivalent to $\frac{f_2(\boldsymbol{x}_{i_2})}{f_1(\boldsymbol{x}_{i_2})} > \frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$.

(iii) For $i \in \mathcal{O}$, i is classified in the reserved judgement region if and only if $L_{1i} < 0$ and $L_{2i} < 0$. Equivalently, $\pi_1 f_1(\boldsymbol{x}_i) - f_2(\boldsymbol{x}_i)\lambda_{21} < 0$ and $\pi_2 f_2(\boldsymbol{x}_i) - f_1(\boldsymbol{x}_i)\lambda_{12} < 0$. Note that both λ_{21} and λ_{12} are positive. The inequalities are equivalent to $\frac{\pi_1}{\lambda_{21}} < \frac{f_2(\boldsymbol{x}_i)}{f_1(\boldsymbol{x}_i)} < \frac{\lambda_{12}}{\pi_2}$.

(iv) $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}} - \frac{\pi_1}{\lambda_{21}} = \frac{\lambda_{12}\lambda_{21}-\pi_1\pi_2}{(\pi_2+\lambda_{21})\lambda_{21}} > 0$. Similarly, $\frac{\lambda_{12}}{\pi_2} - \frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}} = \frac{\lambda_{12}\lambda_{21}-\pi_1\pi_2}{\pi_2(\pi_2+\lambda_{21})} > 0$.

□

We then have a proposition to characterize the two-group problem without misclassification constraints.

Proposition 3.2.2. *In the two-group case without misclassification constraints, there exists an optimal solution such that at least one of λ_{12} or λ_{21} is zero. Furthermore, in this solution, no observation is classified in the reserved judgement region.*

Proof. We will prove the first part of the proposition by showing that, given any feasible solution $(\lambda_{12}, \lambda_{21}) > (0, 0)$, we can find another solution $(\bar{\lambda}_{12}, \bar{\lambda}_{21})$ in which at least one of $\bar{\lambda}_{12}$ or $\bar{\lambda}_{21}$ is zero such that all correctly-classified observations under the old solution $(\lambda_{12}, \lambda_{21})$ are still correctly classified under the new solution $(\bar{\lambda}_{12}, \bar{\lambda}_{21})$.

(i) If $\frac{\pi_1}{\pi_2} < \frac{\lambda_{12}}{\lambda_{21}}$, let $\bar{\lambda}_{21} = 0$ and let $\bar{\lambda}_{12}$ satisfy $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}} = \frac{\pi_1+\bar{\lambda}_{12}}{\pi_2}$. Simplifying the equality, we get $\bar{\lambda}_{12} = \frac{\pi_2\lambda_{12}-\pi_1\lambda_{21}}{\pi_2+\lambda_{21}}$. Note that $0 < \bar{\lambda}_{12} < \lambda_{12}$.

(ii) If $\frac{\pi_1}{\pi_2} > \frac{\lambda_{12}}{\lambda_{21}}$, let $\bar{\lambda}_{12} = 0$ and let $\bar{\lambda}_{21}$ satisfy $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}} = \frac{\pi_1}{\pi_2+\bar{\lambda}_{21}}$. Simplifying the equality, we get $\bar{\lambda}_{21} = \frac{\pi_1\lambda_{21}-\pi_2\lambda_{12}}{\pi_1+\lambda_{12}}$. Note that $0 < \bar{\lambda}_{21} < \lambda_{21}$.

(iii) If $\frac{\pi_1}{\pi_2} = \frac{\lambda_{12}}{\lambda_{21}}$, let $\bar{\lambda}_{12} = 0$ and $\bar{\lambda}_{21} = 0$.

In all cases, $(\bar{\lambda}_{12}, \bar{\lambda}_{21})$ satisfies $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}} = \frac{\pi_1+\bar{\lambda}_{12}}{\pi_2+\bar{\lambda}_{21}}$ and $(\bar{\lambda}_{12}, \bar{\lambda}_{21}) < (\lambda_{12}, \lambda_{21})$. Lemma 3.2.1(i) and (ii) contain the necessary and sufficient conditions for observations to be correctly classified. If these conditions hold under $(\lambda_{12}, \lambda_{21})$, they will still hold under $(\bar{\lambda}_{12}, \bar{\lambda}_{21})$.

If λ_{12} or λ_{21} is zero, $L_{2i} > 0$ or $L_{1i} > 0$ will hold for all $i \in \mathcal{O}$. Then no observations will be classified as reserved. This proves the second part of the proposition.

□

In the following proposition, we explain graphically the insights to develop an algorithm to solve the two-group classification problem regardless of the misclassification constraints.

Proposition 3.2.3. *We draw the observations of group 1 and 2 on the $\frac{f_2(\boldsymbol{x}_i)}{f_1(\boldsymbol{x}_i)}$-axes, as shown in Figure 3.2.1(a). Once the values of λ_{12} and λ_{21} are determined, so are the values of $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$, $\frac{\pi_1}{\lambda_{21}}$ (if exists), and $\frac{\lambda_{12}}{\pi_2}$.*

(i) If λ_{12} and λ_{21} are small enough such that $\frac{\pi_1}{\lambda_{21}} < \frac{\lambda_{12}}{\pi_2}$ is not satisfied, the lines can be partitioned at $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$ into correctly-classified regions (C) and misclassified regions (M), as described in Figure 3.2.1(b).

(ii) If λ_{12} and λ_{21} are large enough such that $\frac{\pi_1}{\lambda_{21}} < \frac{\lambda_{12}}{\pi_2}$ is satisfied, the lines can be partitioned at $\frac{\pi_1}{\lambda_{21}}$ and $\frac{\lambda_{12}}{\pi_2}$ into correctly-classified regions (C), misclassified regions (M), and reserved judgement regions (R), as described in Figure 3.2.1(c).

Proof. (i) Suppose λ_{12} and λ_{21} are small enough such that $\frac{\pi_1}{\lambda_{21}} < \frac{\lambda_{12}}{\pi_2}$ is not satisfied. By Lemma 3.2.1(iii), no reserved judgement region is constructed. The observations i_1 of group 1 in region M satisfy $\frac{f_2(\boldsymbol{x}_{i_1})}{f_1(\boldsymbol{x}_{i_1})} > \frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$; by Lemma 3.2.1(i), they are not correctly classified. Since there is no reserved judgement region, they are misclassified. The observations i_1 of group 1 in region C satisfy $\frac{f_2(\boldsymbol{x}_{i_1})}{f_1(\boldsymbol{x}_{i_1})} < \frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$, or equivalently, $L_{1i_1} > L_{2i_1}$. Since there is no reserved judgement region, $L_{1i_1} > 0$ holds. Therefore the observations i_1 of group 1 in region C are correctly classified. Proofs for observations of group 2 are similar.

(ii) Suppose λ_{12} and λ_{21} are large enough such that $\frac{\pi_1}{\lambda_{21}} < \frac{\lambda_{12}}{\pi_2}$ is satisfied. Lemma 3.2.1(iv) describes the order of $\frac{\pi_1}{\lambda_{21}}$, $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$, and $\frac{\lambda_{12}}{\pi_2}$. By Lemma 3.2.1(iii), the region R's are exactly the reserved judgement regions. The observations i_1 of group 1 in region M satisfy $\frac{f_2(\boldsymbol{x}_{i_1})}{f_1(\boldsymbol{x}_{i_1})} > \frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$; they are not reserved nor correctly classified (by Lemma 3.2.1(i)), so they are misclassified. The observations i_1 of group 1 in region C satisfy $\frac{f_2(\boldsymbol{x}_{i_1})}{f_1(\boldsymbol{x}_{i_1})} < \frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$, or equivalently, $L_{1i_1} > L_{2i_1}$. Since these observations are not reserved, $L_{1i_1} > 0$ holds. Therefore the observations i_1 of group 1 in region C are correctly classified. Proofs for observations of group 2 are similar.

□

Based on Proposition 3.2.3, we give some detailed ideas of developing an algorithm in the following remarks.

Remarks:

1. The misclassified group-one observations will be the rightmost ones on the $\frac{f_2(\boldsymbol{x}_i)}{f_1(\boldsymbol{x}_i)}$ axis; similarly, the misclassified group-two observations will be the leftmost ones. In the problem with misclassification constraints, we can point out the observations which are allowed to be misclassified. Figure 3.2.2 demonstrates an

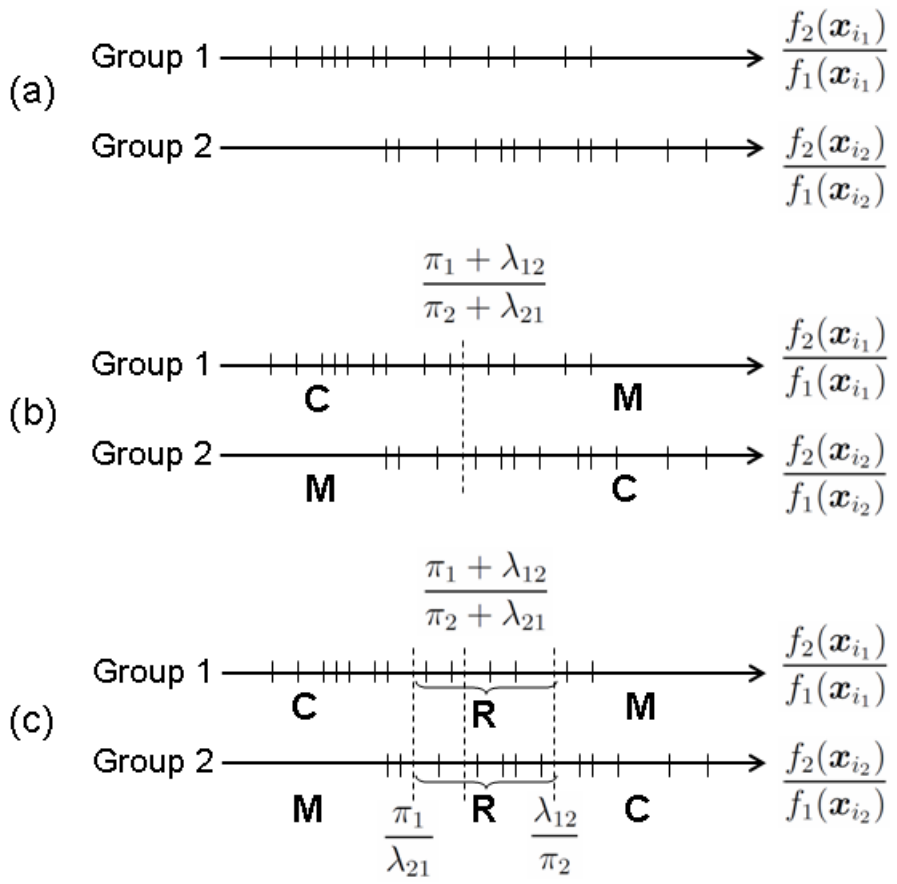

Figure 3.2.1: Two-group problem on the $\frac{f_2(\boldsymbol{x}_i)}{f_1(\boldsymbol{x}_i)}$ axis. (a) Representation of observations. (b) The lines are partitioned at $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$ into correctly-classified regions (C) and misclassified regions (M). (c) The lines are partitioned at $\frac{\pi_1}{\lambda_{21}}$ and $\frac{\lambda_{12}}{\pi_2}$ into correctly-classified regions (C), misclassified regions (M), and reserved judgement regions (R).

example of the two-group problem with misclassification constrains; the bolder points are the observations allowed to be misclassified. The maximum numbers of allowable misclassification of group one and two are $\lfloor \alpha_{12} n_1 \rfloor$ and $\lfloor \alpha_{21} n_2 \rfloor$, respectively. All non-bold observations have to be correctly classified or reserved.

2. The reserved judgement region is constructed when λ_{12} and λ_{21} are large enough to form the interval $(\frac{\pi_1}{\lambda_{21}}, \frac{\lambda_{12}}{\pi_2})$. Once the interval is generated, all observations in this interval are classified as reserved, no matter whether they could potentially

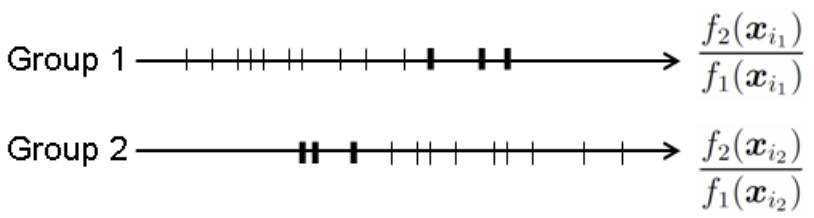

Figure 3.2.2: Representation of observations allowed to be misclassified.

be classified correctly or not. Since the objective function is to maximize the number of corrected-classified observations, we don't necessarily need the reserved judgement region unless misclassification constraints cannot be satisfied without the reserved judgement region.

Figure 3.2.3 shows the cases in which the reserved judgement region is not or is needed. Let a_1 be the largest $\frac{f_2(\boldsymbol{x}_{i_1})}{f_1(\boldsymbol{x}_{i_1})}$-value of the group-one observations i_1's which are not allowed to be misclassified. Similarly, let a_2 be the smallest $\frac{f_2(\boldsymbol{x}_{i_1})}{f_1(\boldsymbol{x}_{i_1})}$-value of the group-two observations i_2's which are not allowed to be misclassified.

- If $a_1 < a_2$ as shown in Figure 3.2.3(a), then any $(\lambda_{12}, \lambda_{21})$ which satisfies $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}} = t \in (a_1, a_2)$ and is small enough to not form the interval $(\frac{\pi_1}{\lambda_{21}}, \frac{\lambda_{12}}{\pi_2})$ is a feasible solution which meets the misclassification constrains but does not use the reserved judgement region. In this case we don't need the reserved judgement region.
- If $a_1 \geq a_2$ as shown in Figure 3.2.3(b), then the misclassification constrains cannot be satisfied without the reserved judgement region. In this case, we determine the optimal interval (r_L, r_R) as the reserved judgement region and get $(\lambda_{12}, \lambda_{21})$ by solving $\frac{\pi_1}{\lambda_{21}} = r_L$ and $\frac{\lambda_{12}}{\pi_2} = r_R$. The optimal r_L is chosen to be *on the immediate left* of a_2 (i.e., r_L is on the left of a_2 but also on the right of all points which are on the left of a_2); the optimal r_R is chosen

to be on the immediate right of a_1. Choosing r_L and r_R in this way not only meets the misclassification constraints but also guarantee optimality, i.e., leave as many observations in the correctly-classified region as possible.

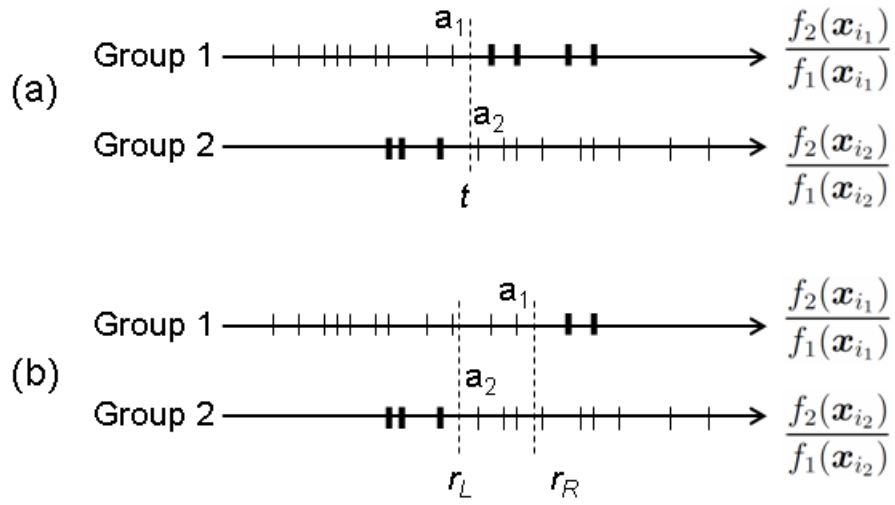

Figure 3.2.3: Cases in which reserved judgement region is not or is needed. (a) $a_1 < a_2$. Reserved judgement region is not needed. (b) $a_1 \geq a_2$. Reserved judgement region is needed.

3. Suppose we don't need the reserved judgement region to satisfy the misclassification constraints no matter weather there are misclassification constraints or not. In this case we only need to determine the optimal value of $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$. Once the value of $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$ is determined, the value of $(\lambda_{12}, \lambda_{21})$ can be chosen so that at least one of them is zero, which guarantees to not form the reserved judgement region.

   We then determine an interval, (l_L, l_R), in which the optimal value of $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$ will be located. There are four cases as shown in Figure 3.2.4.

   - Case (a): No misclassification constraints. Let l_L be on the immediate left

of the leftmost group-two observation and let l_R be on the immediate right of the rightmost group-one observation.

- Case (b): A misclassification constraints on group one. Let l_L be on the immediate right of a_1 and let l_R be on the immediate right of the rightmost group-one observation.
- Case (c): A misclassification constraints on group two. Let l_L be on the immediate left of the leftmost group-two observation and let l_R be on the immediate left of a_2.
- Case (d): Misclassification constraints on both groups with $a_1 < a_2$. Let l_L be on the immediate right of a_1 and let l_R be on the immediate left of a_2.

It is sufficient to search for optimal $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$ in the interval (l_L, l_R) defined above. For example, in case (a), if the value of $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$ goes leftward beyond l_L, the number of correctly-classified group-one observations may decrease without any gain on the number of correctly-classified group-two observations. This shows that l_L is a lower bound of the optimal $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$ in this case.

4. When we search for the optimal value of $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$ in the interval (l_L, l_R), we only have to check the objective value at the positions where the closest observation on the left is of group one and the closest observation on the right is of group two. Figure 3.2.5 demonstrates an example in which we only need to check at positions $t_1, \cdots, t_5$. At other positions, the value of $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$ can either go leftward to classify more group-two observations correctly or go rightward to classify more group-one observations correctly without affecting any observations already classified correctly.

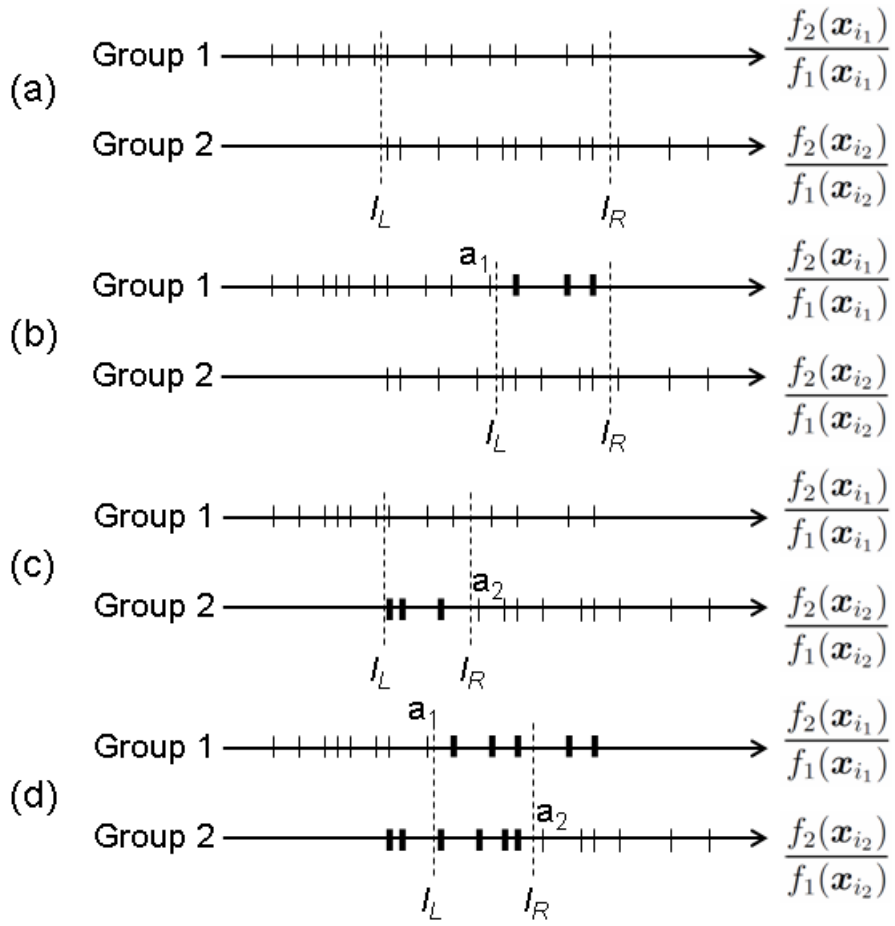

Figure 3.2.4: The interval (l_L, l_R) in which optimal $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$ is located. (a) No misclassification constraints. (b) A constraint on group 1. (c) A constraint on group 2. (d) Constraints on both groups with $a_1 < a_2$.

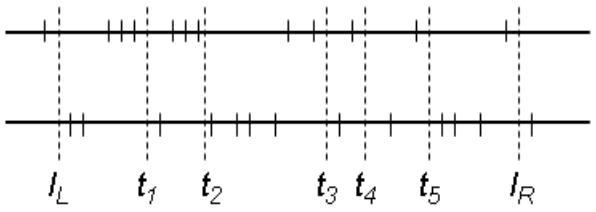

Figure 3.2.5: Positions to check the optimal value of $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$ within (l_L, l_R).

Algorithm 3.2.4. Exact algorithm for the two-group problem.

1. Sort group-one and group-two observations separately in ascending order by the sort key $\frac{f_2(\boldsymbol{x}_i)}{f_1(\boldsymbol{x}_i)}$.

2. Find a_1 and a_2 if there is misclassification constraints on group one and two, respectively (See Remark 2).

3. (The case of using the reserved judgement region.)

   If there are misclassification constraints on both groups and if $a_1 \geq a_2$:

   (a) Choose r_L and r_R (See Remark 2 / Figure 3.2.3(b)).

   (b) Calculate λ_{21} and λ_{21} (See Remark 2).

   (c) Stop.

4. (The case of not using the reserved judgement region.)

   (a) Choose l_L and l_R (See Remark 3 / Figure 3.2.4).

   (b) Check which possible position within (l_L, l_R) for the value of $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$ has the maximum number of correctly-classified observations (See Remark 4). Decide the value of $\frac{\pi_1+\lambda_{12}}{\pi_2+\lambda_{21}}$.

   (c) Calculate λ_{21} and λ_{21} (See Remark 3).

The complexity of step 1 is $O(n \log n)$ and the complexity of other steps is $O(n)$ or $O(1)$, so the overall complexity of Algorithm 3.2.4 is $O(n \log n)$. Besides, the remarks give explanation for the correctness of the algorithm.

Based on Proposition 3.2.2, we develop an alternative exact algorithm for the two-group problem without misclassification limits. The idea of this algorithm is as the following. Assuming $\lambda_{21} = 0$, we find the optimal value of λ_{12}; assuming $\lambda_{12} = 0$, we find the optimal value of λ_{21}; the better case of these two is optimal overall.

Assume $\lambda_{21} = 0$, $L_{1i} > L_{2i}$ is equivalent to $\pi_2 \frac{f_2(\boldsymbol{x}_i)}{f_1(\boldsymbol{x}_i)} - \pi_1 < \lambda_{12}$. We draw the $\pi_2 \frac{f_2(\boldsymbol{x}_i)}{f_1(\boldsymbol{x}_i)} - \pi_1$-values of the observations and conclude that the value of λ_{12} determines

the correctly-classified regions and misclassified regions of both groups, as shown in Figure 3.2.6. Therefore we can find the best position of λ_{12} in the similar way as in the previous algorithm. Since λ_{12} is nonnegative, we consider only the observations with positive $\pi_2 \frac{f_2(\boldsymbol{x}_i)}{f_1(\boldsymbol{x}_i)} - \pi_1$-values. Under the assumption of $\lambda_{12} = 0$, the analysis is symmetric. Here is the algorithm.

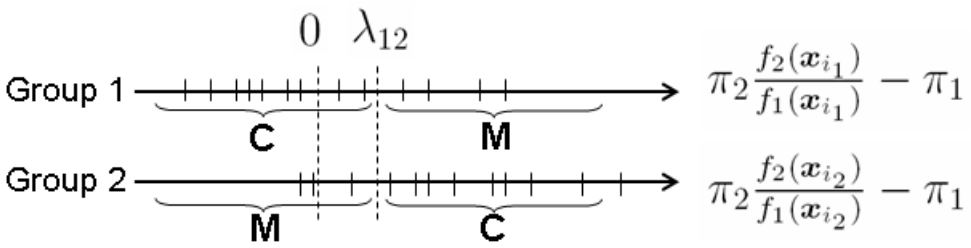

Figure 3.2.6: Two-group problem on the $\pi_2 \frac{f_2(\boldsymbol{x}_i)}{f_1(\boldsymbol{x}_i)} - \pi_1$ axis given that $\lambda_{21} = 0$. The lines are partitioned at λ_{12} into correctly-classified regions (C) and misclassified regions (M).

Algorithm 3.2.5. Exact algorithm for the two-group problem without misclassification limits.

1. (Case: Assume $\lambda_{21} = 0$, find the optimal λ_{12}.)

   (a) Sort group-one and group-two observations separately in ascending order by the sort key $\pi_2 \frac{f_2(\boldsymbol{x}_i)}{f_1(\boldsymbol{x}_i)} - \pi_1$. Only observations with positive sort key values are sorted and stored.

   (b) Check which possible position for the value of λ_{12} has the maximum number of correctly-classified observations by counting the number of group-one observations with sort key values less than λ_{12} and the number of group-two observations with sort key values greater than λ_{12}.

2. (Case: Assume $\lambda_{12} = 0$, find the optimal λ_{21}.)

(a) Sort group-one and group-two observations separately in ascending order by the sort key $\pi_1 \frac{f_1(\boldsymbol{x}_i)}{f_2(\boldsymbol{x}_i)} - \pi_2$. Only observations with positive sort key values are sorted and stored.

(b) Check which possible position for the value of λ_{21} has the maximum number of correctly-classified observations by counting the number of group-one observations with sort key values greater than λ_{21} and the number of group-two observations with sort key values less than λ_{21}.

3. The case which has larger number of correctly-classified observations determines the optimal λ_{12} and λ_{21}.

The complexity of Algorithm 3.2.5 is $O(n \log n)$, same as that of Algorithm 3.2.4. Although Algorithm 3.2.5 has to do two sorting instead of one, it handles in the sorting only the observations with positive sort key values, which enhances the speed. In the situation without misclassification constraints, our computational experience shows that the computational time of these two algorithms has no significant difference.

Algorithm 3.2.4 and 3.2.5 can be generalized to solve the DAMIP in which the objective function is modified to be maximizing the average group accuracy (i.e., formula (1.3.13)) or maximizing the minimum group accuracy (i.e., formula (1.3.14)-(1.3.16)) instead of the original one, maximizing the total number of correctly-classified observations (i.e., formula (1.3.8)). The generalization is in step 4(b) of Algorithm 3.2.4 and step 1(b) and 2(b) of Algorithm 3.2.5: Check which possible position has the maximum value of the objective function, which can be the number of correctly-classified observations, the average group accuracy, or the minimum group accuracy, depending on which model we are using. In fact, the algorithms give the optimal

solutions no matter which objective function is used.

Furthermore, when checking which possible position has the maximum value of the objective function, we can also incorporate a secondary objective function. That is, when two different solutions tie under the primary objective, we compare the solutions by the secondary one.

3.3 Solving DAMIP without Misclassification Constraints by Greedy Algorithm

We develop a greedy algorithm to solve the multigroup DAMIP problem without misclassification constraints. We also provide computational results to show the performance of the algorithm.

3.3.1 Introduction

The idea of the greedy algorithm for multigroup DAMIP is to separate observations from only two groups in a single step and handle all pairs of groups in turn. When we separate group h and k in a step, we search for good λ_{hk} or λ_{kh} while letting other λ's remain unchanged; λ_{hk} or λ_{kh} are main factors for separating group h and k although other λ's also have effects. The search for a good λ_{hk} or λ_{kh} is similar to Algorithm 3.2.5 and is based on the fact that $L_{hi} > L_{ki}$ is equivalent to $(\pi_k + \lambda_{kh})\frac{f_k(\boldsymbol{x}_i)}{f_h(\boldsymbol{x}_i)} + \sum_{j\in\mathcal{G}, j\neq h,k}(\lambda_{jh} - \lambda_{jk})\frac{f_j(\boldsymbol{x}_i)}{f_h(\boldsymbol{x}_i)} - \pi_h < \lambda_{hk}$. Here is the greedy algorithm.

Algorithm 3.3.1. Greedy algorithm for the multigroup problem without misclassification limits.

Input: $(h_1, k_1), (h_2, k_2), \cdots, (h_T, k_T)$, where $h_t, k_t \in \mathcal{G},\ h_t \neq k_t$ for all $t = 1, \cdots, T$.

Initialization: $\bar{\lambda}_{hk} = 0$ for all $h, k \in \mathcal{G},\ h \neq k$.

$\bar{z} = 0$.

Loop $t = 1$ to T

1. (Case: Find the pseudo-optimal $\lambda_{h_t k_t}$ while other λ's are fixed at $\bar{\lambda}$'s.)

   (a) Sort the observations of group h_t and k_t separately in ascending order by the sort key $(\pi_{k_t} + \bar{\lambda}_{k_t h_t})\frac{f_{k_t}(\boldsymbol{x}_i)}{f_{h_t}(\boldsymbol{x}_i)} + \sum_{j\in\mathcal{G}, j\neq h_t, k_t}(\bar{\lambda}_{jh_t} - \bar{\lambda}_{jk_t})\frac{f_j(\boldsymbol{x}_i)}{f_{h_t}(\boldsymbol{x}_i)} - \pi_{h_t}$. Only observations with positive sort key values are sorted and stored.

(b) Check which possible position for the value of $\lambda_{h_t k_t}$ has the maximum greedy objective value defined by the sum of the number of group-h_t observations with sort key values less than $\lambda_{h_t k_t}$ and the number of group-k_t observations with sort key values greater than $\lambda_{h_t k_t}$. Let $\hat{\lambda}_{h_t k_t}$ be an optimal solution.

(c) Calculate the true objective value (i.e., the number of correctly-classified observations in all groups), $\hat{z}_1$, using $\bar{\lambda}$'s with $\hat{\lambda}_{h_t k_t}$ instead of $\bar{\lambda}_{h_t k_t}$.

2. (Case: Find the pseudo-optimal $\lambda_{k_t h_t}$ while other λ's are fixed at $\bar{\lambda}$'s.)

(a) Sort the observations of group k_t and h_t separately in ascending order by the sort key $(\pi_{h_t} + \bar{\lambda}_{h_t k_t}) \frac{f_{h_t}(\boldsymbol{x}_i)}{f_{k_t}(\boldsymbol{x}_i)} + \sum_{j \in \mathcal{G}, j \neq k_t, h_t} (\bar{\lambda}_{j k_t} - \bar{\lambda}_{j h_t}) \frac{f_j(\boldsymbol{x}_i)}{f_{k_t}(\boldsymbol{x}_i)} - \pi_{k_t}$. Only observations with positive sort key values are sorted and stored.

(b) Check which possible position for the value of $\lambda_{k_t h_t}$ has the maximum greedy objective value defined by the sum of the number of group-k_t observations with sort key values less than $\lambda_{k_t h_t}$ and the number of group-h_t observations with sort key values greater than $\lambda_{k_t h_t}$. Let $\hat{\lambda}_{k_t t_t}$ be an optimal solution.

(c) Calculate the true objective value (i.e., the number of correctly-classified observations in all groups), $\hat{z}_2$, using $\bar{\lambda}$'s with $\hat{\lambda}_{k_t h_t}$ instead of $\bar{\lambda}_{k_t h_t}$.

3. If $\max\{\hat{z}_1, \hat{z}_2\} > \bar{z}$, then

(a) Update $\bar{z}$ by $\max\{\hat{z}_1, \hat{z}_2\}$.

(b) If $\hat{z}_1 > \hat{z}_2$, update $\bar{\lambda}_{h_t k_t}$ by $\hat{\lambda}_{h_t k_t}$; otherwise update $\bar{\lambda}_{k_t h_t}$ by $\hat{\lambda}_{k_t h_t}$.

End Loop

To generalize the greedy algorithm for alternative objective functions such as maximizing the average group accuracy (i.e., formula (1.3.13)) or maximizing the

minimum group accuracy (i.e., formula (1.3.14)-(1.3.16)), the alternative objective functions are used when we calculate the true objective value in step 1(c) and 2(c) of Algorithm 3.3.1.

The greedy algorithm can easily be modified for the multigroup DAMIP problem with misclassification constraints. Suppose the constraint $\sum_{i:\ i\in\mathcal{O}_h} u_{ki} \le \lfloor \alpha_{hk} n_h \rfloor$ is not satisfied by the current solution. In this case, we increase λ_{hk} and decrease λ_{kh} to make this constraint valid.

3.3.2 Computational Study—Three-Group Case

The first part of the computational study is for the case $K = 3$. Guidelines for input parameter selection are provided based on numerous computational results from simulated data sets. Solutions obtained by the greedy algorithm are also compared with those by CPLEX.

3.3.2.1 Input Parameter Selection

In this section we study how to choose good input parameters, (h_1, k_1), (h_2, k_2), $\cdots$, (h_T, k_T) in the three-group case. Intuitively we have to consider all pairs of groups, but the order of the pairs could matter. The best order of group-pairs could be affected by the distances between groups and the sizes of groups. Furthermore, we could handle a group-pair more than once, i.e., having $(h_{t_1}, k_{t_1}) = (h_{t_2}, k_{t_2})$ for some $t_1 \neq t_2$. We perform computational study in the three-group case to find the best strategies, i.e., the choices of group-pairs, when distances between groups or group sizes vary.

The design of the computational study follows the procedures in [84, 18, 19].

Table 3.3.1: Configurations in settings of the computational study

Settings	Means			Mahalanobis distances		
	Group 1	Group 2	Group 3	d(1,2)	d(1,3)	d(2,3)
A1 $\sim$ A7	(0, 0)	(1, 0)	(0.5, 0.8660)	1	1	1
B1 $\sim$ B7	(0, 0)	(2, 0)	(1, 1.7321)	2	2	2
C1 $\sim$ C7	(0, 0)	(3, 0)	(1.5, 2.5981)	3	3	3
D1 $\sim$ D15	(0, 0)	(1, 0)	(0.5, 1.4142)	1	1.5	1.5
E1 $\sim$ E15	(0, 0)	(2, 0)	(1, 2.8284)	2	3	3
F1 $\sim$ F15	(0, 0)	(1, 0)	(0.5, 1.9365)	1	2	2
G1 $\sim$ G15	(0, 0)	(1.5, 0)	(0.75, 2.9047)	1.5	3	3
H1 $\sim$ H15	(0, 0)	(1.5, 0)	(0.75, 0.6614)	1.5	1	1
I1 $\sim$ I15	(0, 0)	(3, 0)	(1.5, 1.3229)	3	2	2
J1 $\sim$ J15	(0, 0)	(1.8, 0)	(0.9, 0.4359)	1.8	1	1
K1 $\sim$ K15	(0, 0)	(2.7, 0)	(1.35, 0.6538)	2.7	1.5	1.5
L1 $\sim$ L25	(0, 0)	(2, 0)	(0.6875, 0.7262)	2	1	1.5
M1 $\sim$ M25	(0, 0)	(3, 0)	(1.0733, 0.5367)	3	1.2	2

Different distances between groups and different group sizes are designed in settings from A1 to M25. In each setting, 200 simulated data sets are generated. In 100 data sets, the observations of each group are generated from bivariate normal distributions (i.e., the number of features is two) in which the means are given in Table 3.3.1 and the covariance matrices are 2-by-2 identity matrices. In the other 100 data sets, the observations of each group are generated from contaminated normal distribution in which 10% of the observations is generated from a normal distribution with the covariance matrix multiplied by 100 while any other design remains the same. In each setting, the Mahalanobis distances between groups are shown in Table 3.3.1 and Figure 3.3.1. The Mahalanobis distance between group i and j is

$$d(i,j) = \sqrt{(m_i - m_j)^T S^{-1} (m_i - m_j)}, \tag{3.3.1}$$

where m_i, m_j are the group means and S is the common covariance matrix. Table 3.3.2 shows the sizes of groups in each setting.

We solve the DAMIP instance of each data set by Algorithm 3.3.1 using 42 different

Table 3.3.2: Group sizes in settings of the computational study

Settings	Number of observations		
	Group 1	Group 2	Group 3
A1, B1, C1	100	100	100
A2, B2, C2	100	100	150
A3, B3, C3	100	150	150
A4, B4, C4	100	150	200
A5, B5, C5	30	30	300
A6, B6, C6	30	300	300
A7, B7, C7	30	100	300
D1, E1, F1, G1, H1, I1, J1, K1	100	100	100
D2, E2, F2, G2, H2, I2, J2, K2	100	150	100
D3, E3, F3, G3, H3, I3, J3, K3	100	100	150
D4, E4, F4, G4, H4, I4, J4, K4	150	150	100
D5, E5, F5, G5, H5, I5, J5, K5	100	150	150
D6, E6, F6, G6, H6, I6, J6, K6	100	150	200
D7, E7, F7, G7, H7, I7, J7, K7	100	200	150
D8, E8, F8, G8, H8, I8, J8, K8	150	200	100
D9, E9, F9, G9, H9, I9, J9, K9	30	300	30
D10, E10, F10, G10, H10, I10, J10, K10	30	30	300
D11, E11, F11, G11, H11, I11, J11, K11	300	300	30
D12, E12, F12, G12, H12, I12, J12, K12	30	300	300
D13, E13, F13, G13, H13, I13, J13, K13	30	100	300
D14, E14, F14, G14, H14, I14, J14, K14	30	300	100
D15, E15, F15, G15, H15, I15, J15, K15	100	300	30
L1, M1	100	100	100
L2, M2	100	100	150
L3, M3	100	150	100
L4, M4	150	100	100
L5, M5	100	150	150
L6, M6	150	100	150
L7, M7	150	150	100
L8, M8	100	150	200
L9, M9	100	200	150
L10, M10	150	100	200
L11, M11	150	200	100
L12, M12	200	100	150
L13, M13	200	150	100
L14, M14	30	30	300
L15, M15	30	300	30
L16, M16	300	30	30
L17, M17	30	300	300
L18, M18	300	30	300
L19, M19	300	300	30
L20, M20	30	100	300
L21, M21	30	300	100
L22, M22	100	30	300
L23, M23	100	300	30
L24, M24	300	30	100
L25, M25	300	100	30

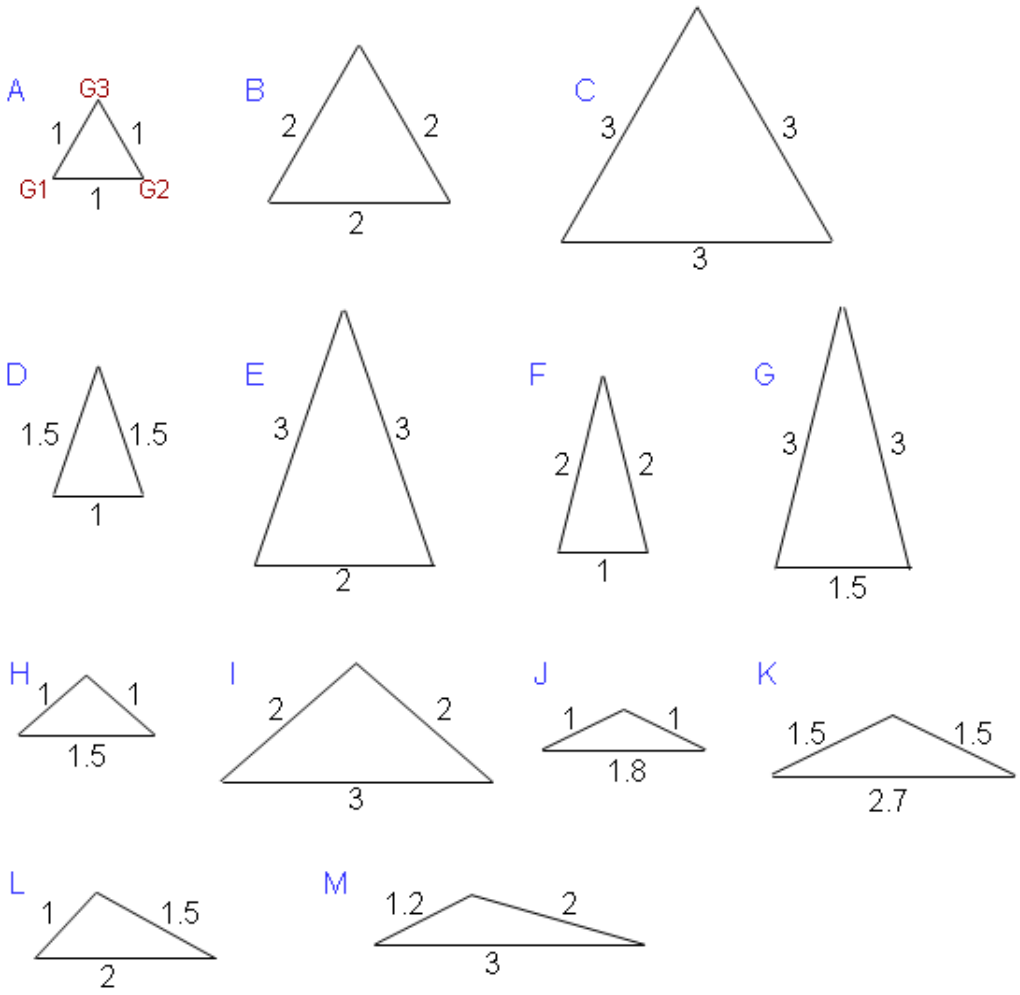

Figure 3.3.1: Configurations in settings of the computational study.

strategies, in which 6 are "one-round" and the other 36 are "two-round". We abbreviate the notation of the strategy (the input of Algorithm 3.3.1), (h_1, k_1), (h_2, k_2), $\cdots$, (h_T, k_T), to $h_1k_1 - h_2k_2 - \cdots - h_Tk_T$. The six one-round strategies are 12-13-23, 12-23-13, 13-12-23, 13-23-12, 23-12-13, and 23-13-12, all possible permutations of group-pairs in which each group-pair appears once. The two-round strategies are the combinations of a one-round strategy followed by a one-round strategy, for example, 12-13-23-12-13-23.

We have 191 settings (A1∼M25), 200 data sets in each setting (half from normal

and half from contaminated normal), and 44 runs for each data set (6 using one-round strategy and 36 using two-round strategy). To reiterate, a run means solving an DAMIP instance using the greedy algorithm with a certain strategy.

First of all we analyze the computational time. The computational time is one second or less in all the runs. Most of the time it is less than one second.

Secondly, we analyze how frequently an observation is classified as reserved. Larger λ's will create larger reserved judgement region and force more observations into that region. In the case without misclassification constraints, we don't want the reserved region too large. In the classification results, 0.19% of the runs have observations classified as reserved. Among the runs which have reserved observations, the average proportion of observations classified as reserved is 0.0065; among all runs, the average proportion of observations classified as reserved is 0.000012. These numbers show that the λ's obtained by the greedy method are not too large to create large reserved judgement region. Besides analyzing among all runs, we also analyze among runs with one-round strategy and with two-round strategy separately. The results are summarized in Table 3.3.3. One-round strategies classify fewer observations as reserved.

Thirdly, we analyze how much the classification accuracy improves by using two-round strategy rather than one-round strategy. The classification accuracy is the number of correctly-classified observations divided by the number of observations. The classification accuracy of any two-round-strategy run is compared with that of its corresponding one-round-strategy run (for example, the corresponding one-round strategy of 12-13-23-23-12-13 is 12-13-23, the first three group-pairs) and the improvement is measured by the difference between these two accuracies. The results show

Table 3.3.3: Results about reserved observations in the computational study

	All runs	Runs with one-round strategy	Runs with two-round strategy
Percentage of runs having reserved observations	0.19%	0.09%	0.20%
Average proportion of reserved observations over runs having reserved observations	0.0065	0.0060	0.0066
Average proportion of reserved observations over runs	0.000012	0.000006	0.000013

that about 25% of the two-round-strategy runs improve their one-round counterparts while 75% do not improve. Among the 191×200 data sets, about 62% of them have at least one improved two-round-strategy run while 38% do not have any. The average improved accuracy over all two-round-strategy runs is 0.0023, and the average improved accuracy over all improved two-round-strategy runs is 0.0092. In summary, the two-round strategy does not improve the classification accuracy much. Therefore we will focus on the one-round strategies in the remaining analysis.

The forth part of the analysis of the computational results is on which one-round strategy performs better when distance between groups and group sizes vary. For each setting, we treat the data sets generated from normal distribution and contaminated normal distribution separately. Table 3.3.5 lists good strategies in each setting, in which the letter "n" and "c" in each setting indicate normal and contaminated normal, respectively, and the content of each strategy index is shown in Table 3.3.4. Good strategies are defined as the following. Each setting under normal or contaminated normal consists of 100 data sets, and the average accuracies (over 100 data sets) of the one-round strategies are sorted in descending order. We determine which strategies other than the highest-accurate one are also good enough, i.e., there is no

Table 3.3.4: List of one-round strategies

Strategy index	Strategy
1	12-13-23
2	12-23-13
3	13-12-23
4	13-23-12
5	23-12-13
6	23-13-12

significant difference between the accuracies between that strategy and the highest-accurate strategy, by performing the paired t-test (100 samples, $\alpha = 0.10$, two-tail). The best strategy together with the ones whose accuracies are not significant different from the best one are all considered as good strategies. Instead of using $\alpha = 0.10$, similar analysis is done using 0.05 and 0.20. From Table 3.3.5 we find that in some settings there exist obviously good strategies, and furthermore, these good strategies for normal and contaminated normal data sets match each other. For example, setting A4n and A4c, A7n and A7c, and so on.

Besides, we find that all settings contain common good strategies in normal and contaminated normal data sets except setting H4, K10, and M4. When good strategies are obtained by using $\alpha = 0.20$ in paired t-test, only 6 settings have no common good strategies; when using $\alpha = 0.05$, all settings have common good strategies. Therefore in the following analysis we will not distinguish between normal or contaminated normal data sets.

We categorize the settings into scenarios based on distances between groups and group sizes and look for the best strategies for each scenario. In Table 3.3.6, each row represents one scenario characterized by some settings. Note that in the table the "distances between groups" column and the "group sizes" column do not fully

Table 3.3.5: Good strategies in each setting

Setting	Strategy index	Setting	Strategy index	Setting	Strategy index	Setting	Strategy index	Setting	Strategy index	Setting	Strategy index
A1n	1 2 3 4 5 6	D12n	3 4 2 1	F14n	3 4 2	I1n	5 2 3 6 1 4	K3n	2 1 6 5 3	L20n	4 6 1 2 5
A1c	1 2 3 4 6 5	D12c	1 2 4 3	F14c	2 3	I1c	1 6 2 3 4 5	K3c	6 4 1	L20c	4 1 2
A2n	4 6	D13n	4 2	F15n	2 5 3 4	I2n	5 2	K4n	4 1 3 6 2 5	L21n	2
A2c	4 6 1	D13c	4	F15c	5 3 2	I2c	6 2 5	K4c	2 6 1	L21c	2
A3n	2 1 4 5 6	D14n	3 4 2	G1n	3 4 2 1 5	I3n	6 2 1	K5n	2 1 4 6 5	L22n	4 6
A3c	1 5 2 4 6	D14c	2	G1c	5 4 3 6 2 1	I3c	4 1 2	K5c	6	L22c	6 1
A4n	4	D15n	5 2	G2n	2 5	I4n	4 5 1 3 6 2	K6n	4 6 1	L23n	5
A4c	4	D15c	5	G2c	2 1 5 3	I4c	6 4 3 1	K6c	4 3 1 2 6 5	L23c	5 6
A5n	6 4 1 2	E1n	4 3 5 2 1 6	G3n	4 6 5 1 2 3	I5n	2 3 5 6 1 4	K7n	2 5	L24n	1 3
A5c	6 4	E1c	1 6 4 5	G3c	4 6 3	I5c	6 5 2	K7c	2 6 5	L24c	1
A6n	2 4 3	E2n	2 3	G4n	3 5 4 6 1 2	I6n	4 1 2	K8n	5	L25n	6 3
A6c	4 3 1 2	E2c	5 2 1 3	G4c	1 3 2 6 4 5	I6c	1 4 6 2 5	K8c	2 6 5 4	L25c	3 4
A7n	4 2	E3n	4 6 3	G5n	2 4 3 6 5 1	I7n	5 2 4	K9n	5	M1n	6 5 4 2 3 1
A7c	4	E3c	4 6	G5c	6 4 5 2 1	I7c	2	K9c	5	M1c	4 1
B1n	3 4 1 2 6 5	E4n	1 2 5 3 6	G6n	4 6 1 5 2	I8n	5 2 4	K10n	4 6	M2n	6 4 5 3
B1c	3 1 4 2	E4c	1 5 3 2 6	G6c	4 6 3	I8c	5 2	K10c	2 1	M2c	4 6 2 1
B2n	4 6 1 2 3	E5n	5 4 6 2 3	G7n	2	I9n	5 4	K11n	5 4 3	M3n	2 5 6
B2c	6 4	E5c	1 2 3 5 4 6	G7c	2 3 5 6 4	I9c	5 4	K11c	6 3 5 4	M3c	6 2 5
B3n	2 1 4 3 5 6	E6n	4 6 3 2 5 1	G8n	3 5 2 6	I10n	4 6 1 2	K12n	1 2 4 3	M4n	3
B3c	6 5 4 2	E6c	4 5	G8c	5	I10c	1 4 2	K12c	4 3	M4c	1
B4	4 1 2	E7	2	G9	2 5 3	I11	4	K13	4 6 1 2	M5	2 5 6
B4c	6 4 3	E7c	2 5 6	G9c	3 2	I11c	6 5 4 3	K13c	4 1 2	M5c	2
B5	1 2 4 6	E8	2 5 3 4 1 6	G10	6	I12	4 2	K14	2 4	M6	1 4 2 6 5
B5c	6 4	E8c	5 1 6 4	G10c	4 6	I12c	2 4 1 3	K14c	2	M6c	4 1
B6	2 4 3 1	E9	2 5 3	G11	3 4 5 6	I13	4 6 1 2	K15	4 5	M7	4 1 6 2
B6c	4 2 3 1	E9c	2 3 4	G11c	5 3	I13c	4 1 2	K15c	5	M7c	6 2 4 1 3
B7	4	E10	6 4	G12	2 4 3 1	I14	2 4 5	L1	1 5 4 2 3	M8	2 1 4 5 3
B7c	4	E10c	6 4	G12c	2 1	I14c	2	L1c	6 5 4 1 3	M8c	1 2 6 4
C1	2 1 3 5 6 4	E11	5 6	G13	4	I15	4 5	L2	6 4 1	M9	2 5 6
C1c	1	E11c	5 6 3 4	G13c	4 3	I15c	5 6	L2c	6 1 4 5 2	M9c	2
C2	4 3 6 1 2 5	E12	1 2	G14	3 2	J1	2 1 6 5 3 4	L3	5 2 6	M10	1 4 6
C2c	4 2 3 6 1 5	E12c	2 1 3 4	G14c	1 2 3	J1c	2 1 5 6 3 4	L3c	6 5 2	M10c	4 6
C3	6 2 3 1 5	E13	4 2 1	G15	2 5 3	J2	5 2	L4	3 1	M11	6 5 2 4
C3c	2 3 5 1 4 6	E13c	4 3	G15c	3 5 2	J2c	5 6	L4c	1 3 4 2 6	M11c	6 2 5
C4	6 2 4 3 1 5	E14	2 3	H1	1 2 5 6 3 4	J3	4 6 1	L5	4 1 6 2	M12	3 1
C4c	6 4	E14c	3 2 1	H1c	5 6 4 1 2 3	J3c	4 6 2 1	L5c	2 1 4 3	M12c	4 1 3
C5	6 4 2 1	E15	3 4 5 2	H2	2 5	J4	3 4 2 5	L6	1 3 4 6	M13	3 4 6
C5c	4 6 1	E15c	5 2	H2c	5 6 2	J4c	4 3 6	L6c	4 1 2	M13c	4 3
C6	1 3 4 2	F1	3 5 4 6 1 2	H3	4 1 3 6 2	J5	2 6 4 1	L7	1 2 6 5 4 3	M14	5 4 6 1 2
C6c	2 3 4 1	F1c	4 3 6 5 2 1	H3c	4 3 1	J5c	2 5 6	L7c	5 6 2 4 3 1	M14c	1 4
C7	4	F2	2 3	H4	5 2	J6	4 1 2	L8	4 1 6 3	M15	5
C7c	4	F2c	2 1 5	H4c	6	J6c	6 4	L8c	4 1 2	M15c	5
D1	1 3 4 6	F3	6 4	H5	2 4 1 5	J7	2 5 6	L9	2	M16	3
D1c	3 5 6 2 4	F3c	4 6 3 5	H5c	6 4 2 5 1 3	J7c	2 6	L9c	2	M16c	3
D2	2 5 3	F4	5 3	H6	1 4 6 2 3	J8	5	L10	6 4 2 1	M17	1 4 3
D2c	2 3	F4c	5 2	H6c	4 6 3	J8c	5	L10c	6 4	M17c	3 4
D3	6 4	F5	2 5 6 3 4	H7	2	J9	5	L11	5 3 4	M18	5 6 1 2
D3c	6	F5c	3 2 5	H7c	2 1	J9c	5	L11c	6 5	M18c	6 5 1
D4	3 5 1 4	F6	4	H8	5	J10	1 2 4 6 3	L12	1	M19	6 5 4 3
D4c	2 5 1 3	F6c	4 6 3	H8c	5	J10c	4 6 3 1 5 2	L12c	1 3 4 2	M19c	3 4 6
D5	5 4 2 6	F7	2 3	H9	5 4	J11	6 5 3 4	L13	3 6	M20	1 2 4 6 5
D5c	6 5 2 4 3 1	F7c	2 1	H9c	5	J11c	3 4 5 6	L13c	3 6 4 1	M20c	4 1
D6	4	F8	5 2	H10	4 6 1 2	J12	3 4 2 1	L14	4 6 1 2	M21	2 4 1
D6c	4	F8c	5 2 3	H10c	6 4 5 2 1	J12c	3 4	L14c	4 3	M21c	2
D7	2 3	F9	3 2 5 4	H11	3	J13	4 6 1 2 5	L15	5	M22	4 6 1 2
D7c	2	F9c	2 3	H11c	5 6 4 3	J13c	4 2 1	L15c	5	M22c	1 2 4 6
D8	5 3 2 4	F10	6 4	H12	2 3 4	J14	2	L16	6 3	M23	5
D8c	5 2 1	F10c	6 4	H12c	3 4 2 1	J14c	2	L16c	3	M23c	5
D9	3 2 5 4	F11	5 4 3 6	H13	1 2 4 6	J15	5	L17	4 3 1 2	M24	3 1
D9c	2 1	F11c	3 4 5	H13c	4 2	J15c	5	L17c	3 4 1	M24c	1
D10	4 6 2	F12	3 4 2 1	H14	2 4	K1	6 1 5 4 2	L18	6 1 5 2	M25	6 3
D10c	4 6	F12c	1 2 3	H14c	2 4	K1c	4 5 2 6 1	L18c	5 6	M25c	3 4
D11	4 3	F13	4	H15	3 5	K2	5 2 4 6 3	L19	5 3 6 4		
D11c	3 4 5 6	F13c	4	H15c	5	K2c	2 6 5 4	L19c	5 6 3 4		

describe a scenario; for example, two long distances and one large group size can result in two situations: The large group can adjacent to two long distances or to one long and one short distance. However, settings in a row characterize a unique scenario.

Table 3.3.6 shows the probability of being a good strategy. Take the first scenario (setting A1, B1, C1) and strategy 2 as an example, strategy 2 appears five times in the good strategy list of the six setting A1n, A1c, B1n, B1c, C1n, C1n (see Table 3.3.5), so its probability of being a good strategy is $\frac{5}{6} = 0.8333$. In each scenario, we put '*' to indicate the strategy with highest probability and the strategies whose probabilities do not significantly differ from the highest one. The two-proportion z-test ($\alpha = 0.1$, two-tail) is performed here to help for indicating relatively large probabilities although the assumption of the test, certain sizes greater than five, may not satisfy.

We draw the conclusion of using certain strategies in certain occasions by checking the information in Table 3.3.6 and the same kind of information when we use $\alpha = 0.20$ and $\alpha = 0.05$ in the paired t-test described above. In summary, the group size is more important than the distance between groups as conditions of a data set to determine a dominant strategy of the greedy algorithm. The suggested strategies under certain conditions of the data sets are given in Table 3.3.7. The conditions include that one group has notably larger size (say group 1, as seen in Table 3.3.7), two groups have notably larger size, and three groups have notably different sizes. #Gk denotes the size of group k and $d(h, k)$ denotes the distance between group h and k. When none of these conditions are satisfied, there might not exist dominant strategies, and one practical way is to run all six one-round strategies and pick the best solution.

Table 3.3.6: Probabilities of being good strategies in each scenario

Scenario		Settings	Probability of being a good strategy for each strategy index '*' indicates relatively large probabilities					
Distances between groups	Group sizes		1	2	3	4	5	6
same	same	A1 B1 C1	*1.0000	*0.8333	*0.8333	*0.8333	*0.6667	*0.6667
same	1 large	A2 A5 B2 B5 C2 C5	0.6667	0.5000	0.2500	*1.0000	0.1667	*1.0000
same	2 large	A3 A6 B3 B6 C3 C6	*0.8333	*1.0000	0.7500	*0.9167	0.5000	0.5000
same	diff.	A4 A7 B4 B7 C4 C7	0.1667	0.2500	0.1667	*1.0000	0.0833	0.2500
2 long	same	D1 E1 F1 G1	*0.8750	*0.7500	*0.8750	*1.0000	*0.8750	*0.8750
2 long	1 large	D2 D9 E2 E9 F2 F9 G2 G9	0.2500	*1.0000	0.8125	0.1875	0.5625	0.0000
2 long	1 large	D3 D10 E3 E10 F3 F10 G3 G10	0.0625	0.1250	0.2500	*0.8750	0.1250	*1.0000
2 long	2 large	D4 D11 E4 E11 F4 F11 G4 G11	0.3750	0.3750	*0.8750	0.5625	*0.9375	0.5625
2 long	2 large	D5 D12 E5 E12 F5 F12 G5 G12	0.7500	*1.0000	0.7500	0.7500	0.5000	0.4375
2 long	diff.	D6 D13 E6 E13 F6 F13 G6 G13	0.1875	0.2500	0.3125	*1.0000	0.1875	0.2500
2 long	diff.	D7 D14 E7 E14 F7 F14 G7 G14	0.1875	*1.0000	0.6250	0.1875	0.1250	0.1250
2 long	diff.	D8 D15 E8 E15 F8 F15 G8 G15	0.1875	0.8125	0.5625	0.3125	*1.0000	0.1875
1 long	same	H1 I1 J1 K1	*1.0000	*1.0000	*0.7500	*1.0000	*1.0000	*1.0000
1 long	1 large	H2 H9 I2 I9 J2 J9 K2 K9	0.0000	0.4375	0.0625	0.3125	*1.0000	0.3125
1 long	1 large	H3 H10 I3 I10 J3 J10 K3 K10	*0.9375	*0.7500	0.3125	*0.8125	0.1875	*0.7500
1 long	2 large	H4 H11 I4 I11 J4 J11 K4 K11	0.2500	0.3125	*0.7500	*0.7500	*0.6250	*0.6875
1 long	2 large	H5 H12 I5 I12 J5 J12 K5 K12	*0.5625	*0.8125	*0.5625	*0.8125	0.3750	0.4375
1 long	diff.	H6 H13 I6 I13 J6 J13 K6 K13	0.8125	0.8125	0.1875	*1.0000	0.1875	0.6250
1 long	diff.	H7 H14 I7 I14 J7 J14 K7 K14	0.0625	*1.0000	0.0000	0.3125	0.3125	0.1875
1 long	diff.	H8 H15 I8 I15 J8 J15 K8 K15	0.0000	0.1875	0.0625	0.2500	*1.0000	0.1250
diff.	same	L1 M1	*1.0000	*0.5000	*0.7500	*1.0000	*0.7500	*0.5000
diff.	1 large	L2 L14 M2 M14	*0.7500	0.5000	0.2500	*1.0000	0.3750	*0.7500
diff.	1 large	L3 L15 M3 M15	0.0000	0.5000	0.0000	0.0000	*1.0000	0.5000
diff.	1 large	L4 L16 M4 M16	0.3750	0.1250	*0.8750	0.1250	0.0000	0.2500
diff.	2 large	L5 L17 M5 M17	*0.6250	*0.6250	*0.6250	*0.7500	0.1250	0.2500
diff.	2 large	L6 L18 M6 M18	*0.8750	*0.5000	0.1250	*0.5000	*0.6250	*0.7500
diff.	2 large	L7 L19 M7 M19	0.5000	0.5000	*0.8750	*1.0000	0.6250	*1.0000
diff.	diff.	L8 L20 M8 M20	*1.0000	*0.7500	0.2500	*1.0000	0.3750	0.5000
diff.	diff.	L9 L21 M9 M21	0.1250	*1.0000	0.0000	0.1250	0.1250	0.1250
diff.	diff.	L10 L22 M10 M22	0.6250	0.3750	0.0000	*0.8750	0.0000	*1.0000
diff.	diff.	L11 L23 M11 M23	0.0000	0.2500	0.1250	0.2500	*1.0000	0.5000
diff.	diff.	L12 L24 M12 M24	*1.0000	0.1250	0.6250	0.2500	0.0000	0.0000
diff.	diff.	L13 L25 M13 M25	0.1250	0.0000	*1.0000	0.6250	0.0000	0.6250

Table 3.3.7: Suggested strategies of the greedy algorithm given certain conditions of the data sets

Conditions	Suggested strategy
#G1 > #G2, #G3	12-13-23 or 13-12-23 (If $d(1,2) < d(1,3)$, choose 12-13-23; if $d(1,3) < d(1,2)$, choose 13-12-23.)
#G1, #G2 > #G3	13-12-23 or 23-12-13 (If $d(1,3) < d(2,3)$, choose 13-12-23; if $d(2,3) < d(1,3)$, choose 23-12-13.)
#G1 > #G2 > #G3	13-12-23

3.3.2.2 Compared with CPLEX

We compare the solutions obtained by the greedy algorithm with those obtained by running CPLEX (IBM ILOG CPLEX version 12.2). In CPLEX we use default settings except that we set the thread number to be one, in which case the running is less affected by other jobs in the workstation so the computational time is more comparable with each other. We also set the time limit to 3600 seconds; if CPLEX cannot solve an instance in one hour, we terminate it and use the best solution. In each setting, instead of running all 100 data sets, we only run the first five data sets by CPLEX.

It is interesting to observe that CPLEX solves instances of data sets from normal distribution much better than those from contaminated normal. Among the 955 normal data sets, CPLEX solves 379 of them to optimality in an hour; the mean (standard deviation) of the computational time of the 379 sets is 512 (819) seconds. Among the 955 contaminated normal data sets, CPLEX solves only 165 sets to optimality in an hour; the mean (standard deviation) of the computational time of the 165 sets is 1342 (1024).

The comparison between solutions by CPLEX and by the greedy algorithm is shown in Table 3.3.8. The solution by the greedy algorithm is represented by the best solution among six one-round-strategy solutions. Solutions by CPLEX are categorized into three situations: solved to optimality within an hour, not optimal but better than solution by greedy, and not optimal but worse than solution by greedy. The accuracy difference is calculated by accuracy of CPLEX minus accuracy of greedy and gap is calculated by accuracy difference over accuracy of CPLEX. The results show that the greedy algorithm provides good solutions for the DAMIP problem.

Table 3.3.8: Comparison between solutions by CPLEX and by the greedy algorithm

Situation of solution by CPLEX	Counts	Accuracy difference Mean (Std.)	Gap Mean (Std.)
Solution is optimal	544	0.0053 (0.0074)	0.62% (0.86%)
Solution is better	1031	0.0098 (0.0096)	1.39% (1.45%)
Solution is worse	335	—	—

3.3.3 Computational Study—More-Than-Three-Group Case

The second part of the computational study is for the case $K > 3$. Computational study is performed on four medical/biological data sets from UCI machine learning repository [37], including data set "Dermatology", "Ecoli", "Heart Disease", and "Nursery". For each data set, we (1) choose a suitable one-round strategy, i.e., order of all group-pairs, as the input parameter of the greedy algorithm (Algorithm 3.3.1); (2) compare the DAMIP solutions solved by the greedy algorithm and by CPLEX; and (3) compare the classification results by DAMIP, Bayes classifier, and DALP.

We use CPLEX to solve DALP to optimality with input parameter $(c_1, c_2) = (1, 0)$, $(2, 1)$, $(1, 2)$, and $(0, 1)$ suggested in [84]. Recall that c_1 emphasizes on correctly classifying the observations and c_2 emphasizes on placing observations in the reserved judgement region. From the results we observe that when c_2 is relatively large, there could be too many observations placed in the reserved judgement region. For the ease of comparison, we will show the results using $(c_1, c_2) = (1, 0)$.

3.3.3.1 Data Set "Dermatology"

The six groups of the "Dermatology" data set represent different diagnosis of erythemato-squamous diseases, including psoriasis, seboreic dermatitis, lichen planus, pityriasis

Table 3.3.9: "Dermatology": Mahalanobis distances between groups

	p.	s.d.	l.p.	p.r.	c.d.	p.r.p.
p.	–	8.17	20.96	8.68	11.14	17.17
s.d.	–	–	19.74	3.52	9.55	16.01
l.p.	–	–	–	19.74	19.93	24.92
p.r.	–	–	–	–	9.71	16.24
c.d.	–	–	–	–	–	17.24
p.r.p.	–	–	–	–	–	–

rosea, cronic dermatitis, and pityriasis rubra pilaris. The total number of the observations is 366, the numbers of observations in each group are 112, 61, 72, 49, 52, and 20, and the number of features is 34. Few missing values are replaced by their group means. Table 3.3.9 shows the Mahalanobis distances between groups (see Equation (3.3.1)). We observe that all groups are far from each other, indicating that this is an easy classification problem.

For choosing a suitable one-round strategy, we start to test on all permutations of group-pairs. However, we observe that all group-pairs encountered give the same objective values, so we stop trying all permutations and arbitrarily choose 12-13-14-15-16-23-24-25-26-34-35-36-45-46-56 as the input of the algorithm.

The greedy algorithm and CPLEX give exactly the same objective value 361 (or 0.9863 when divided by the number of observations). Both ways are solved in one second. Here we run CPLEX using default settings with maximum eight threads.

The ten-fold cross-validation classification results by DAMIP (solved by CPLEX), DAMIP (solved by greedy algorithm), Bayes, and DALP ($(c_1, c_2) = (1, 0)$, solved by CPLEX) are shown in Table 3.3.10 to 3.3.13. In summary, the overall accuracies are 0.9754, 0.9699, 0.9672, and 0.9672, respectively. The solution times are 4, 3, 1, and 3

Table 3.3.10: "Dermatology": Classification results by DAMIP solved by CPLEX

Ten-fold cross-validation

	p.	s.d.	l.p.	p.r.	c.d.	p.r.p.	p.	s.d.	l.p.	p.r.	c.d.	p.r.p.
p.	112	0	0	0	0	0	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
s.d.	0	59	0	2	0	0	0.0000	0.9672	0.0000	0.0328	0.0000	0.0000
l.p.	0	0	71	0	1	0	0.0000	0.0000	0.9861	0.0000	0.0139	0.0000
p.r.	0	5	0	44	0	0	0.0000	0.1020	0.0000	0.8980	0.0000	0.0000
c.d.	0	0	0	0	52	0	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
p.r.p.	0	1	0	0	0	19	0.0000	0.0500	0.0000	0.0000	0.0000	0.9500
Overall accuracy: 0.9754												

Table 3.3.11: "Dermatology": Classification results by DAMIP solved by greedy algorithm

Ten-fold cross-validation

	p.	s.d.	l.p.	p.r.	c.d.	p.r.p.	p.	s.d.	l.p.	p.r.	c.d.	p.r.p.
p.	111	1	0	0	0	0	0.9911	0.0089	0.0000	0.0000	0.0000	0.0000
s.d.	0	58	0	3	0	0	0.0000	0.9508	0.0000	0.0492	0.0000	0.0000
l.p.	0	0	71	0	1	0	0.0000	0.0000	0.9861	0.0000	0.0139	0.0000
p.r.	0	5	0	44	0	0	0.0000	0.1020	0.0000	0.8980	0.0000	0.0000
c.d.	0	0	0	0	52	0	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
p.r.p.	0	1	0	0	0	19	0.0000	0.0500	0.0000	0.0000	0.0000	0.9500
Overall accuracy: 0.9699												

seconds. CPLEX are run in default settings with maximum one thread.

3.3.3.2 Data Set "Ecoli"

The original "Ecoli" data set consists of eight groups, representing different localization sites of protein. The eight groups are cp (cytoplasm), im (inner membrane without signal sequence), pp (perisplasm), imU (inner membrane, uncleavable signal sequence), om (outer membrane), omL (outer membrane lipoprotein), imL (inner membrane lipoprotein), and imS (inner membrane, cleavable signal sequence); the

Table 3.3.12: "Dermatology": Classification results by Bayes

Ten-fold cross-validation

	p.	s.d.	l.p.	p.r.	c.d.	p.r.p.	p.	s.d.	l.p.	p.r.	c.d.	p.r.p.
p.	111	1	0	0	0	0	0.9911	0.0089	0.0000	0.0000	0.0000	0.0000
s.d.	0	55	0	6	0	0	0.0000	0.9016	0.0000	0.0984	0.0000	0.0000
l.p.	0	0	71	0	1	0	0.0000	0.0000	0.9861	0.0000	0.0139	0.0000
p.r.	0	3	0	46	0	0	0.0000	0.0612	0.0000	0.9388	0.0000	0.0000
c.d.	0	0	0	0	52	0	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
p.r.p.	0	1	0	0	0	19	0.0000	0.0500	0.0000	0.0000	0.0000	0.9500
Overall accuracy: 0.9672												

Table 3.3.13: "Dermatology": Classification results by DALP

Ten-fold cross-validation

	p.	s.d.	l.p.	p.r.	c.d.	p.r.p.	p.	s.d.	l.p.	p.r.	c.d.	p.r.p.
p.	111	1	0	0	0	0	0.9911	0.0089	0.0000	0.0000	0.0000	0.0000
s.d.	0	55	0	6	0	0	0.0000	0.9016	0.0000	0.0984	0.0000	0.0000
l.p.	0	0	71	0	1	0	0.0000	0.0000	0.9861	0.0000	0.0139	0.0000
p.r.	0	3	0	46	0	0	0.0000	0.0612	0.0000	0.9388	0.0000	0.0000
c.d.	0	0	0	0	52	0	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
p.r.p.	0	1	0	0	0	19	0.0000	0.0500	0.0000	0.0000	0.0000	0.9500
Overall accuracy: 0.9672												

Table 3.3.14: "Ecoli": Mahalanobis distances between groups

	cp	im	pp	imU	om
cp	–	4.61	3.79	4.93	5.70
im	–	–	4.32	2.00	5.66
pp	–	–	–	4.31	3.55
imU	–	–	–	–	5.56
om	–	–	–	–	–

numbers of observations of each group are 143, 77, 52, 35, 20, 5, 2, and 2, respectively. Since the sizes of the last three groups are very small, we only consider the first five groups. When the group size shrinks to five, one of the seven predictive features has identical value in all observations and thus eliminated. Therefore, the data set we are using consists of 5 groups, 327 observations, and 6 features. Table 3.3.14 shows the Mahalanobis distances between groups.

We run all $\binom{5}{2}! = 3628800$ one-round strategies for input parameter of the greedy algorithm, which is done in 4285 seconds. The objective values of all one-round strategies are one of the three values: 297, 299, and 300, as seen in the histogram in Figure 3.3.2. The histogram shows that if we only run part of the permutations of group-pairs as the one-round strategy, we have a high probability to get the best one; even if we are unlucky to get the best one, the other two objective values are also close. Among the best results we choose 34-23-24-25-12-13-14-15-35-45 for further use on this data set.

The greedy algorithm gives the objective value 300 (or 0.9174 as the accuracy) in less than one second; CPLEX gives 301 (or 0.9205 as the accuracy) in one second. Here we run CPLEX using default settings with maximum eight threads.

The ten-fold cross-validation classification results by DAMIP (solved by CPLEX),

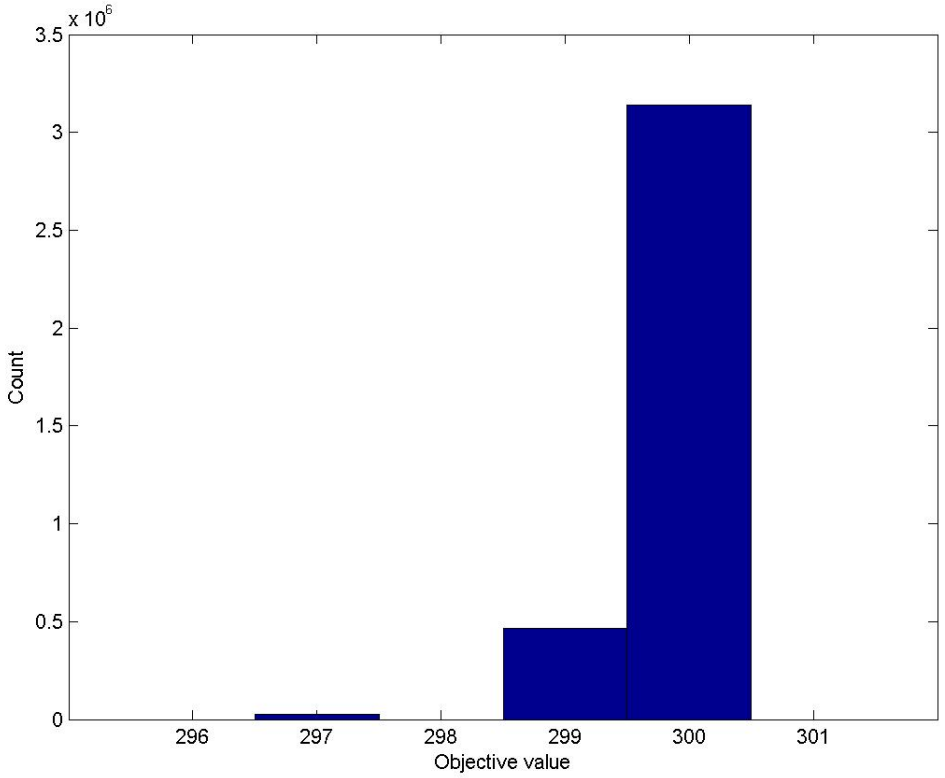

Figure 3.3.2: "Ecoli": Histogram of the objective values of all one-round strategies for the greedy algorithm

DAMIP (solved by greedy algorithm), Bayes, and DALP ($(c_1, c_2) = (1, 0)$, solved by CPLEX) are shown in Table 3.3.15 to 3.3.18. In summary, the overall accuracies are 0.8869, 0.8838, 0.8746, and 0.8899, respectively. The solution times are 28, < 1, < 1, and 1 seconds. CPLEX are run in default settings with maximum one thread.

Besides using all features in the computational experiment, we also run different classification methods using all possible subsets of the features in which the sizes of the subsets are greater than or equal to two. Figure 3.3.3 and 3.3.4 demonstrate the overall accuracy and minimum group accuracy, respectively, of the ten-fold cross-validation results. Each tick of the x-axis represents a unique feature subset, and from left to right the sizes of the subsets increase from two to six. The methods include

Table 3.3.15: "Ecoli": Classification results by DAMIP solved by CPLEX

Ten-fold cross-validation

	cp	im	pp	imU	om	cp	im	pp	imU	om
cp	138	1	4	0	0	0.9650	0.0070	0.0280	0.0000	0.0000
im	2	59	2	14	0	0.0260	0.7662	0.0260	0.1818	0.0000
pp	3	2	45	1	1	0.0577	0.0385	0.8654	0.0192	0.0192
imU	1	5	0	29	0	0.0286	0.1429	0.0000	0.8286	0.0000
om	0	0	1	0	19	0.0000	0.0000	0.0500	0.0000	0.9500
Overall accuracy: 0.8869										

Table 3.3.16: "Ecoli": Classification results by DAMIP solved by greedy algorithm

Ten-fold cross-validation

	cp	im	pp	imU	om	cp	im	pp	imU	om
cp	138	1	4	0	0	0.9650	0.0070	0.0280	0.0000	0.0000
im	2	59	2	14	0	0.0260	0.7662	0.0260	0.1818	0.0000
pp	3	1	46	1	1	0.0577	0.0192	0.8846	0.0192	0.0192
imU	1	5	0	29	0	0.0286	0.1429	0.0000	0.8286	0.0000
om	0	0	2	1	17	0.0000	0.0000	0.1000	0.0500	0.8500
Overall accuracy: 0.8838										

Table 3.3.17: "Ecoli": Classification results by Bayes

Ten-fold cross-validation

	cp	im	pp	imU	om	cp	im	pp	imU	om
cp	138	1	4	0	0	0.9650	0.0070	0.0280	0.0000	0.0000
im	2	56	1	18	0	0.0260	0.7273	0.0130	0.2338	0.0000
pp	3	1	43	1	4	0.0577	0.0192	0.8269	0.0192	0.0769
imU	1	3	0	31	0	0.0286	0.0857	0.0000	0.8857	0.0000
om	0	0	1	1	18	0.0000	0.0000	0.0500	0.0500	0.9000
Overall accuracy: 0.8746										

Table 3.3.18: "Ecoli": Classification results by DALP

Ten-fold cross-validation

	cp	im	pp	imU	om	cp	im	pp	imU	om
cp	138	1	4	0	0	0.9650	0.0070	0.0280	0.0000	0.0000
im	2	59	2	14	0	0.0260	0.7662	0.0260	0.1818	0.0000
pp	3	1	46	0	2	0.0577	0.0192	0.8846	0.0000	0.0385
imU	1	4	0	30	0	0.0286	0.1143	0.0000	0.8571	0.0000
om	0	0	2	0	18	0.0000	0.0000	0.1000	0.0000	0.9000
Overall accuracy: 0.8899										

Bayes, DALP ($(c_1, c_2) = (1, 0)$, solved by CPLEX using one thread), DAMIP (solved by greedy algorithm), and DAMIP with maximizing-minimum-group-accuracy objective (solved by greedy algorithm). The choice of input parameter for DAMIP with the alternative objective is done by testing all possible one-round strategies, too. The computational times are 1, 20, 3, and 4 seconds, respectively. We see that DAMIP generally gives the best overall accuracy but it can have low minimum group accuracy; DAMIP with maximizing-minimum-group-accuracy objective overcomes the drawback and still gives good overall accuracies.

3.3.3.3 Data Set "Heart Disease"

The "Heart Disease" data set consists of five groups, denoted by 0, 1, 2, 3, and 4, which represent different levels of diagnosis of heart disease (larger number means more severe). The total number of observations is 920, coming from four locations; the numbers in each group are 411, 196, 135, 135, and 43, respectively. Two of the 13 used features are discarded since they have missing values in more than half of all observations; we use the remaining 11 features and replace any missing value by the group mean. Table 3.3.19 shows the Mahalanobis distances between groups. We observe that the groups are close to each other, indicating that this is a hard classification problem.

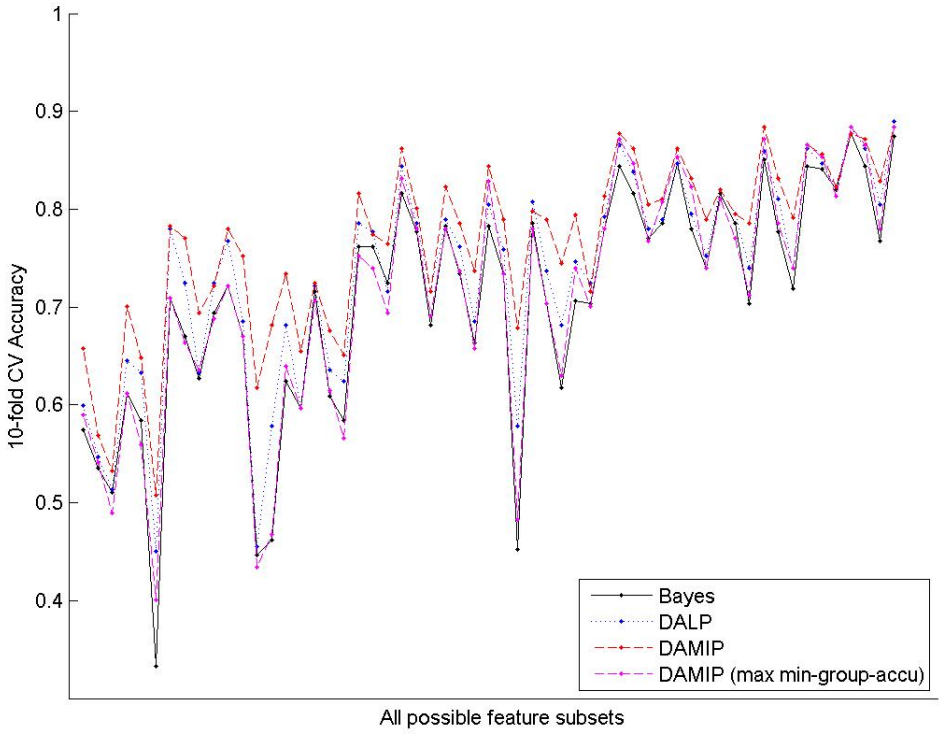

Figure 3.3.3: "Ecoli": 10-fold CV accuracy on all possible subsets of the features

We run all $\binom{5}{2}! = 3628800$ one-round strategies for input parameter of the greedy algorithm, which is done in 7840 seconds. The objective values fall in the interval $[518, 550]$, as seen in the histogram in Figure 3.3.5. Among the best results we choose 15-24-35-13-14-25-12-23-34-45 for further use.

The greedy algorithm gives the objective value 550 (or 0.5978 as the accuracy) in less than one second. CPLEX cannot get the optimal solution in 24 hours using default settings with maximum eight threads. The best objective value obtained in 24 hours is 482 (or 0.5239 as the accuracy) with a 65.73% gap. Note that the objective value obtained by the greedy algorithm with any one-round strategy is better than this one from CPLEX.

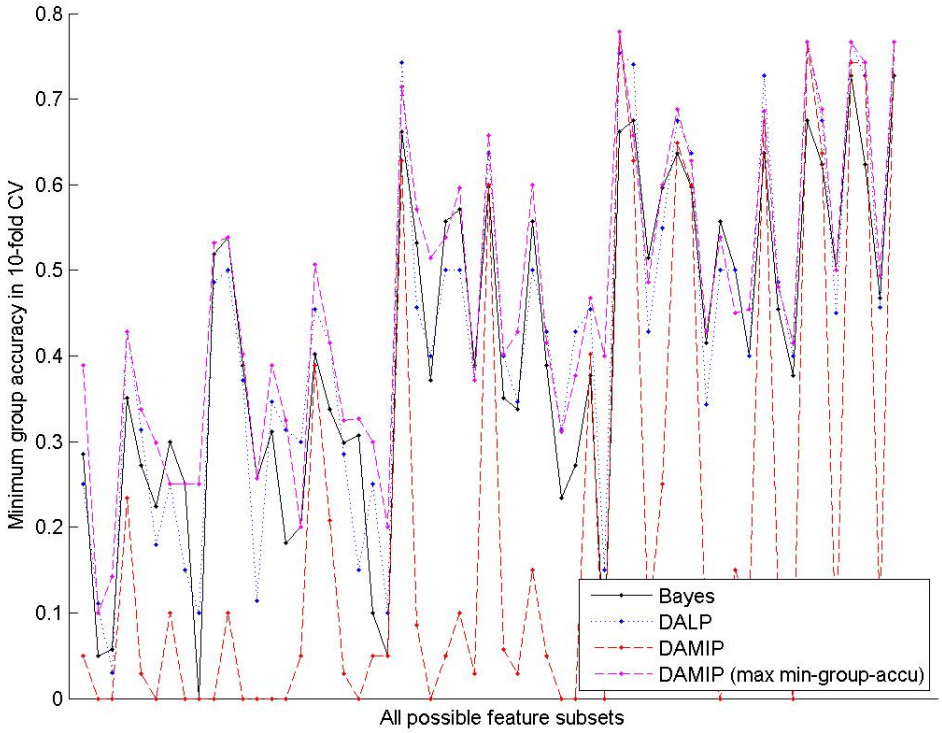

Figure 3.3.4: "Ecoli": Minimum group accuracy (10-fold CV) on all possible subsets of the features

The ten-fold cross-validation classification results by DAMIP (solved by greedy algorithm), Bayes, and DALP ($(c_1, c_2) = (1, 0)$, solved by CPLEX) are shown in Table 3.3.20 to 3.3.22. In summary, the overall accuracies are 0.5435, 0.5141, and 0.5326, respectively. The solution times are 1, 1, and 4 seconds. CPLEX are run in default settings with maximum one thread.

There are groups whose group accuracies are very low, so we try alternative objective functions in the greedy algorithms, including maximizing the average group accuracy (i.e., formula (1.3.13)) and maximizing the minimum group accuracy (i.e.,

Table 3.3.19: "Heart Disease": Mahalanobis distances between groups

	0	1	2	3	4
0	–	1.58	2.04	2.24	2.58
1	–	–	0.65	1.04	1.65
2	–	–	–	0.53	1.18
3	–	–	–	–	0.97
4	–	–	–	–	–

Table 3.3.20: "Heart Disease": Classification results by DAMIP solved by greedy algorithm

Ten-fold cross-validation

	0	1	2	3	4	0	1	2	3	4
0	331	55	9	14	2	0.8054	0.1338	0.0219	0.0341	0.0049
1	50	87	31	28	0	0.2551	0.4439	0.1582	0.1429	0.0000
2	18	52	32	26	7	0.1333	0.3852	0.2370	0.1926	0.0519
3	14	37	29	44	11	0.1037	0.2741	0.2148	0.3259	0.0815
4	5	11	9	12	6	0.1163	0.2558	0.2093	0.2791	0.1395
Overall accuracy: 0.5435										

Table 3.3.21: "Heart Disease": Classification results by Bayes

Ten-fold cross-validation

	0	1	2	3	4	0	1	2	3	4
0	319	59	7	15	11	0.7762	0.1436	0.0170	0.0365	0.0268
1	41	68	31	29	27	0.2092	0.3469	0.1582	0.1480	0.1378
2	12	41	21	25	36	0.0889	0.3037	0.1556	0.1852	0.2667
3	8	25	19	41	42	0.0593	0.1852	0.1407	0.3037	0.3111
4	2	6	3	8	24	0.0465	0.1395	0.0698	0.1860	0.5581
Overall accuracy: 0.5141										

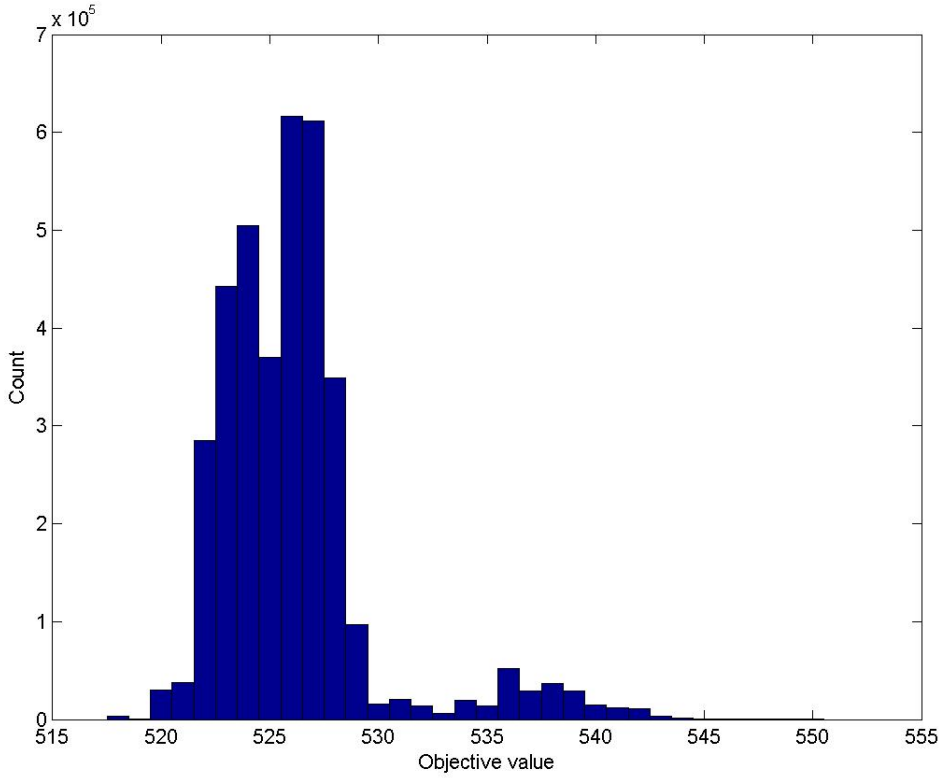

Figure 3.3.5: "Heart Disease": Histogram of the objective values of all one-round strategies for the greedy algorithm

formula (1.3.14)-(1.3.16)). The choices of input parameter for the alternative objectives are done by testing all possible one-round strategies, too. The maximizing-minimum-group-accuracy objective improves the group accuracies a little bit, as seen in Table 3.3.23.

Besides using all features in the computational experiment, we also run different classification methods using all possible subsets of the features in which the sizes of the subsets are greater than or equal to two. Figure 3.3.6 and 3.3.7 demonstrate the overall accuracy and minimum group accuracy, respectively, of the ten-fold cross-validation results. Each tick of the x-axis represents a unique feature subset, and from left to right the sizes of the subsets increase from two to eleven. The methods include

Table 3.3.22: "Heart Disease": Classification results by DALP

Ten-fold cross-validation

	0	1	2	3	4	0	1	2	3	4
0	337	45	10	14	5	0.8200	0.1095	0.0243	0.0341	0.0122
1	49	60	45	33	9	0.2500	0.3061	0.2296	0.1684	0.0459
2	18	44	31	30	12	0.1333	0.3259	0.2296	0.2222	0.0889
3	11	26	27	49	22	0.0815	0.1926	0.2000	0.3630	0.1630
4	5	7	8	10	13	0.1163	0.1628	0.1860	0.2326	0.3023
Overall accuracy: 0.5326										

Table 3.3.23: "Heart Disease": Classification results by DAMIP solved by greedy algorithm using the maximizing-minimum-group-accuracy objective

Ten-fold cross-validation

	0	1	2	3	4	0	1	2	3	4
0	326	51	19	12	3	0.7932	0.1241	0.0462	0.0292	0.0073
1	46	59	54	23	14	0.2347	0.3010	0.2755	0.1173	0.0714
2	13	40	41	24	17	0.0963	0.2963	0.3037	0.1778	0.1259
3	10	22	44	38	21	0.0741	0.1630	0.3259	0.2815	0.1556
4	4	7	9	8	15	0.0930	0.1628	0.2093	0.1860	0.3488
Overall accuracy: 0.5207										

Bayes, DALP ($(c_1, c_2) = (1, 0)$, solved by CPLEX using one thread), DAMIP (solved by greedy algorithm), and DAMIP with maximizing-minimum-group-accuracy objective (solved by greedy algorithm). The computational times are 155, 4346, 656, and 657 seconds, respectively. We see that DAMIP generally gives the best overall accuracy but it can have low minimum group accuracy; DAMIP with maximizing-minimum-group-accuracy objective overcomes the drawback and still gives good overall accuracies.

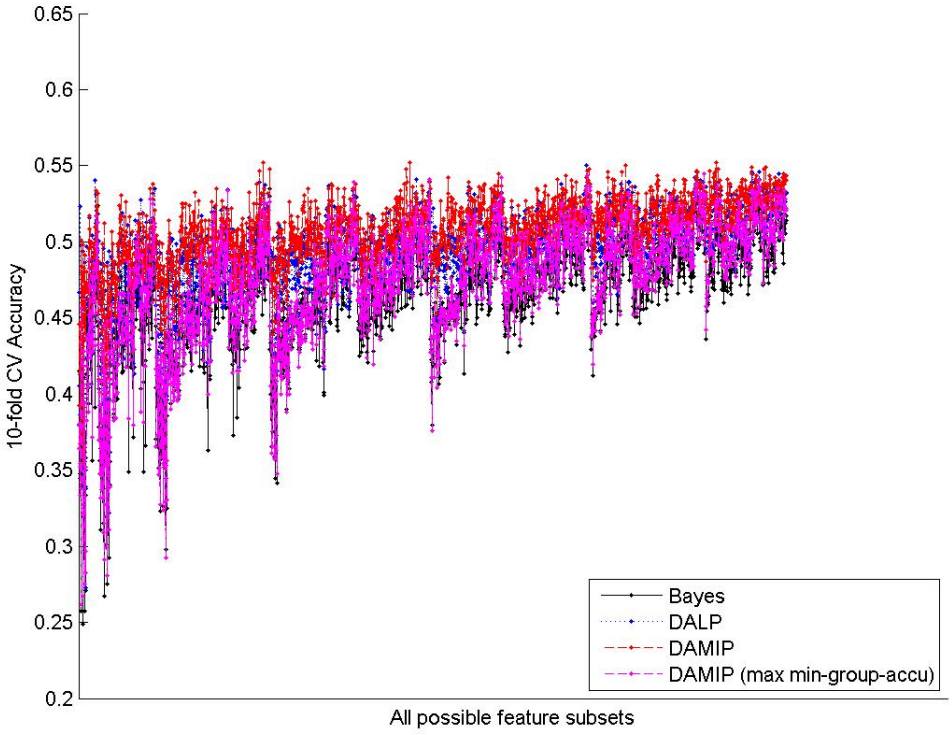

Figure 3.3.6: "Heart Disease": 10-fold CV accuracy on all possible subsets of the features

3.3.3.4 Data Set "Nursery"

The original "Nursery" data set consists of five groups, representing the rank of applications for nursery schools. The groups are not recommended, recommend, very

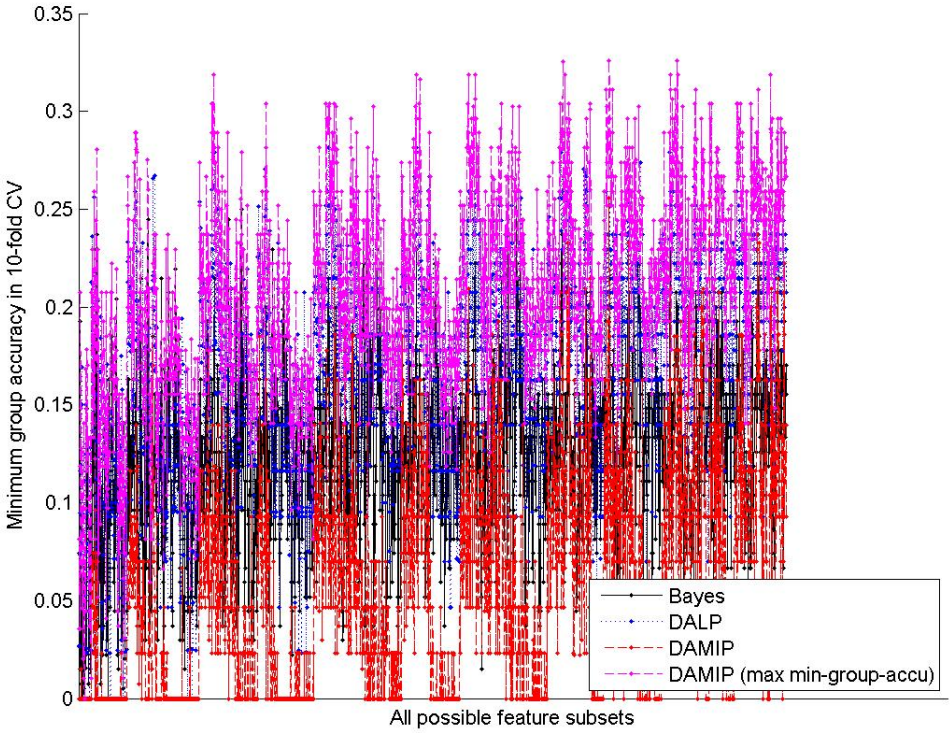

Figure 3.3.7: "Heart Disease": Minimum group accuracy (10-fold CV) on all possible subsets of the features

recommended, priority, and special priority; the numbers in each group are 4320, 2, 328, 4266, and 4044. Since the "recommend" group is too small, we combine it with the "very recom" group, resulting in four groups. The total number of observations is 12960 and the number of features is 8. Table 3.3.24 shows the Mahalanobis distances between groups.

We run all $\binom{4}{2}! = 720$ one-round strategies for input parameter of the greedy algorithm, which is done in 12 seconds. The objective values of all one-round strategies are either 11694 or 11696, as seen in the histogram in Figure 3.3.8. The histogram shows that if we only run part of the permutations, we will get the best or close to the best objective value of all one-round strategies. Among the best results we choose

Table 3.3.24: "Nursery": Mahalanobis distances between groups

	not recom	very recom	priority	spec prior
not recom	–	5.89	4.27	3.54
very recom	–	–	1.89	3.63
priority	–	–	–	1.95
spec prior	–	–	–	–

34-23-24-25-12-13-14-15-35-45 for further use on this data set.

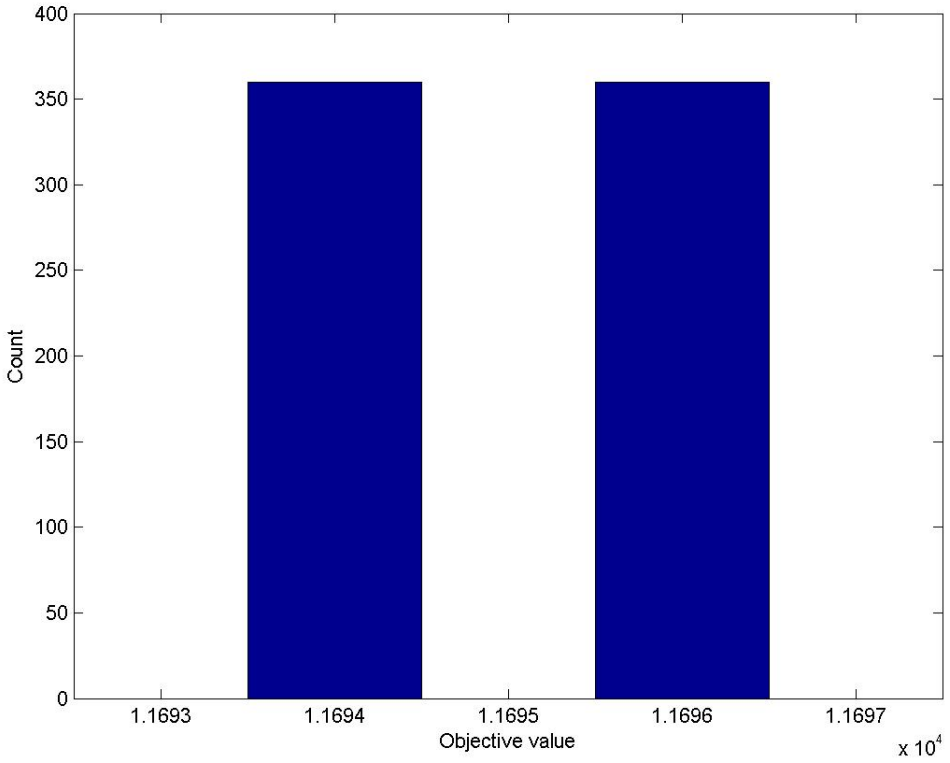

Figure 3.3.8: "Ecoli": Histogram of the objective values of all one-round strategies for the greedy algorithm

The greedy algorithm gives the objective value 11696 (or 0.9025 as the accuracy) in one second. Originally we run CPLEX using default settings with maximum eight threads, but it cannot get any feasible solution in 24 hours. We change the CPLEX

Table 3.3.25: "Nursery": Classification results by DAMIP solved by greedy algorithm

Ten-fold cross-validation

	not recom	very recom	priority	spec prior	not recom	very recom	priority	spec prior
not recom	4320	0	0	0	1.0000	0.0000	0.0000	0.0000
very recom	0	145	185	0	0.0000	0.4394	0.5606	0.0000
priority	0	69	3544	653	0.0000	0.0162	0.8308	0.1531
spec prior	0	0	596	3448	0.0000	0.0000	0.1474	0.8526
Overall accuracy: 0.8840								

setting to emphasize on MIP feasibility instead of balancing feasibility and optimality, and by doing so we get the objective value 11550 (or 0.8912 as the accuracy) in 24 hours with a 11.44% gap.

The ten-fold cross-validation classification results by DAMIP (solved by greedy algorithm), DAMIP using the maximizing-minimum-group-accuracy objective (solved by greedy algorithm), Bayes, and DALP ($(c_1, c_2) = (1, 0)$, solved by CPLEX) are shown in Table 3.3.25 to 3.3.28. The choice of input parameter for DAMIP with the alternative objective is done by testing all possible one-round strategies, too. In summary, the overall accuracies are 0.8840, 0.8576, 0.8415, and 0.8928, respectively. The solution times are 7, 6, 1, and 124 seconds. CPLEX are run in default settings with maximum one thread.

Besides using all features in the computational experiment, we also run different classification methods using all possible subsets of the features in which the sizes of the subsets are greater than or equal to two. Figure 3.3.9 and 3.3.10 demonstrate the overall accuracy and minimum group accuracy, respectively, of the ten-fold cross-validation results. Each tick of the x-axis represents a unique feature subset, and from left to right the sizes of the subsets increase from two to eight. The methods include

Table 3.3.26: "Nursery": Classification results by DAMIP solved by greedy algorithm using the maximizing-minimum-group-accuracy objective

Ten-fold cross-validation

	not recom	very recom	priority	spec prior	not recom	very recom	priority	spec prior
not recom	4320	0	0	0	1.0000	0.0000	0.0000	0.0000
very recom	0	309	21	0	0.0000	0.9364	0.0636	0.0000
priority	0	652	3061	553	0.0000	0.1528	0.7175	0.1296
spec prior	0	0	619	3425	0.0000	0.0000	0.1531	0.8469
Overall accuracy: 0.8576								

Table 3.3.27: "Nursery": Classification results by Bayes

Ten-fold cross-validation

	not recom	very recom	priority	spec prior	not recom	very recom	priority	spec prior
not recom	4320	0	0	0	1.0000	0.0000	0.0000	0.0000
very recom	0	314	16	0	0.0000	0.9515	0.0485	0.0000
priority	0	895	2806	565	0.0000	0.2098	0.6578	0.1324
spec prior	0	5	573	3466	0.0000	0.0012	0.1417	0.8571
Overall accuracy: 0.8415								

Table 3.3.28: "Nursery": Classification results by DALP

Ten-fold cross-validation

	not recom	very recom	priority	spec prior	not recom	very recom	priority	spec prior
not recom	4320	0	0	0	1.0000	0.0000	0.0000	0.0000
very recom	0	231	99	0	0.0000	0.7000	0.3000	0.0000
priority	0	225	3663	378	0.0000	0.0527	0.8586	0.0886
spec prior	2	0	685	3357	0.0005	0.0000	0.1694	0.8301
Overall accuracy: 0.8928								

Bayes, DALP ($(c_1, c_2) = (1, 0)$, solved by CPLEX using one thread), DAMIP (solved by greedy algorithm), and DAMIP with maximizing-minimum-group-accuracy objective (solved by greedy algorithm). The computational times are 184, 68800, 637, and 630 seconds, respectively. We see that DAMIP generally gives the best overall accuracy but it can have low minimum group accuracy; DAMIP with maximizing-minimum-group-accuracy objective overcomes the drawback and still gives good overall accuracies.

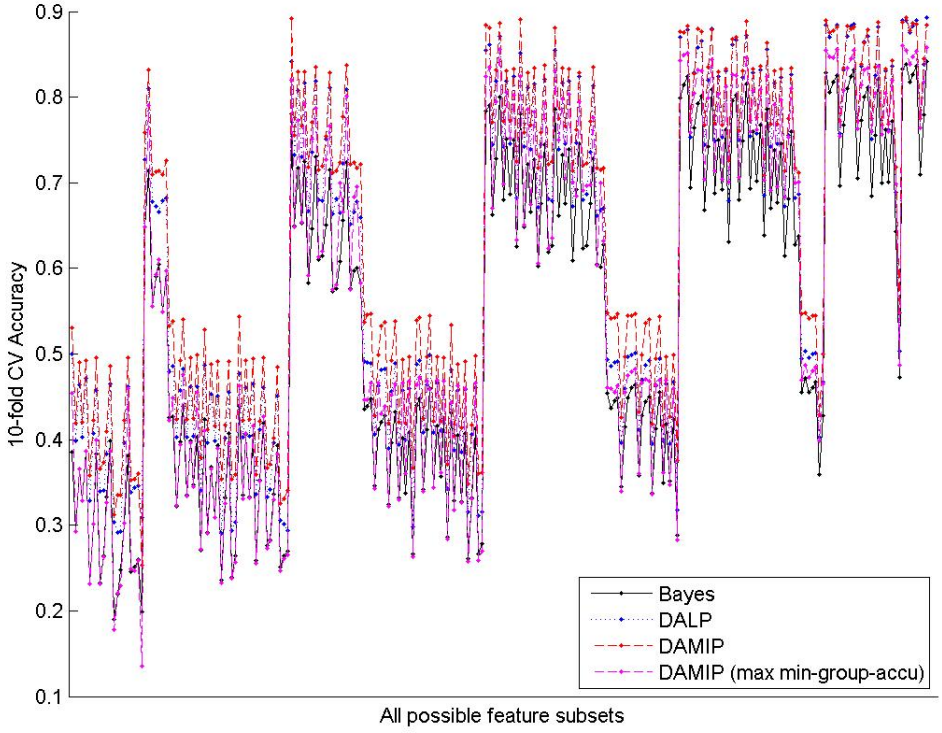

Figure 3.3.9: "Nursery": 10-fold CV accuracy on all possible subsets of the features

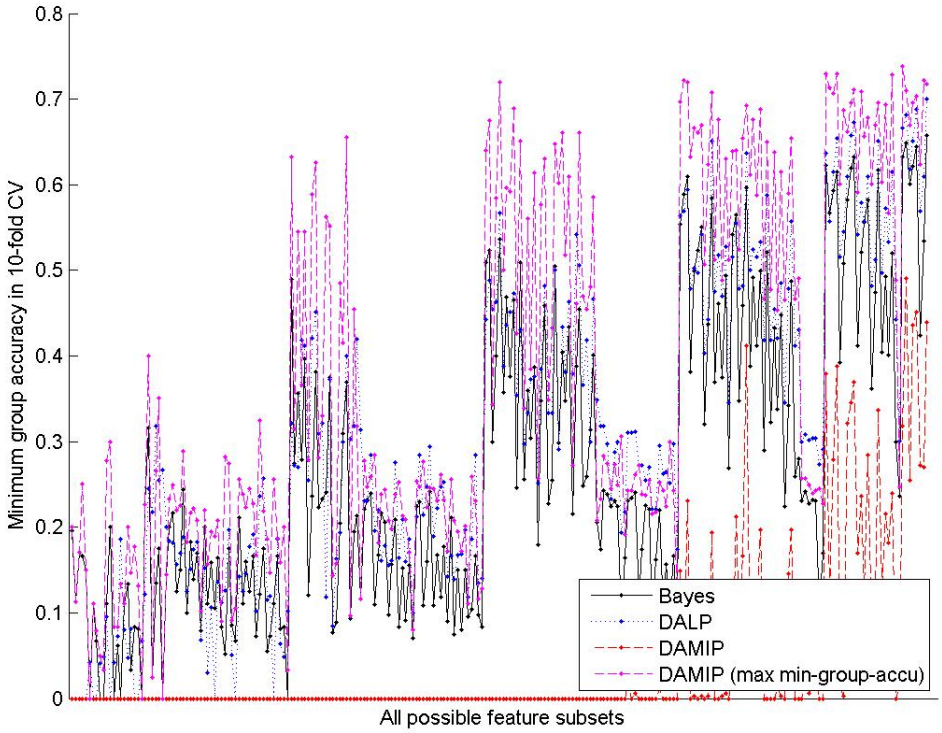

Figure 3.3.10: "Nursery": Minimum group accuracy (10-fold CV) on all possible subsets of the features

3.4 Trials on Solving DAMIP with Cuts

3.4.1 Combinatorial Benders' Cuts

The formulation of DAMIP possesses the property that, in the inequalities which contain both continuous and binary variables, exactly one binary variable appears. Furthermore, the objective function contains only the binary variables but no continuous ones. These properties are suitable for the application of the Combinatorial Benders' (CB) cuts [24].

Let x be the vector of integer variables, y be the vector of continuous variables, and B and G be index sets of general-integer and binary variables. Consider the following mixed integer program:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Fx \leq g \\ & Mx + Ay \geq b \\ & Dy \geq e, \\ & x_j \in \{0,1\} \quad \text{for } j \in B \\ & x_j \text{ integer} \quad \text{for } j \in G \end{aligned}$$

where M is a matrix with exactly one nonzero element in each row i, say column $j(i)$, and the corresponding variable $x_{j(i)}$ is a binary variable. The problem can be decomposed into the master problem and the slave problem:

MASTER:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Fx \leq g \\ & x_j \in \{0,1\} \quad \text{for } j \in B \\ & x_j \text{ integer} \quad \text{for } j \in G \end{aligned}$$

SLAVE($\tilde{x}$):

$$\begin{aligned} & Ay \geq b - M\tilde{x} \\ & Dy \geq e \end{aligned}$$

Let x^* be an optimal solution of the master problem. If the linear system SLAVE(x^*) has a solution y^*, then (x^*, y^*) is optimal to the original problem; otherwise we find a minimal infeasible subsystem of the slave problem (let the corresponding rows of A be indexed by C), add the cut (3.4.1) to the master problem, and solve the master again.

$$\sum_{i \in C:\ x^*_{j(i)}=0} x_j + \sum_{i \in C:\ x^*_{j(i)=1}} (1 - x_j) \geq 1 \tag{3.4.1}$$

The cut (3.4.1) is the Combinatorial Benders' cut, which requires to change the value of at least one $x^*_{j(i)}$ where $i \in C$. We look for the set of indices C by the fact that the indices of the minimal infeasible subsystems are exactly the supports of the extreme rays of the dual polyhedron of the slave problem [48, 116].

In the application of the CB cut to the DAMIP problem, the big-M values will not be used, which is an advantage of this method. The big-M values of DAMIP appear in the M matrix in the slave problem. But when any row of $Ay \geq b - Mx^*$ still has the big-M term after the value of x^* is put in, that row is in fact a redundant inequality, which can be taken away. We simply assign the dual variable of that row to zero in the slave problem.

We implement the CB cuts in CPLEX. However, our computational experience shows that our implementation does not reduce the solution time; default solving by CPLEX is faster.

3.4.2 Projected Chvatal-Gomory Cuts

The property of DAMIP that the objective function contains only the binary variables is suitable for the application of the projected Chvatal-Gomory (pro-CG) cuts [14]. It is shown that Combinatorial Benders' cuts are pro-CG cuts but pro-CG cuts can be much stronger than CB cuts [14].

Consider the mixed integer linear program

$$\begin{aligned} \min \quad & c^T x + f^T y \\ \text{s.t.} \quad & Ax + Cy \leq b \\ & x \geq 0,\ x \text{ integer} \\ & y \geq 0 \end{aligned}$$

where A has size $m \times n$ and C has size $m \times r$. Consider the related polyhedron

$$P(x, y) \equiv \{(x, y) \in \mathbb{R}^n_+ \times \mathbb{R}^r_+ : Ax + Cy \leq b\}.$$

Let $P(x)$ be the projection of $P(x, y)$ onto the space of the integer variables x. The projected Chvatal-Gomory cut is defined as a Chvatal-Gomory cut derived from the system which describes $P(x)$. Equivalently, a pro-CG cut is an inequality of the form

$$\lfloor u^T A \rfloor x \leq \lfloor u^T b \rfloor \quad \text{for any } u \geq 0 \text{ such that } u^T C \geq 0^T$$

To obtain a pro-CG cut $\alpha^T x \leq \alpha_0$, the separation problem can be modeled by the mixed integer linear program:

$$\begin{aligned}
\max \quad & \alpha^T x^* - \alpha_0 & \\
\text{s.t.} \quad & \alpha_j \leq u^T A_j & \text{for } j = 1, \ldots, n \\
& 0 \leq u^T C_j & \text{for } j = 1, \ldots, r \\
& \alpha_0 + 1 - \epsilon \geq u^T b & \\
& u_i \geq 0 & \text{for } i = 1, \ldots, m \\
& \alpha_j \text{ integer} & \text{for } j = 0, \ldots, n
\end{aligned}$$

where A_j and C_j are the columns of A and C, and ϵ is a small positive number.

We implement the pro-CG cuts in CPLEX to solve DAMIP. However, we do not gain in applying pro-CG cuts; default solving by CPLEX is faster.

CHAPTER IV

APPLICATIONS

This chapter applies the PSO/DAMIP classification framework to several real-world medical and biological prediction problems, including Alzheimer's disease, cardiovascular disease, sulfur amino acid intake, and CpG islands.

4.1 Alzheimer's Disease

4.1.1 Background

This section describes the background of this study, including Alzheimer's disease and mild cognitive impairment, neuropsychological tests, predictive analysis using neuropsychological data, data from Emory, and data from LONI/ADNI.

4.1.1.1 Alzheimer's Disease and Mild Cognitive Impairment

Alzheimer's disease (AD), the 7th leading cause of death in the United States, is a progressive and irreversible brain disease which causes memory loss and other cognitive problems severe enough to affect daily life. Dementia is a collection of symptoms of cognitive function problems, such as thinking, remembering, or reasoning problems, and AD is the most common cause of dementia. Mostly AD occurs in people over 65, although familial AD has an earlier onset. Currently, AD is incurable; drugs are used to manage the symptoms or to prevent or slow the progress of the disease.

Mild cognitive impairment (MCI) is a condition that there is clear evidence of cognitive problems, most often involving short term memory, but normal day to day

functioning is preserved. In other words, MCI is a situation between normal aging and dementia. People with MCI may or may not develop dementia in the future, but people with MCI are at higher risk of developing dementia than those without MCI.

The evaluation of AD or MCI depends on some clinical and patient data, including complete medical history, neurological exam, laboratory tests, neuropsychological tests, brain scans (CT or MRI), and information from close family members.

4.1.1.2 Neuropsychological Tests

Neuropsychological changes in the expression of cognitive declines are important to the diagnosis of AD and MCI. Bondi et al. [15] review neuropsychological changes during the prodromal period of Alzheimer's disease, which are important to the early identification of the disease. Nelson and OConnor [113] review mild cognitive impairment from the neuropsychological perspective, including the MCI diagnostic criteria, MCI subtypes, and neuropsychological tests, for the purpose of early identification of Alzheimer's disease. The neuropsychological tests which follow certain criteria are good instruments for evaluating neuropsychological status.

4.1.1.3 Predictive Analysis Using Neuropsychological Data

Statistical analyses as predictive analysis tools are applied to neuropsychological data to understand MCI patents. Lopez et al. [92] analyze neuropsychological characteristics of normal subjects, MCI-amnestic type (MCI-AT) subjects, and MCI-multiple cognitive deficits type (MCI-MCDT) subjects. Tabert et al. [153] conduct hypothesis testing to compare (1) MCI patients with controls, and (2) MCI patients who converted to AD with MCI patients who did not, in a follow-up duration.

Besides statistical analyses, some classification models are also applied to neuropsychological data for predictive analysis. Stuss and Trites [150] apply discriminant function analysis to discriminate the control group, the brain-damaged group with a positive physical neurological exam, and the brain-damaged group with a negative result of the same exam. Kluger et al. [72] apply logistic regression and stepwise entry procedure to predict (1) whether nondemented elderly subsequently declined to any diagnosis of dementia; and (2) whether nondemented elderly subsequently declined to a diagnosis of probable Alzheimer's disease. Possible predictor variables include demographic variables, Global Deterioration Scale (GDS) score, and nine cognitive test scores from the neuropsychological battery of NYU Aging and Dementia Research Center.

4.1.1.4 Data from Emory

We apply our methods to classify subjects from three groups: Subjects of Alzheimer's disease, subjects of mild cognitive impairment, and the control group, using neuropsychological test data.

Data of neuropsychological tests from 35 subjects were collected in Emory Alzheimer's Disease Research Center from 2004 to 2007. Eighteen kinds of neuropsychological tests were applied to the subjects, but only four of them were applied to all subjects, thus being used in our predictive model. These tests included

1. Mini Mental State Examination (MMSE),
2. Clock drawing test,
3. Word list memory tasks by the Consortium to Establish a Registry for Alzheimer's Disease (CERAD),

4. Geriatric depression scale (GDS).

The MMSE is a screening tool for cognitive impairment, which is brief, but covers five areas of cognitive function, including orientation, registration, attention and calculation, recall, and language. The clock drawing test assesses cognitive functions, particularly visuo-spatial abilities and executive control functions. The CERAD word list memory tasks assess learning ability for new verbal information. The tasks include word list memory with repetition, word list recall, and word list recognition. The GDS is a screening tool to assess the depression in older population.

Initially we have 153 features, including raw data from four neuropsychological tests as well as subjects age. Raw data from tests contain information of individual questions in the tests. Discarding features which contain missing values or which are undiscriminating (i.e., features which contain almost the same value among all subjects), we get 100 features for feature selection and classification. Besides, we also use only the nine score-type features (i.e., total or subtotal scores in different tests) instead of all raw data for feature selection and classification for comparison.

Our data comes from two trials. The number of subjects in two trials is listed in Table 4.1.1, in which Ctl represents the control group. Besides mixing subjects from both trials for analysis, we also train subjects of one trial and blind predict those of the other trial.

4.1.1.5 Data from LONI/ADNI

The Alzheimer's Disease Neuroimaging Initiative (ADNI) data website at Laboratory of Neuro Imaging (LONI), UCLA, includes repository of clinical and imaging

Table 4.1.1: Number of subjects of three groups from two trials in data from Emory

	AD	MCI	Ctl	Total
Trial 1	5	3	2	10
Trial 2	2	13	10	25
Total	7	16	12	35

data. Clinical data include data of several neuropsychological tests, which are used for classification in this study. The neuropsychological tests include clock drawing test, category fluency test, Boston naming test, and so on. The category fluency test requires the systematic retrieval of hierarchically organized information from semantic memory; the Boston naming test measures the ability to name objects of line drawings.

The data set contain results of neuropsychological tests taken by subjects at several time points; we use the data taken at the baseline time point, i.e., the first time a subject took the tests. Data include 819 subjects and 59 features. The features are score-type ones rather than raw data of the tests. After we handle missing values by discarding some data, 786 subjects and 54 features are left for feature selection and classification. The numbers of AD, MCI, and the control group are 223, 388, and 175, respectively.

4.1.2 Results

Besides training one trial and blind predicting the other trial in Emory data, in all other cases we randomly select 67% of the subjects in each group for training using 10-fold cross-validation and use the remaining subjects for blind prediction. We apply the PSO/DAMIP classification framework to discriminate subjects from AD, MCI, and control groups. The best classification results as well as the selected features in

Table 4.1.2: Classification results of Emory data. Five selected features: MMSE–cMMtotal, WordList–cWL2Butter, WordList–cWL2Queen, WordList–cWL2Ticket, GDS–GDS13.

Ten-fold cross-validation

	AD	MCI	Ctl	AD	MCI	Ctl
AD	4	1	0	0.80	0.20	0.00
MCI	0	11	0	0.00	1.00	0.00
Ctl	0	0	8	0.00	0.00	1.00
Overall accuracy: 0.96						

Blind prediction

	AD	MCI	Ctl	AD	MCI	Ctl
AD	2	0	0	1.00	0.00	0.00
MCI	1	4	0	0.20	0.80	0.00
Ctl	0	0	4	0.00	0.00	1.00
Overall accuracy: 0.91						

each case are demonstrated in the following tables.

Classification results of Emory data are shown from Table 4.1.2 to Table 4.1.9. Table 4.1.2, 4.1.3, 4.1.4, and 4.1.5 show the results of 10-fold cross-validation and blind prediction from 100 features; Table 4.1.6 and 4.1.7 show the results of training trial 1 and blind predicting trial 2 from 100 features; Table 4.1.8 shows the results of training trial 2 and blind predicting trial 1 from 100 features; Table 4.1.9 shows the results of 10-fold cross-validation and blind prediction from 9 score-type features.

Classification results of LONI/ADNI data are shown in Table 4.1.10 and 4.1.11, which are results of 10-fold cross-validation and blind prediction from 54 features.

Using the PSO/DAMIP classification framework, we successfully discriminated subjects from AD, MCI, and control groups with 80% accuracy in both training and

Table 4.1.3: Classification results of Emory data. Four selected features: MMSE–cMMtotal, MMSE–cMMz, WordList–cWL1Queen, GDS–GDS13; or five selected features: MMSE–cMMtotal, MMSE–cMMz, WordList–cWL2Butter, WordList–cWL1Queen, GDS–GDS13.

Ten-fold cross-validation

	AD	MCI	Ctl	AD	MCI	Ctl
AD	4	1	0	0.80	0.20	0.00
MCI	0	10	1	0.00	0.91	0.09
Ctl	0	0	8	0.00	0.00	1.00
Overall accuracy: 0.92						

Blind prediction

	AD	MCI	Ctl	AD	MCI	Ctl
AD	2	0	0	1.00	0.00	0.00
MCI	0	5	0	0.00	1.00	0.00
Ctl	0	1	3	0.00	0.25	0.75
Overall accuracy: 0.91						

Table 4.1.4: Classification results of Emory data. Five selected features: MMSE–cMMsRapple, WordList–cWL1Queen, WordList–cWL3Engine, GDS–GDS9, GDS–GDS13.

Ten-fold cross-validation

	AD	MCI	Ctl	AD	MCI	Ctl
AD	5	0	0	1.00	0.00	0.00
MCI	0	10	1	0.00	0.91	0.09
Ctl	0	1	7	0.00	0.13	0.88
Overall accuracy: 0.92						

Blind prediction

	AD	MCI	Ctl	AD	MCI	Ctl
AD	2	0	0	1.00	0.00	0.00
MCI	1	4	0	0.20	0.80	0.00
Ctl	0	0	4	0.00	0.00	1.00
Overall accuracy: 0.91						

Table 4.1.5: Classification results of Emory data. Five selected features: MMSE–cMMtotal, WordList–cWL3Queen, WordList–cWL2Engine, GDS–GDS13, GDS–GDS15.

Ten-fold cross-validation

	AD	MCI	Ctl	AD	MCI	Ctl
AD	3	2	0	0.60	0.40	0.00
MCI	0	11	0	0.00	1.00	0.00
Ctl	0	0	8	0.00	0.00	1.00
Overall accuracy: 0.92						

Blind prediction

	AD	MCI	Ctl	AD	MCI	Ctl
AD	1	1	0	0.50	0.50	0.00
MCI	0	5	0	0.00	1.00	0.00
Ctl	0	0	4	0.00	0.00	1.00
Overall accuracy: 0.91						

Table 4.1.6: Classification results of Emory data, training trial 1 and blind predicting trial 2 from 100 features. Five selected features: MMSE–cMMsCounty, MMSE–cMMsWorld, Clock–cClockHands1, WordList–cWL2Queen, WordList–cWRyShore.

Training

	AD	MCI	Ctl	AD	MCI	Ctl
AD	5	0	0	1.00	0.00	0.00
MCI	0	3	0	0.00	1.00	0.00
Ctl	0	0	2	0.00	0.00	1.00
Overall accuracy: 1.00						

Blind prediction

	AD	MCI	Ctl	AD	MCI	Ctl
AD	2	0	0	1.00	0.00	0.00
MCI	0	9	4	0.00	0.69	0.31
Ctl	0	1	9	0.00	0.10	0.90
Overall accuracy: 0.80						

Table 4.1.7: Classification results of Emory data, training trial 1 and blind predicting trial 2 from 100 features. Five selected features: MMSE–cMMsCounty, MMSE–cMMsWorld, Clock–cClockHands1, WordList–cWL2Queen, WordList–cWRyArm.

Training

	AD	MCI	Ctl	AD	MCI	Ctl
AD	5	0	0	1.00	0.00	0.00
MCI	0	3	0	0.00	1.00	0.00
Ctl	0	0	2	0.00	0.00	1.00
Overall accuracy: 1.00						

Blind prediction

	AD	MCI	Ctl	AD	MCI	Ctl
AD	2	0	0	1.00	0.00	0.00
MCI	2	9	2	0.15	0.69	0.15
Ctl	0	1	9	0.00	0.10	0.90
Overall accuracy: 0.80						

Table 4.1.8: Classification results of Emory data, training trial 2 and blind predicting trial 1 from 100 features. Five selected features: Age, MMSE–cMMsRapple, WordList–cWL2Queen, WordList–cWL2Engine, GDS–GDS13.

Training

	AD	MCI	Ctl	AD	MCI	Ctl
AD	2	0	0	1.00	0.00	0.00
MCI	1	12	0	0.08	0.92	0.00
Ctl	0	0	10	0.00	0.00	1.00
Overall accuracy: 0.96						

Blind prediction

	AD	MCI	Ctl	AD	MCI	Ctl
AD	4	1	0	0.80	0.20	0.00
MCI	0	3	0	0.00	1.00	0.00
Ctl	0	0	2	0.00	0.00	1.00
Overall accuracy: 0.90						

Table 4.1.9: Classification results of Emory data from 9 score-type features. Two selected features: MMSE–cMMtotal, Word List–cWLcorTotal.

Ten-fold cross-validation

	AD	MCI	Ctl	AD	MCI	Ctl
AD	4	1	0	0.80	0.20	0.00
MCI	1	9	1	0.09	0.82	0.09
Ctl	0	2	6	0.00	0.25	0.75
Overall accuracy: 0.79						

Blind prediction

	AD	MCI	Ctl	AD	MCI	Ctl
AD	1	1	0	0.50	0.50	0.00
MCI	0	5	0	0.00	1.00	0.00
Ctl	0	1	3	0.00	0.25	0.75
Overall accuracy: 0.82						

Table 4.1.10: Classification results of LONI/ADNI data. Five selected features: CLOCKHAND, AVTOT5, AVTOT6, CATVEGESC, TRABERROM.

Ten-fold cross-validation

	AD	MCI	Ctl	AD	MCI	Ctl
AD	114	35	0	0.77	0.23	0.00
MCI	38	175	47	0.15	0.67	0.18
Ctl	3	42	72	0.03	0.36	0.62
Overall accuracy: 0.69						

Blind prediction

	AD	MCI	Ctl	AD	MCI	Ctl
AD	56	17	1	0.76	0.23	0.01
MCI	21	85	22	0.16	0.66	0.17
Ctl	0	22	36	0.00	0.38	0.62
Overall accuracy: 0.68						

Table 4.1.11: Classification results of LONI/ADNI data. Five selected features: AVTOT5, AVTOT6, CATVEGESC, TRABSCOR, TRABERROM.

Ten-fold cross-validation

	AD	MCI	Ctl	AD	MCI	Ctl
AD	113	35	1	0.76	0.23	0.01
MCI	36	173	51	0.14	0.67	0.20
Ctl	1	43	73	0.01	0.37	0.62
Overall accuracy: 0.68						

Blind prediction

	AD	MCI	Ctl	AD	MCI	Ctl
AD	57	17	0	0.77	0.23	0.00
MCI	20	85	23	0.16	0.66	0.18
Ctl	0	23	35	0.00	0.40	0.60
Overall accuracy: 0.68						

blind prediction for raw data from Emory. We conclude that raw data of neuropsychological tests have potential to predict subjects from AD, MCI, and control groups.

4.2 Cardiovascular Disease

Cardiovascular disease has been the top one leading cause of death in the United States for many years, and atherosclerosis is a main cause of cardiovascular disease. Early detection of atherosclerosis is very important.

4.2.1 Background

This section describes the background of this study, including subjects, measurement of the biomarkers, carotid intima-media thickness (IMT), brachial artery flow-mediated vasodilation (FMD), groups for classification, and features.

4.2.1.1 Subjects

A total of 124 healthy nonsmoking volunteers between the ages of 30 and 65 years without any known cardiovascular risk factors, such as hypertension, diabetes, or hypercholesterolemia, and without clinically evident atherosclerosis were recruited by advertisement. Subjects were excluded if they were known to had a history of diabetes (fasting glucose of $>$ 126 mg/dL or hemoglobin A1c of $>$ 7%), hypertension (elevated systolic [$>$ 140 mm Hg] or diastolic blood pressure [$>$ 90 mm Hg] on 3 separate measurements), or hyperlipidemia requiring treatment; were smoking in last 3 months; or were on any vasoactive medications, vitamins, or supplements. Pregnant women and those with acute or chronic illnesses were also excluded. The study was approved by the Emory University Institutional Review Committee. Informed consent was obtained from all of the subjects.

After answering a questionnaire and a routine physical examination, overnight fasting blood samples were obtained. Plasma levels of total, low-density lipoprotein (LDL), and high-density lipoprotein (HDL) cholesterol; triglycerides; and glucose

were measured. Highsensitivity (hs) C-reactive protein (CRP) was measured by immunonephelometry (Dade Behring).

4.2.1.2 Measurement of Thiol and Disulfide Forms of Glutathione and Cysteine, Their Redox States, and the CySSG

Detailed procedures for measurements of blood GSH, GSSG, Cys, CySS, CySSG, Eh GSH/GSSG, and Eh Cys/CySS have been described previously (15,1921). Samples were collected directly into specially prepared tubes containing a preservative to reduce autooxidation, centrifuged, and the supernatant frozen at -80C, which shows no significant loss for = 1 year. Analyses by highperformance liquid chromatography were performed after dansyl derivatization on a 3-aminopropyl column with fluorescence detection. 19 Metabolites were identified by coelution with standards and quantified by integration relative to the internal standard, with validation relative to external standards. Issues of sample collection, stability, analysis, and standardization have been extensively studied, and the method has been used in several clinical studies (22). The coefficient of variation for GSH was 5%, and the coefficient of variation for GSSG was 9.7%. The SD for week-to-week variation among individuals for GSH redox potential was 3.22 mV. The reproducibility values were similar for Cys, CySS, CySSG, and Eh Cys/CySS.

4.2.1.3 Measurement of Carotid IMT

IMT was measured using ultrasonography and standard techniques (25,26). Longitudinal images of the distal 1.0 cm of both common carotid arteries, proximal to the carotid bulb were obtained using multiple scanning angles. The images were stored digitally, and measurements were made off-line using a semi-automated computerized analytical software (Carotid Tools, MIA Inc., Iowa City, Iowa), by two observers

blinded to the test results. Average values of the IMT of each of the four segments of the distal 1.0 cm of both common carotid arteries (right near and far walls, and left near and far walls) were used as the IMT values for each subject. Inter-observer variability for carotid IMT was 0.03 ± 0.02 mm between measurements made in 20 subjects by 2 observers. Intra-observer variability was 0.02 ± 0.02 mm between 2 measurements made 1 week apart on 10 subjects.

4.2.1.4 Measurement of Brachial Artery FMD

Endothelium-dependent brachial artery FMD was determined as described previously after the blood sample for biomarker evaluation was obtained (2,23). Briefly, ultrasound images were obtained at baseline under standardized conditions and 60 seconds after induction of reactive hyperemia by 5-minute cuff occlusion of the forearm. After a 15-minute period to re-establish baseline conditions, endothelium-independent dilation of the brachial artery was assessed from images obtained before and 3 to 5 minutes after administration of 0.4 mg of sublingual nitroglycerin. Images were digitized online, and arterial diameters were measured with customized software (Medical Imaging Applications, Inc) by individuals blinded to the clinical status and laboratory status of the subjects. FMD and endothelium-independent vasodilation were expressed as the percentage increase in diameter from baseline. In our laboratory, the mean difference in FMD between 2 consecutive assessments performed in 11 subjects an average of 8 days apart was 1.26±0.76%, with a correlation coefficient of 0.75. The mean difference in the FMD between 2 readings of the same 11 measurements was 0.82±0.48% (r=0.97).

Table 4.2.1: Three grouping ways for classification

	Relatively High Risk		Relatively Low Risk	
	Criteria	# Subjects	Criteria	# Subjects
Grouping by IMT	IMT≥0.68	29	IMT<0.68	92
Grouping by FMD	FMD≤8.25	75	FMD>8.25	46
Grouping by IMD & FMD	IMT≥0.68 & FMD≤8.25	22	IMT<0.68 & FMD>8.25	39

4.2.1.5 Groups for Classification

Carotid IMT is a measure of early atherosclerosis; brachial artery FMD is a measure of vascular endothelial function, and endothelial dysfunction is known to precede the development of atherosclerosis. We use the values of IMT and/or FMD as the measure of the risk of atherosclerosis.

We group the subjects for classification in three ways, shown in Table 4.2.1. In the first way subjects of high and low risks are separated by 0.68 mm of IMT. In the second way subjects of high and low risks are separated by 8.25% of FMD, in which the cut point 8.25% is obtained by the k-means clustering method. In the third way we put the first two criteria together, resulting in half of the total number of subjects remained for analysis.

4.2.1.6 Features

We have 25 candidate features, including (1) 11 traditional risk factors: age, gender, body mass index (BMI), triglyceride (TG), LDL, HDL, total cholesterol (TC), diabetes mellitus, hypertension, prior smoking history, family history of coronary artery disease (CAD), (2) Framingham risk score, (3) inflammatory marker: hs-CRP, (4) 7

Table 4.2.2: Classification results on IMT, selecting five features: Age, BMI, Family CAD history, Eh GSH/GSSG, d-ROM

Ten-fold cross-validation

	Large IMT	Small IMT	Large IMT	Small IMT
Large IMT	15	5	0.7500	0.2500
Small IMT	4	57	0.0656	0.9344
Overall accuracy: 0.8889				

Blind prediction

	Large IMT	Small IMT	Large IMT	Small IMT
Large IMT	7	2	0.7778	0.2222
Small IMT	1	30	0.0323	0.9677
Overall accuracy: 0.9250				

oxidative stress markers: GSH, GSSG, Eh GSH/GSSG, Cys, CySS, Eh Cys/CySS, CySSG, and (5) 5 other factors: myeloperoxidase, systolic blood pressure (BP), diastolic BP, fasting insulin, and d-ROM.

4.2.2 Results

This section shows the classification results for grouping by IMT, FMD, and both. We randomly chose two-third of the subjects for training and the remaining one-third for blind prediction. We conclude that our classification model is able to discriminate healthy subjects of relatively high and low risk of atherosclerosis.

4.2.2.1 Classification Results for Grouping by IMT

Table 4.2.2 and 4.2.3 show two of the best classification results for grouping by IMT.

We aggregate the number of times each feature appears in the feature sets which result in good classification results. For grouping by IMT, we select the results where classification accuracies are greater than or equal to 75% in both groups and in both

Table 4.2.3: Classification results on IMT, selecting six features: Age, Gender, Hypertension, Family CAD history, Eh GSH/GSSG, d-ROM

Ten-fold cross-validation

	Large IMT	Small IMT	Large IMT	Small IMT
Large IMT	15	5	0.7500	0.2500
Small IMT	6	55	0.0984	0.9016
Overall accuracy: 0.8642				

Blind prediction

	Large IMT	Small IMT	Large IMT	Small IMT
Large IMT	7	2	0.7778	0.2222
Small IMT	2	29	0.0645	0.9355
Overall accuracy: 0.9000				

training and blind prediction. For all these results, we accumulate the number of times each feature appears in two ways: (1) non-weighted, i.e., each feature is counted once regardless of the size of the feature set it come from, and (2) weighted in reverse proportion of the size of the feature set, i.e., each count is weighted by the reciprocal of the size of the feature set. See Table 4.2.4 for the number of feature appearance. Note that due to rounding error, the total number of weighted counts could be non-integral or the sum of the percentages could be non-unity.

We also show the results where candidate features come from only the 11 traditional risk factors (age, gender, BMI, TG, LDL, HDL, TC, diabetes mellitus, hypertension, prior smoking history, family CAD history). Table 4.2.5 shows two feature sets with the same results.

If we use only the 7 oxidative stress markers (GSH, GSSG, Eh GSH/GSSG, Cys, CySS, Eh Cys/CySS, CySSG) as candidate features, none of the results have overall accuracies in training and blind prediction greater than or equal to 65%. Neither do the results for grouping by FMD and by IMT and FMD.

Table 4.2.4: Feature appearance for grouping by IMT of all 75% or better results

Non-weighted			Weighted		
Feature Name	Count	%	Feature Name	Count	%
Family CAD history	190	12.6	Family CAD history	25.09	12.7
d-ROM	187	12.5	d-ROM	24.83	12.5
Eh GSH/GSSG	153	10.2	Eh GSH/GSSG	20.12	10.2
Age	133	8.9	Age	17.44	8.8
BMI	91	6.1	BMI	12.09	6.1
Hypertension	72	4.8	Hypertension	9.46	4.8
HDL	59	3.9	Framingham risk score	7.97	4.0
Framingham risk score	59	3.9	HDL	7.82	3.9
Myeloperoxidase	57	3.8	Myeloperoxidase	7.42	3.7
Diabetes	55	3.7	Diabetes	7.26	3.7
Fasting insulin	55	3.7	Fasting insulin	7.24	3.7
Systolic BP	52	3.5	Systolic BP	6.84	3.5
hs-CRP	46	3.1	hs-CRP	6.04	3.0
TG (Triglyceride)	42	2.8	TG (Triglyceride)	5.43	2.7
Prior smoking	39	2.6	Prior smoking	5.06	2.6
Gender	38	2.5	Gender	4.99	2.5
LDL	33	2.2	LDL	4.27	2.2
TC (Cholesterol)	29	1.9	TC (Cholesterol)	3.73	1.9
Eh Cys/CySS	23	1.5	Eh Cys/CySS	3.07	1.6
Cys	22	1.5	Cys	2.84	1.4
CySSG	21	1.4	CySSG	2.68	1.4
GSSG	15	1.0	GSSG	1.95	1.0
GSH	13	0.9	Diastolic BP	1.81	0.9
Diastolic BP	12	0.8	GSH	1.68	0.8
CySS	6	0.4	CySS	0.89	0.4
Total	1502	100.2	Total	198.02	100.0

Table 4.2.5: Classification results on IMT, selecting (1) six features: Age, Gender, BMI, TC, Hypertension, Family CAD history, or (2) six features: Age, Gender, BMI, LDL, HDL, TC

Ten-fold cross-validation

	Large IMT	Small IMT	Large IMT	Small IMT
Large IMT	14	6	0.7000	0.3000
Small IMT	7	54	0.1148	0.8852
Overall accuracy: 0. 8395				

Blind prediction

	Large IMT	Small IMT	Large IMT	Small IMT
Large IMT	7	2	0.7778	0.2222
Small IMT	1	30	0.0323	0.9677
Overall accuracy: 0. 9250				

4.2.2.2 Classification Results for Grouping by FMD

Table 4.2.6 and 4.2.7 show two of the best classification results for grouping by FMD. Note that all features in this set are traditional risk factors.

Table 4.2.8 shows the feature appearance for grouping by FMD, where classification accuracies are greater than or equal to 70% in both groups and in both training and blind prediction.

4.2.2.3 Classification Results for Grouping by both IMT and FMD

Table 4.2.9 shows the best classification results for grouping by IMT and FMD. Note that all selected features are traditional risk factors.

Table 4.2.10 shows the feature appearance for grouping by IMT and FMD, where classification accuracies are greater than or equal to 80% in both groups and in both

Table 4.2.6: Classification results on FMD, selecting five features: Gender, BMI, LDL, Hypertension, Family CAD history

Ten-fold cross-validation

	Small FMD	Large FMD	Small FMD	Large FMD
Small FMD	40	13	0.7547	0.2453
Large FMD	8	20	0.2857	0.7143
Overall accuracy: 0.7407				

Blind prediction

	Small FMD	Large FMD	Small FMD	Large FMD
Small FMD	16	6	0.7273	0.2727
Large FMD	5	13	0.2778	0.7222
Overall accuracy: 0.7250				

Table 4.2.7: Classification results on FMD, selecting six features: Gender, Hypertension, GSSG, CySS, CySSG, Diastolic BP

Ten-fold cross-validation

	Small FMD	Large FMD	Small FMD	Large FMD
Small FMD	40	13	0.7547	0.2453
Large FMD	7	21	0.2500	0.7500
Overall accuracy: 0.7531				

Blind prediction

	Small FMD	Large FMD	Small FMD	Large FMD
Small FMD	17	5	0.7727	0.2273
Large FMD	5	13	0.2778	0.7222
Overall accuracy: 0.7500				

Table 4.2.8: Feature appearance for grouping by FMD of all 70% or better results

Non-weighted			Weighted		
Feature Name	Count	%	Feature Name	Count	%
Gender	231	13.2	Gender	30.82	13.3
CySSG	190	10.9	CySSG	25.10	10.9
Hypertension	165	9.4	Hypertension	22.12	9.6
Diastolic BP	136	7.8	Diastolic BP	17.96	7.8
GSSG	120	6.9	GSSG	15.90	6.9
d-ROM	97	5.5	BMI	12.66	5.5
GSH	95	5.4	d-ROM	12.48	5.4
BMI	94	5.4	GSH	12.46	5.4
LDL	74	4.2	LDL	9.75	4.2
Fasting insulin	60	3.4	Fasting insulin	7.83	3.4
Myeloperoxidase	53	3.0	Myeloperoxidase	6.84	3.0
hs-CRP	51	2.9	hs-CRP	6.71	2.9
TG (Triglyceride)	49	2.8	TG (Triglyceride)	6.34	2.7
CySS	48	2.7	CySS	6.30	2.7
Eh Cys/CySS	46	2.6	Eh Cys/CySS	6.03	2.6
TC (Cholesterol)	42	2.4	TC (Cholesterol)	5.47	2.4
Systolic BP	38	2.2	Systolic BP	4.89	2.1
Prior smoking	37	2.1	Prior smoking	4.77	2.1
Eh GSH/GSSG	36	2.1	Eh GSH/GSSG	4.65	2.0
Cys	30	1.7	Cys	3.93	1.7
Diabetes	28	1.6	Diabetes	3.74	1.6
Family CAD history	13	0.7	Family CAD history	2.00	0.9
HDL	12	0.7	HDL	1.60	0.7
Framingham risk score	5	0.3	Framingham risk score	0.64	0.3
Age	0	0.0	Age	0.00	0.0
Total	1750	99.9	Total	230.99	100.1

Table 4.2.9: Classification results on IMT and FMD, selecting (1) four features: Age, HDL, Hypertension, Family CAD history, (2) five features: Age, HDL, Hypertension, Family CAD history, Fasting insulin, or (3) six features: Age, HDL, Hypertension, Family CAD history, Framingham risk score, CySS

Ten-fold cross-validation

	Large IMT & Small FMD	Small IMT & Large FMD	Large IMT & Small FMD	Small IMT & Large FMD
Large IMT & Small FMD	14	3	0.8235	0.1765
Small IMT & Large FMD	1	24	0.0400	0.9600
Overall accuracy: 0.9048				

Blind prediction

	Large IMT & Small FMD	Small IMT & Large FMD	Large IMT & Small FMD	Small IMT & Large FMD
Large IMT & Small FMD	4	1	0.8000	0.2000
Small IMT & Large FMD	0	14	0.0000	1.0000
Overall accuracy: 0.9474				

training and blind prediction.

Table 4.2.10: Feature appearance for grouping by IMT and FMD of all 80% or better results

Non-weighted			Weighted		
Feature Name	Count	%	Feature Name	Count	%
d-ROM	1844	11.6	d-ROM	250.95	11.7
Age	1608	10.1	Age	220.44	10.2
Gender	1472	9.3	Gender	198.88	9.2
Hypertension	1080	6.8	Hypertension	147.49	6.9
Family CAD history	961	6.0	Family CAD history	130.37	6.1
BMI	865	5.4	BMI	117.89	5.5
Eh GSH/GSSG	716	4.5	Eh GSH/GSSG	95.65	4.4
Systolic BP	598	3.8	Systolic BP	80.50	3.7
Fasting insulin	581	3.7	Fasting insulin	77.57	3.6
Myeloperoxidase	524	3.3	Myeloperoxidase	71.68	3.3
Framingham risk score	514	3.2	Framingham risk score	69.28	3.2
hs-CRP	499	3.1	HDL	67.20	3.1
CySS	493	3.1	hs-CRP	66.84	3.1
HDL	490	3.1	CySS	65.26	3.0
Eh Cys/CySS	431	2.7	Eh Cys/CySS	57.45	2.7
CySSG	404	2.5	CySSG	54.25	2.5
Cys	389	2.4	Diastolic BP	53.55	2.5
Diastolic BP	382	2.4	Cys	52.29	2.4
Prior smoking	363	2.3	Prior smoking	49.26	2.3
GSH	342	2.2	GSH	47.08	2.2
TG (Triglyceride)	285	1.8	Diabetes	37.96	1.8
Diabetes	283	1.8	TG (Triglyceride)	37.90	1.8
GSSG	266	1.7	GSSG	35.44	1.6
TC (Cholesterol)	261	1.6	TC (Cholesterol)	34.99	1.6
LDL	240	1.5	LDL	31.86	1.5
Total	15891	99.9	Total	2152.03	99.9

4.3 Sulfur Amino Acid Intake

In this study, we investigate whether variation in sulfur amino acid (SAA) intake affects on metabolic changes in human plasma via ^{1}H NMR.

4.3.1 Background

This section describes the background of this study, including study of sulfur amino acid intake, subjects, diet and nutrient intake, ^{1}H NMR spectroscopy, and data preprocessing.

4.3.1.1 Study of Sulfur Amino Acid Intake

Influence of sulfur amino acid deficiency is studied in 1954 by Fillios and Mann [35]. Tor-Agbidye et al. [156] study the relationship between blood cyanide and plasma cyanate concentrations on rats by controlling the SAAs in the diet. Paterson et al. [119] and Bobyn et al. [13] study the effects of sulfur amino acid deficiency on rat brain glutathione concentration by controlling the diets. Park et al. [115] study whether the SAA content of a meal affected postprandial plasma cysteine, cystine, or redox potential in humans and whether SAA intake level (adequate or inadequate) prior to the meal affected these postprandial levels.

This study was conducted as a 13-day study of effects of diet on plasma GSH/GSSG redox state. The overall design included a 3-d equilibration on normal SAA containing food, 5-d SAA free food, and 5-d SAA containing food. On the first and last day of each 5-d period, hourly blood draws were taken for plasma metabolomic analyses by NMR spectroscopy. On the other days of each 5-d period, blood draws were taken at 8:30 AM before the breakfast.

4.3.1.2 Subjects

Studies were performed with informed consent under a protocol approved by the Emory Investigational Review Board. Average of age of all subjects was 23±2.93 (Mean±SD). The average of BMI in all individuals was 21.8±1.13 (Mean±SD). Male was 57% and female was 43% of all participants. The race distribution was black (43%), white (43%), and Asian (14%). All subjects were screened in the outpatient unit of the Emory General Clinical Research Center (GCRC), where a history and physical examination, body height and weight, fasting standard blood chemistry and hematology tests and a urinalysis were performed (a serum pregnancy test was also performed in menstruating females). Indirect calorimetry was performed to determine resting energy expenditure (REE). Eligible subjects at the time of the study had to be within 10% of the ideal body weight for height. Individuals who currently smoked were excluded.

Subjects being treated for hypertension were eligible for the study, butthose taking chronic medications for other illnesses or with evidence of any acute disease process were excluded. Because GSH redox state varies with age after 45 y, subjects between 18 and 40 y were recruited, with an approximately equal number of males and females. Subjects were asked to discontinue antioxidants and nutritional supplements (with the exception of once-daily multivitamin-mineral supplements) or acetaminophen two weeks prior to the onset of the studies. Menstruating females were scheduled for study in the follicular phase of their menstrual cycle, defined as between 7 to 10 days after the onset of the last period. Within one month following screening, the subjects were scheduled to begin the study. During the 3-day equilibration period, nutritionally balanced meals providing the RDA for SAA were provided by the GCRC Bionutrition Unit Subjects. Following the equilibration period, subjects were placed on the 0% SAA diet for a 5-day depletion period and then an isoenergetic,

isonitrogenous diet with 3X the RDA for SAA for a 5-day repletion period while remaining in the GCRC inpatient unit.

4.3.1.3 Diet and Nutrient Intake

The SAA-free and SAA containing were isonitrogenous and isoenergetic. The protein equivalent was supplied in the form of specific L-amino acid mixtures, providing 1.0 g/kg per day as outlined in detail. The standard mixture was patterned after hen's egg protein and provided all 9 indispensable (essential) amino acids, including Met, in amounts sufficient for the mean requirements of healthy young adults, but which were higher than the requirements proposed by the World Health Organization. The standard amino acid mixture also contained 8 dispensable (non-essential) amino acids, including Cys and Glu, and was Gln- and taurine-free. To compensate for the difference in Met + Cys between the SAA-free, 0% and SAA diet, the amount of all non-essential amino acids was proportionally changed to maintain a constant dietary nitrogen content. Met:Cys was at the ratio in the RDA (1:4) in SAA containing diet. To improve palatability, a powdered flavoring agent was added to the amino acid mixture. The dietary energy (1.4 times measured REE) was mainly derived from lipid and carbohydrate sources provided in the form of protein-free wheat starch and butter/safflower oil cookies and a sherbet-based drink. Experimental diets were administered by the GCRC nutritionists on a standard schedule; meals at 8:30 AM, 12:30 PM and 5:30 PM and an evening "snack" at 9:30 PM. All meals and snacks were to be consumed over no longer than a 20-minute period. Subjects were highly compliant with these research dietary items. Adequate hydration and vitamin, mineral and electrolyte requirements were provided to all subjects to meet or exceed recommended allowances. Ad libitum intake of water was provided to ensure urine output of at least 700 ml during each 24-h urine collection. All subjects received on

a daily basis 1) one multivitamin-multimineral capsule with iron (One-A-Day; Miles Inc., Elkhart, IN); 2) three potassium tablets (K-LYTE;20 meq each, generic); 3) four calcium tablets (TUMS; SKB Corp., Pittsburgh, PA); 4) two sodium chloride tablets (1 gram tablets; Eli Lilly and Co., Indianapolis, IN); 5) two choline capsules (250-mg; Lee Nutrition Inc., Cambridge, MA), and 6) one magnesium oxide tablet (400 mg tablet). All supplements were administered on a regular schedule by the GCRC research nurses. Body weights were determined daily and vital signs were obtained every 8 h. Low-level activity was allowed and restricted to walking on the GCRC.

4.3.1.4 ^{1}H NMR Spectroscopy

Plasma samples were thawed (600 ml) and mixed with 66 ml of deuterium oxide (D2O) containing DSS [3-(trimethylsilyl)-1-propanesulfonic acid sodium salt (C6H15NaO3SSi, 1% w/w)]. ^{1}H NMR spectra were measured at 600 MHz on a Varian INOVA 600 spectrometer with water presaturation at 25°C. The samples were maintained at 25°C in the magnet at least 10 minutes before measurement in order to ensure temperature stability. All spectra were referenced to the internal standard, DSS, and corrected for phase and baseline to standardize the data after Fourier transform. NMR spectra were measured with 64 scans into 19,802 data points over a spectral width of 6600.7 Hz, which resulted in an acquisition time of 2.55 s per sample (d1=0, pulse=5 ms, presaturation=1 s , acquisition = 1.5 s). To check the reproducibility of the NMR analysis, the plasma was purchased to run NMR on multiple time points (1.5 h, 3h, 4h and 6h). The correlation and coefficients of spectra were 0.96, 0.93, 0.97, and 0.97.

Spectral data from blood samples of 5 subjects are categorized into two groups: Data of the depletion period are considered as group -SAA while data of the repletion period are considered as +SAA. However, data taken at 8:30 AM of the first day of

the depletion period is put in group +SAA and data taken at 8:30 AM of the first day of the repletion period is put in group -SAA. Among the 158 spectral samples 85 samples are in group -SAA and 73 are in group +SAA.

4.3.1.5 Data Preprocessing

The preprocessing of the spectral data includes binning, baseline correction, and normalization, which are based on the procedures in Ressom el al. [132, 131].

(1) Binning:

First we bin the raw spectral data to reduce the noise as well as the dimensionality of the data. We choose the range of chemical shift values which are contained in all spectra and then bin the data with bin size 11. In each bin the mean of the 11 corresponding intensities represents the intensity of this bin. After binning, each spectrum has 1486 features.

(2) Baseline correction:

We perform baseline correction to reduce the effect of background noise. The baseline (background value) of each spectrum is estimated by using shifting windows. Shifting windows are calculated every 30 bins with window size 100 bins. That is, the shifting windows are overlapping. The baseline at every window is estimated by taking the 10% quantile value. The spline function in MATLAB is used to do spline approximation (cubic spline interpolation). The regressed baseline is then subtracted from the spectrum.

(3) Normalization:

We perform normalization to reduce variation in intensity of chemical shift between

Table 4.3.1: First classification results on SAA

Ten-fold cross-validation

	-SAA	+SAA	-SAA	+SAA
-SAA	50	4	0.9259	0.0741
+SAA	2	44	0.0435	0.9565
Overall accuracy: 0.9400				

Blind prediction

	-SAA	+SAA	-SAA	+SAA
-SAA	28	3	0.9032	0.0968
+SAA	1	26	0.0370	0.9630
Overall accuracy: 0.9310				

spectra. The intensities are scaled by (i) dividing by the total intensities of the spectrum (i.e. the area under the curve) and (ii) multiplying 10^7.

4.3.2 Results

We randomly select 100 spectral samples for training by 10-fold cross-validation and use the remaining 58 samples for blind prediction. We apply PSO/DAMIP to select 10 features and we are able to discriminate the two groups with accuracy greater than 90% in both 10-fold cross-validation and blind prediction. Here we show two selected feature sets and the corresponding results. The ten selected features (chemical shift values) in the first result are [9.7631, 8.9280, 7.3762, 6.8442, 6.0904, 4.7677, 4.0509, 2.7503, 2.6247, 0.8955]; the ten selected features in the second result are [8.6694, 7.3614, 7.1102, 6.0165, 2.8759, 2.5212, 2.1517, 1.9744, 1.5827, -0.2647]. Table 4.3.1 and 4.3.2 show the classification accuracies. Figure 4.3.1 and 4.3.2 show the selected features plotted on the spectra.

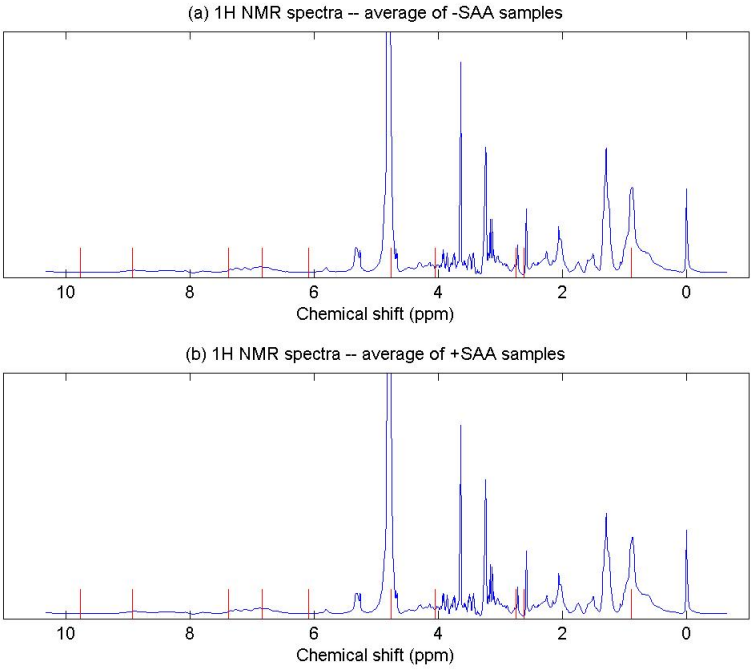

Figure 4.3.1: Selected features from the first result on the spectra. (a) -SAA samples, (b) +SAA samples.

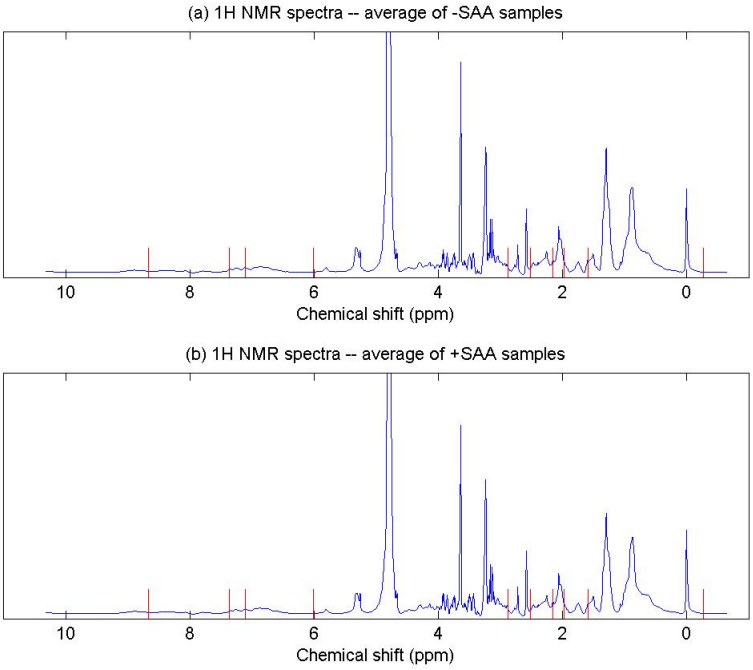

Figure 4.3.2: Selected features from the second result on the spectra. (a) -SAA samples, (b) +SAA samples.

Table 4.3.2: Second classification results on SAA

Ten-fold cross-validation

	-SAA	+SAA	-SAA	+SAA
-SAA	50	4	0.9259	0.0741
+SAA	4	42	0.0870	0.9130
Overall accuracy: 0.9200				

Blind prediction

	-SAA	+SAA	-SAA	+SAA
-SAA	30	1	0.9677	0.0323
+SAA	2	25	0.0741	0.9259
Overall accuracy: 0.9483				

Our classification model is able to discriminate the ^{1}H NMR spectra of blood plasma samples which relate to no-SAA-intake diets and multiple-SAA-intake diets. The results demonstrate that our model can help for metabolomics study.

4.4 CpG Islands

4.4.1 Background

DNA (deoxyribonucleic acid) is the hereditary material in humans and almost all organisms which contains the genetic instructions to construct other components of cells, such as proteins. DNA consists of two strands of repeated units called nucleotides. A nucleotide is composed of a nucleobase, a five-carbon sugar, and one to three phosphate groups. The order of the nucleotides, or nucleobases, determines the information of DNA. The four kinds of DNA nucleobases, simply called bases, are adenine, thymine, cytosine, and guanine, abbreviated as A, T, C, and G, respectively.

The two strands of DNA form a spiral called a double helix. A DNA strand can only be synthesized in vivo in a particular direction: from 5'-end to 3'-end. By convention, a single DNA strand is written in the 5'-3' direction. In a double helix, the directions of the two strands are opposite to each other; the 5'-end in one strand is paired with the 3'-end in the other strand. Furthermore, the binding bases of the two strands are paired: (1) A paired with T and (2) C paired with G. That is, AT and CT are the two types of DNA base pairs. Base pair, abbreviated as bp, is also a measurement of the length of a DNA sequence. Figure 4.4.1 illustrates two strands of DNA with some bases; the base sequence 'ATTG' on strand 1 has its complementary base sequence 'CAAT' on strand 2. (Note that a sequence is read in the 5'-3' direction by convention.)

CpG islands are short stretches of DNA enriched for the dinucleotide, 5'-CpG-3', which is the substrate for methylation. The letter 'p' indicates that C and G are connected by a phosphodiester bond. Although most CpG islands remain unmethylated in normal adult cells, they can become methylated de novo in human cancer cells. This aberrant methylation of CpG islands plays a critical role in the initiation

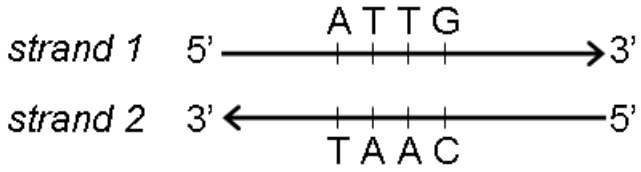

Figure 4.4.1: Example of base pairs on DNA.

Table 4.4.1: Categorization of CpG islands

Category of CpG islands	Range of methylation frequency	Number of sequences
methylation-prone	methy. freq. ≥ 10	69
methylation-sporadic	$1 \leq$ methy. freq. ≤ 9	68
methylation-resistent	methy. freq. $= 0$	143

and progression of cancer. We are interested in looking for sequence signatures which are capable of distinguish between methylation-prone and methylation-resistant CpG islands [31, 32].

4.4.2 Data Description

The input sequences consist of 280 CpG islands with length ranging from 500 to 6118 bps. Each sequence is associated with a methylation frequency, ranging from 0 to 24. A sequence is categorized into methylation-prone, sporadic, or resistant according to its methylation frequency. The range of methylation frequency and the number of sequences for each type of CpG islands are listed in Table 4.4.1.

Initially, we are interested in identifying patterns to discriminate methylation-prone versus methylation-resistant. Besides having CpG islands, we also have the sequences outside the CpG islands, specifically 1000 bps on each side of the CpG islands. In our analysis, we look for discriminatory patterns using the following three

regions of DNA sequences:

1. CpG islands: The sequences of CpG islands.
2. Extended: The sequences of CpG islands plus 1000 bps on each side of the CpG islands.
3. Outer: The sequences on each side of the CpG islands, 1000 bps on each side.

These three cases are treated independently.

4.4.3 Pattern Search, Feature Selection, and Classification

A *pattern* is a short sequence of letters, including A, T, C, G, or others, which is the attribute that we are looking for to discriminate between methylation-prone and methylation-resistant CpG islands. We allow the letter to be not only A, T, C, and G (called a *match letter*) but also other ones which represent two or more bases (called a *wild letter*). Table 4.4.2 lists all letters and their meanings, i.e., possible bases, and Table 4.4.3 lists the letters and their complements.

We also consider the reverse complement of a pattern. That is, we look for patterns together with their reverse complements. For example, 'ATTG' and 'CAAT' are treated together as one pattern, as shown in Figure 4.4.2.

Figure 4.4.2: Example of patterns and their reverse complements on DNA.

Table 4.4.2: Letters and their meanings

Letter	Meaning
A	A
T	T
C	C
G	G
R	G A
Y	T C
M	A C
K	G T
S	G C
W	A T
H	A C T
B	G T C
V	G C A
D	G A T
N	A T C G

Table 4.4.3: Complements of letters

Letter	A	T	C	G	R	Y	M	K	S	W	H	B	V	D	N
Complement	T	A	G	C	Y	R	K	M	S	W	D	V	B	H	N

4.4.3.1 Methodology

We derive procedures for searching patterns that will discriminate methylation-prone and methylation-resistant CpG islands. The algorithm consists of five steps:

1. Apply pattern search algorithms to generate possible patterns.
2. Filter the massive amount of possible patterns and generate some pattern pools by certain criteria.
3. On each pattern pool, apply feature selection and classification methods to select 1 to 20 discriminating patterns.
4. Aggregate patterns which have good classification results from Step 3. These patterns are supposed to be the ones we are looking for. We generate pattern pools again from these "good" patterns.
5. Apply the feature selection and classification methods on the pattern pools to validate the discriminating power of the patterns found in Step 4.

Given a pattern, for example, 'ATBG', we calculate how many times it appears in each sequence of CpG islands; the number of appearance is called the *occurrence frequency* of this pattern. Besides using the occurrence frequency, we can also use the normalized occurrence frequency, where the occurrence frequency is divided by the length of the sequence in the unit of 1000 bps.

We use three fourth of the data for training (159 sequences) and the rest for blind prediction, with the pattern search being applied only to the training set.

Step 1: Generate Possible Patterns

We first generate all possible patterns which have fixed numbers of match and wild letters such that the sum of the (non-normalized or normalized) occurrence frequencies of this pattern in the training sequences is greater than or equal to a frequency threshold. Ten is a frequency threshold to first exclude patterns that appear infrequently among all training sequences. We choose the numbers of match letters from 3 to 15 and the numbers of wild letters from 0 to 2.

First we show the structure of the loops of the pattern generating algorithm, and then we describe the details of (a), (b), (c), (d), and (e) in the looping structure.

```
Given the numbers of match and wild letters
loop for all sequences
    loop for all positions (a)
        loop for all possible patterns with respect to wild letters (b)
            if pattern has not been visited (c)
                store the pattern and its reverse complement (c)
                calculate the occurrence frequencies in all sequences (d)
                output the calculation if threshold is met (e)
            end if
        end loop
    end loop
end loop
```

(a): Given a sequence i, the loop goes from the first position of sequence i to the last possible position to include a pattern. For example, suppose the numbers of match and wild letters are 8 and 2, respectively, and the length of sequence i is 1000 bps, then loop (a) goes from the 1^{st} to the 991^{th} position of sequence i.

(b): Given a (starting) position of sequence i, we have the original pattern, then we loop for all possible patterns with respect to wild letters. For example, suppose

the numbers of match and wild letters are 3 and 1, respectively; suppose the original pattern with length four read from a certain position of sequence i is ATTG. Based on the original pattern ATTG, loop (b) goes through all possible patterns which consist of 3 match letters and 1 wild letter such that the original pattern is an instance of those possible patterns. In this example, ATTR, ATTB, ATYG, ANTG, HTTG, etc., and many other ones are all possible patterns derived from ATTG. Note that we do not allow N, the wild letter which represents all four cases, to appear in the first or last position of a pattern. In this example, we do not look for patterns NTTG and ATTN. Patterns with letter N in the ends (first and last positions) are exactly the shorter patterns without the N's.

(c): Given a pattern generated in loop (b), for example, ATTR, if it has been visited, we do nothing here; otherwise we will store both this pattern and its reverse complement, YAAT, in a hash table. This allows us to use a hash function to map a number to each pattern. We assign 0, 1, 2, 3 to A, T, C, G, respectively, and we define the hash function such that it maps the first seven match letters of a pattern to a quaternary number. For example, ATTR, ATTY, ATMT, ATKT, AHTT, ANTT, BATT are all mapped to 011, which represents the first match letters ATT. In another example, AATTCCGGNM, AATTCCGANM, RAATVTCCGT are all mapped to 0011223, which represents the first seven match letters. We use the chaining strategy for the hash table: Each slot of the array is a pointer to a linked list containing the patterns which have the same hash function values. That is, a pattern is inserted into the end of a particular linked list based on the hash function value, and it can be searched in the time linear to the length of the linked list.

(d): Given a pattern generated in loop (b) that has not been found, for example, ATTR, we count and add up the occurrence frequencies of ATTR and its reverse

complement YAAT in a sequence. Non-normalized and normalized frequencies are both stored. We do the same thing for all sequences.

(e): Given a pattern, if the sum of the occurrence frequencies in all sequences is greater than or equal to the threshold frequency, this pattern together with its reverse complement and the occurrence frequencies are output into the file.

Step 2: Generate Pattern Pools I

Any subsets of patterns found in Step 1 can be candidate patterns in our feature selection and classification model. Further, we can use patterns with certain length or with certain number of wild letters. However, the number of patterns remains very large, in the order of millions. To reduce the numbers, we screen the patterns by some criteria. Specifically, we put together all the patterns with different lengths, and screen the patterns using the following criteria to generate pattern pools:

1. Use or not use wild letters. Denoted by W or no-W.

2. Occurrence frequencies are non-normalized or normalized. Denoted by non-N or N.

3. The sum of occurrence frequencies in the training sequences is greater than or equal to a pre-specified threshold, denoted by TH_1.

4. The correlation coefficient between methylation frequency and occurrence frequency is greater than or equal to a threshold (a positive value), denoted by TH_2.

5. The absolute value of the correlation coefficient between methylation frequency and occurrence frequency is greater than or equal to a threshold (a positive

value), denoted by TH_3.

In our analysis, either Criterion 4 or 5 is being used. When we use Criterion 4, we look for patterns such that the methylation frequency and pattern occurrence frequency are positively correlated. On the other hand, Criterion 5 screens for patterns in which the methylation frequency and pattern occurrence frequency are both positively and negatively correlated.

Step 3: Feature Selection and Classification

On each pattern pool, we apply feature selection and classification methods described in Chapter 3 to select 1 to 20 discriminating patterns.

Step 4: Generate Pattern Pools II

After feature selection and classification is performed in Step 3, we obtain sets of patterns which have good classification accuracies for both groups in both training and blind prediction. Patterns are aggregated for those that result in accuracies that are greater than or equal to a certain level, for CpG islands, extended, outer, and all of the three.

To validate that these aggregated patterns are discriminating, we calculate the occurrence frequencies and normalized occurrence frequencies of these aggregated patterns, and generate several pattern pools to perform feature selection and classification again. Pattern pools are generated according to (1) thresholds of accuracies in the results of Step 3 (80% and 75%), and (2) parts of the results (CpG islands, extended, outer, and all three of them). If a pattern pool is obtained from the results of, for example, CpG islands, then the occurrence frequencies are calculated only in

the part of CpG islands and the classification is done for the CpG island part. On the other hand, if a pattern pool is obtained from the results of CpG island, extended, and outer, then the occurrence frequencies are calculated separately in three parts and the classification is done separately in three parts.

Step 5: Feature Selection and Classification

We apply the feature selection and classification methods described in Chapter 3 on the pattern pools to validate the discriminating power of the patterns found in Step 4.

4.4.4 Results

Results of Step 1: Generate Possible Patterns

Table 4.4.4 shows the number of patterns found in Step 1. Here we consider the number of match letters from 1 to 15 and the number of wild letters from 0 to 2. The length of a pattern is the sum of the numbers of match and wild letters. Partly due to the strenuous computational effort, less number of wild letters will be considered when the number of match letters becomes larger. Further, longer patterns appear less frequently in the sequences, and thus will be excluded by the frequency threshold (TH_1). Empirically, 15 appears to be an appropriate length.

Results of Step 2: Generate Pattern Pools I

Table 4.4.5 shows the number of patterns screened under the five criteria: wild or non-wild, normalized or non-normalized, TH_1 (threshold on the sum of occurrence

Table 4.4.4: Number of patterns under fixed numbers of match and wild letters

#Match	#Wild	Length	CpG islands		Extended		Outer	
			non-N	N	non-N	N	non-N	N
1	2	3	640	640	640	640	640	640
2	1	3	254	254	254	254	254	254
2	2	4	5330	5330	5330	5330	5330	5330
3	0	3	32	32	32	32	32	32
3	1	4	1344	1344	1344	1344	1344	1344
3	2	5	35936	35936	35936	35936	35936	35936
4	0	4	136	136	136	136	136	136
4	1	5	6808	6808	6808	6808	6808	6808
4	2	6	218624	218624	218624	218624	218624	218624
5	0	5	512	512	512	512	512	512
5	1	6	32768	32768	32768	32768	32768	32765
5	2	7	1233878	1231291	1233920	1233094	1233667	1228586
6	0	6	2076	2043	2080	2059	2072	2021
6	1	7	152027	140257	153677	147010	150906	139606
7	0	7	6826	4513	8051	5721	6815	5596
7	1	8	459415	246268	658705	310923	517281	346788
8	0	8	9150	3476	21709	3122	12752	3742
8	1	9	502586	170024	1394844	118666	631741	117181
9	0	9	5717	1552	15372	673	3621	468
10	0	10	2315	386	4771	146	747	199
11	0	11	590	74	1379	60	449	142
12	0	12	146	14	631	44	353	108
13	0	13	41	4	395	28	291	78
14	0	14	15	3	314	15	243	64
15	0	15	10	3	263	9	207	53

Table 4.4.5: Number of patterns under specific criteria

Pool	Criteria					Number of patterns		
index	Wild letter	Normalized	TH_1	TH_2	TH_3	CpG islands	Extended	Outer
0	W	non-N	50	0.15	na	82549	111152	75427
1	W	non-N	50	0.20	na	14698	21942	19561
2	W	non-N	50	0.25	na	1812	3225	3643
3	W	non-N	50	na	0.15	94413	154992	140516
4	W	non-N	50	na	0.20	16028	29091	31611
5	W	non-N	50	na	0.25	1907	3898	4835
6	W	N	50	0.15	na	33958	59669	56664
7	W	N	50	0.20	na	6375	12334	13800
8	W	N	50	0.25	na	779	1919	2334
9	W	N	50	na	0.15	69601	132619	114882
10	W	N	50	na	0.20	11897	25742	24843
11	W	N	50	na	0.25	1273	3248	3473
12	no-W	non-N	10	0.10	na	4375	7454	3028
13	no-W	non-N	10	na	0.10	6225	12746	6610
14	no-W	non-N	10	na	na	27566	55645	28230
15	no-W	non-N	30	na	na	8533	17041	8670
16	no-W	N	10	0.10	na	1381	1461	1404
17	no-W	N	10	na	0.10	2637	3071	3261
18	no-W	N	10	na	na	12748	12557	13151
19	no-W	N	30	na	na	4105	3982	4030

frequencies), TH_2 (threshold on correlation coefficient), and TH_3 (threshold on the absolute value of correlation coefficient). We generate 20 pattern pools using different parameters for the criteria.

Results of Step 3: Feature Selection and Classification

For each pattern pool (pool $0 \sim 19$), feature selection and classification is performed to select from 1 to 20 features to form the discriminatory sets. To filter the results, accuracy threshold is applied to the classification accuracies of Group 1 and Group 2 in both ten-fold cross validation and blind prediction.

Table 4.4.6: Number of patterns in pattern pools for CpG islands

Pool index	Source of patterns	Normalized	Number of patterns
0	80%, CpG, extended, outer	non-N	2710
1		N	
2	80%, CpG	non-N	2352
3		N	
4	75%, CpG, extended, outer	non-N	41287
5		N	
6	75%, CpG	non-N	27480
7		N	
8	75%, CpG, #appearnace≥20	non-N	3444
9		N	
10	75%, CpG, #appearnace≥50	non-N	1724
11		N	
12	75%, CpG, #appearnace≥100	non-N	935
13		N	

Results of Step 4: Generate Pattern Pools II

Table 4.4.6, 4.4.7, and 4.4.8 show the way we obtain each pattern pool and the number of patterns in each pool in CpG island, extended, and outer, respectively. Note that in the case of CpG islands, we have 27,480 patterns from the results of 75%; we generate further pattern pools from these where the number of appearance of each pattern is greater than 20, 50, or 100. The pattern pool formed by more frequently appeared patterns are expected to be more discriminating.

Results of Step 5: Feature Selection and Classification

For each pattern pool, we run feature selection and classification to select 3 to 10 patterns. Compared to pattern pools obtained from Step 2, the pools from Step 4 contain more discriminating patterns, thus the results here are much better.

Table 4.4.7: Number of patterns in pattern pools for extended

Pool index	Source of patterns	Normalized	Number of patterns
0	80%, CpG, extended, outer	non-N	2710
1		N	
2	80%, extended	non-N	136
3		N	
4	75%, CpG, extended, outer	non-N	41287
5		N	
6	75%, extended	non-N	7803
7		N	

Table 4.4.8: Number of patterns in pattern pools for outer

Pool index	Source of patterns	Normalized	Number of patterns
0	80%, CpG, extended, outer	non-N	2710
1		N	
2	80%, outer	non-N	235
3		N	
4	75%, CpG, extended, outer	non-N	41287
5		N	
6	75%, outer	non-N	8307
7		N	

Table 4.4.9: Classification results on CpG islands, selecting nine features: (ANGGCHA, TDGCCNT), (BGSAA, TTSCV), (CCCBGTK, MACVGGG), (AACCBBA, TVVGGTT), (AAGTVAV, BTBACTT), (AGMGTTR, YAACKCT), (CAHGWTG, CAWCDTG), (CGCCCGCGC, GCGCGGGCG), (GTCGCDD, HHGCGAC)

Ten-fold cross-validation

	methylation -prone	methylation -resistent	methylation -prone	methylation -resistent
methylation-prone	46	6	0.8846	0.1154
methylation-resistent	12	95	0.1121	0.8879
Overall accuracy: 0.8868				

Blind prediction

	methylation -prone	methylation -resistent	methylation -prone	methylation -resistent
methylation-prone	16	1	0.9412	0.0588
methylation-resistent	5	31	0.1389	0.8611
Overall accuracy: 0.8868				

We list some of the classification results of Step 5 in which the overall and group accuracies in cross-validation training and blind prediction are greater than or equal to 85%. Table 4.4.9 and 4.4.10 are for CpG islands, Table 4.4.11 and 4.4.12 are for extended, and Table 4.4.13 and 4.4.14 are for outer.

Table 4.4.10: Classification results on CpG islands, selecting ten features: (AGCYAGS, SCTRGCT), (CGGCGGASG, CSTCCGCCG), (AAGTMAV, BTKACTT), (AHYTACC, GGTARDT), (TANGTNA, TNACNTA), (CAGAWTD, HAWTCTG), (HCKGTGA, TCACMGD), (BATCSAA, TTSGATV), (CASWAGG, CCTWSTG), (AKTDGAA, TTCHAMT)

Ten-fold cross-validation

	methylation -prone	methylation -resistent	methylation -prone	methylation -resistent
methylation-prone	45	7	0.8654	0.1346
methylation-resistent	12	95	0.1121	0.8879
Overall accuracy: 0.8805				

Blind prediction

	methylation -prone	methylation -resistent	methylation -prone	methylation -resistent
methylation-prone	15	2	0.8824	0.1176
methylation-resistent	4	32	0.1111	0.8889
Overall accuracy: 0.8868				

Table 4.4.11: Classification results on Extended, selecting ten features: (CAHTAGK, MCTADTG), (GVCTKTA, TAMAGBC), (AGGTDTV, BAHACCT), (CAMTAGB, VCTAKTG), (CSCACCCCC, GGGGGTGSG), (ACGTAVM, KBTACGT), (ABTCCYA, TRGGAVT), (CGGHANA, TNTDCCG), (BAGGTKC, GMACCTV), (BTACAGY, RCTGTAV)

Ten-fold cross-validation

	methylation -prone	methylation -resistent	methylation -prone	methylation -resistent
methylation-prone	45	7	0.8654	0.1346
methylation-resistent	12	95	0.1121	0.8879
Overall accuracy: 0.8805				

Blind prediction

	methylation -prone	methylation -resistent	methylation -prone	methylation -resistent
methylation-prone	15	2	0.8824	0.1176
methylation-resistent	3	33	0.0833	0.9167
Overall accuracy: 0.9057				

Table 4.4.12: Classification results on Extended, selecting ten features: (GGCABTD, HAVTGCC), (AVGGCWA, TWGCCBT), (DGCTGCAA, TTGCAGCH), (ACACAGVG, CBCTGTGT), (CAMTAGB, VCTAKTG), (CSCACCCCC, GGGGGTGSG), (AACTRRG, CYYAGTT), (CGGHAHA, TDTDCCG), (GGCTGGAA, TTCCAGCC), (GGGAGAAA, TTTCTCCC)

Ten-fold cross-validation

	methylation -prone	methylation -resistent	methylation -prone	methylation -resistent
methylation-prone	45	7	0.8654	0.1346
methylation-resistent	9	98	0.0841	0.9159
Overall accuracy: 0.8994				

Blind prediction

	methylation -prone	methylation -resistent	methylation -prone	methylation -resistent
methylation-prone	15	2	0.8824	0.1176
methylation-resistent	5	31	0.1389	0.8611
Overall accuracy: 0.8679				

Table 4.4.13: Classification results on Outer, selecting ten features: (GAWSGAC, GTCSWTC), (AVCTGGCC, GGCCAGBT), (CDGTYG, CRACHG), (RCCGANA, TNTCGGY), (AGATNGS, SCNATCT), (AHGHTAG, CTADCDT), (CAHTAGK, MCTADTG), (CTTWRAC, GTYWAAG), (CDAACCD, HGGTTHG), (KATCCAM, KTGGATM)

Ten-fold cross-validation

	methylation -prone	methylation -resistent	methylation -prone	methylation -resistent
methylation-prone	46	6	0.8846	0.1154
methylation-resistent	16	91	0.1495	0.8505
Overall accuracy: 0.8616				

Blind prediction

	methylation -prone	methylation -resistent	methylation -prone	methylation -resistent
methylation-prone	16	1	0.9412	0.0588
methylation-resistent	5	31	0.1389	0.8611
Overall accuracy: 0.8868				

Table 4.4.14: Classification results on Outer, selecting ten features: (GAWSGAC, GTCSWTC), (CDGTYG, CRACHG), (RCCGANA, TNTCGGY), (ATAMGCH, DGCKTAT), (ATGMTAG, CTAKCAT), (AGATNGS, SCNATCT), (CAHTAGK, MCTADTG), (CTTWRAC, GTYWAAG), (CDAACCD, HGGTTHG), (KATCCAM, KTGGATM)

Ten-fold cross-validation

	methylation -prone	methylation -resistent	methylation -prone	methylation -resistent
methylation-prone	46	6	0.8846	0.1154
methylation-resistent	11	96	0.1028	0.8972
Overall accuracy: 0.8931				

Blind prediction

	methylation -prone	methylation -resistent	methylation -prone	methylation -resistent
methylation-prone	15	2	0.8824	0.1176
methylation-resistent	5	31	0.1389	0.8611
Overall accuracy: 0.8679				

REFERENCES

[1] Abad, P. and Banks, W., "New LP based heuristics for the classification problem," *European Journal of Operational Research*, vol. 67, pp. 88–100, 1993.

[2] Agrafiotis, D. and Cedeno, W., "Feature selection for structure-activity correlation using binary particle swarms," *Journal of Medicinal Chemistry*, vol. 45, no. 5, pp. 1098–1107, 2002.

[3] Anderson, J., "Constrained discrimination between k populations," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 31, no. 1, pp. 123–139, 1969.

[4] Asparoukhov, O. and Stam, A., "Mathematical programming formulations for two-group classification with binary variables," *Annals of Operations Research*, vol. 74, pp. 89–112, 1997.

[5] Bajgier, S. and Hill, A., "An experimental comparison of statistical and linear programming approaches to the discriminant problem," *Decision Sciences*, vol. 13, pp. 604–618, 1982.

[6] Banks, W. and Abad, P., "An efficient optimal solution algorithm for the classification problem," *Decision Sciences*, vol. 22, pp. 1008–1023, 1991.

[7] Banks, W. and Abad, P., "On the performance of linear programming heuristics applied on a quadratic transformation in the classification problem," *European Journal of Operational Research*, vol. 74, pp. 23–28, 1994.

[8] Bennett, K., "Decision tree construction via linear programming," in *Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society Conference* (Evans, M., ed.), pp. 97–101, 1992.

[9] Bennett, K. and Mangasarian, O., "Robust linear programming discrimination of two linearly inseparable sets," *Optimization Methods and Software*, vol. 1, pp. 23–34, 1992.

[10] Bennett, K. and Mangasarian, O., "Multicategory discrimination via linear programming," *Optimization Methods and Software*, vol. 3, pp. 27–39, 1994.

[11] Bertolazzi, P., Felici, G., Festa, P., and Lancia, G., "Logic classification and feature selection for biomedical data," *Computers and Mathematics with Applications*, vol. 55, no. 5, pp. 889–899, 2008.

[12] Blum, A. and Langley, P., "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, pp. 245–271, 1997.

[13] Bobyn, P., Franklin, J., Wall, C., Thornhill, J., Juurlink, B., and Paterson, P., "The effects of dietary sulfur amino acid deficiency on rat brain glutathione concentration and neural damage in global hemispheric hypoxia-ischemia," *Nutritional Neuroscience*, vol. 5, pp. 407–416, 2002.

[14] Bonami, P., Cornuejols, G., Dash, S., Fischetti, M., and Lodi, A., "Projected Chvatal-Gomory cuts for mixed integer linear programs," *Mathematical Programming*, vol. 113, no. 2, pp. 241–257, 2008.

[15] Bondi, M., Jak, A., Delano-Wood, L., Jacobson, M., Delis, D., and Salmon, D., "Neuropsychological contributions to the early identification of Alzheimers disease," *Neuropsychology Review*, vol. 18, pp. 73–90, 2008.

[16] Bradley, P. and Mangasarian, O., "Massive data discrimination via linear support vector machines," *Optimization Methods and Software*, vol. 13, no. 1, pp. 1–10, 2000.

[17] Brooks, J., *Solving a mixed-integer programming formulation of a classification model with misclassification limits*. PhD thesis, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, 2005.

[18] Brooks, J. and Lee, E., "Analysis of the consistency of a mixed integer programming-based multi-category constrained discriminant model," *Annals of Operations Research – Data Mining*, vol. 174, pp. 147–168, 2010.

[19] Brooks, J. and Lee, E., "Solving a mixed integer programming multi-category classification model with misclassification constraints." Revising, 2010.

[20] Bruzzone, L. and Serpico, S., "A technique for feature selection in multiclass problems," *International Journal of Remote Sensing*, vol. 21, no. 3, pp. 549–563, 2000.

[21] Burges, C., "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[22] Chen, C. and Mangasarian, O., "Hybrid misclassification minimization," *Advances in Computational Mathematics*, vol. 5, pp. 127–136, 1996.

[23] Clerc, M. and Kennedy, J., "The particle swarm—explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[24] Codato, G. and Fischetti, M., "Combinatorial Bender's cuts for mixed-integer linear programming," *Operations Research*, vol. 54, no. 4, pp. 756–766, 2006.

[25] Cooren, Y., Clerc, M., and Siarry, P., "Performance evaluation of TRIBES, an adaptive particle swarm optimization algorithm," *Swarm Intelligence*, vol. 3, pp. 149–178, 2009.

[26] Correa, E., Freitas, A., and Johnson, C., "A new discrete particle swarm algorithm applied to attribute selection in a bioinformatics data set," in *Genetic and Evolutionary Computation Conference*, (New York, NY), pp. 35–42, ACM, 2006.

[27] Dash, M. and Liu, H., "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, pp. 131–156, 1997.

[28] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R., "Least angle regression," *Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[29] Elbeltagi, E., Hegazy, T., and Grierson, D., "Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics*, vol. 19, pp. 43–53, 2005.

[30] Erenguc, S. and Koehler, G., "Survey of mathematical programming models and experimental results for linear discriminant analysis," *Managerial and Decision Economics*, vol. 11, pp. 215–225, 1990.

[31] Feltus, F., Lee, E., Costello, J., Plass, C., and Vertino, P., "Predicting aberrant CpG island methylation," *Proceedings of the National Academy of Sciences*, vol. 100, pp. 12253–12258, 2003.

[32] Feltus, F., Lee, E., Costello, J., Plass, C., and Vertino, P., "DNA signatures associated with CpG island methylation states," *Genomics*, vol. 87, pp. 572–579, 2006.

[33] Fernandez-Martinez, J. and Garcia-Gonzalo, E., "The generalized PSO: a new door to PSO evolution," *Journal of Artificial Evolution and Applications*, 2008. doi:10.1155/2008/861275. 15 pages.

[34] Fernandez-Martinez, J. and Garcia-Gonzalo, E., "The PSO family: deduction, stochastic analysis and comparison," *Swarm Intelligence*, vol. 3, pp. 245–273, 2009.

[35] Fillios, L. and Mann, G., "Influence of sulfur amino acid deficiency on cholesterol metabolism," *Metabolism-Clinical and Experimental*, vol. 3, pp. 16–26, 1954.

[36] Fisher, R., "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, pp. 179–188, 1936.

[37] Frank, A. and Asuncion, A., "UCI machine learning repository [http://archive.ics.uci.edu/ml]." University of California, Irvine, School of Information and Computer Sciences, 2010.

[38] Freed, N. and Glover, F., "A linear programming approach to the discriminant problem," *Decision Sciences*, vol. 12, pp. 68–74, 1981.

[39] Freed, N. and Glover, F., "Simple but powerful goal programming models for discriminant problems," *European Journal of Operational Research*, vol. 7, pp. 44–60, 1981.

[40] Freed, N. and Glover, F., "Evaluating alternative linear programming models to solve the two-group discriminant problem," *Decision Sciences*, vol. 17, pp. 151–162, 1986.

[41] Freed, N. and Glover, F., "Resolving certain difficulties and improving the classification power of LP discriminant analysis formulations," *Decision Sciences*, vol. 17, pp. 589–595, 1986.

[42] Fung, G. and Mangasarian, O., "Proximal support vector machine classifiers," in *Proceedings KDD-2001*, (San Francisco), August 26-29 2001.

[43] Fung, G. and Mangasarian, O., "Incremental support vector machine classification," in *Proceedings of the Second SIAM International Conference on Data Mining* (Grossman, R., Mannila, H., and Motwani, R., eds.), (Philadelphia), pp. 247–260, SIAM, 2002.

[44] Fung, G. and Mangasarian, O., "Multicategory proximal support vector machine classifiers," *Machine Learning*, vol. 59, pp. 77–97, 2005.

[45] Gallagher, R., Lee, E., and Patterson, D., "An optimization model for constrained discriminant analysis and numerical experiments with iris, thyroid, and heart disease datasets," in *Proceedings of the 1996 American Medical Informatics Association*, October 1996.

[46] Gallagher, R., Lee, E., and Patterson, D., "Constrained discriminant analysis via 0/1 mixed integer programming," *Annals of Operations Research*, vol. 74, pp. 65–88, 1997.

[47] Gehrlein, W., "General mathematical programming formulations for the statistical classification problem," *Operations Research Letters*, vol. 5, no. 6, pp. 299–304, 1986.

[48] Gleeson, J. and Ryan, J., "Identifying minimally infeasible subsystems of inequalities," *ORSA Journal on Computing*, vol. 2, no. 1, pp. 61–63, 1990.

[49] Glen, J., "Integer programming methods for normalisation and variable selection in mathematical programming discriminant analysis models," *Journal of the Operational Research Society*, vol. 50, pp. 1043–1053, 1999.

[50] Glen, J., "Classification accuracy in discriminant analysis: A mixed integer programming approach," *Journal of the Operational Research Society*, vol. 52, pp. 328–339, 2001.

[51] Glen, J., "An iterative mixed integer programming method for classification accuracy maximizing discriminant analysis," *Computers and Operations Research*, vol. 30, pp. 181–198, 2003.

[52] Glen, J., "Dichotomous categorical variable formation in mathematical programming discriminant analysis models," *Naval Research Logistics*, vol. 51, pp. 575–596, 2004.

[53] Glen, J., "Mathematical programming models for piecewise-linear discriminant analysis," *Journal of the Operational Research Society*, vol. 56, pp. 331–341, 2005.

[54] Glen, J., "A comparison of standard and two-stage mathematical programming discriminant analysis methods," *European Journal of Operational Research*, vol. 171, pp. 496–515, 2006.

[55] Glover, F., "Improved linear programming models for discriminant analysis," *Decision Sciences*, vol. 21, pp. 771–785, 1990.

[56] Glover, F., Keene, S., and Duea, B., "A new class of models for the discriminant problem," *Decision Sciences*, vol. 19, pp. 269–280, 1988.

[57] Gochet, W., Stam, A., Srinivasan, V., and Chen, S., "Multigroup discriminant analysis using linear programming," *Operations Research*, vol. 45, no. 2, pp. 213–225, 1997.

[58] Guyon, I. and Elisseeff, A., "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[59] Guyon, I., Weston, J., Barnhill, S., and Vapnik, V., "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, pp. 389–422, 2002.

[60] Hand, D., *Discrimination and classification*. New York: John Wiley, 1981.

[61] Hsu, C.-W. and Lin, C.-J., "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[62] Iannarilli, F. and Rubin, P., "Feature selection for multiclass discrimination via mixed-integer linear programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 779–783, 2003.

[63] Jiang, M., Luo, Y., and Yang, S., "Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm," *Information Processing Letters*, vol. 102, pp. 8–16, 2007.

[64] Joachimsthaler, E. and Stam, A., "Mathematical programming approaches for the classification problem in two-group discriminant analysis," *Multivariate Behavioral Research*, vol. 25, no. 4, pp. 427–454, 1990.

[65] John, G., Kohavi, R., and Pfleger, K., "Irrelevant features and the subset selection problem," in *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 121–129, 1994.

[66] Keerthi, S., "Generalized LARS as an effective feature selection tool for text classification with SVMs," in *Proceedings of the 22nd International Conference on Machine Learning*, pp. 417–424, 2005.

[67] Kennedy, J., "Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance," in *Proceedings of the 1999 Congress on Evolutionary Computation*, (Piscataway, NJ), pp. 1931–1938, IEEE, 1999.

[68] Kennedy, J. and Eberhart, R., "Particle swarm optimisation," in *Proceedings of the IEEE International Conference on Neural Networks*, (Piscataway, NJ), pp. 1942–1948, IEEE, 1995.

[69] Kennedy, J. and Eberhart, R., "A discrete binary version of the particle swarm algorithm," in *1997 IEEE International Conference on Systems, Man, and Cybernetics*, (Piscataway, NJ), pp. 4104–4108, IEEE, 1997.

[70] Kennedy, J. and Mendes, R., "Population structure and particle swarm performance," in *Proceedings of the 2002 Congress on Evolutionary Computation*, (Piscataway, NJ), pp. 1671–1676, IEEE, 2002.

[71] Kennedy, J. and Spears, W., "Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator," in *1998 IEEE international Conference on Evolutionary Computation*, (Piscataway, NJ), pp. 78–83, IEEE, 1998.

[72] Kluger, A., Ferris, S., Golomb, J., Mittelman, M., and Reisberg, B., "Neuropsychological prediction of decline to dementia in nondemented elderly," *Journal of Geriatric Psychiatry and Neurology*, vol. 12, pp. 168–179, 1999.

[73] Koehler, G., "Characterization of unacceptable solutions in LP discriminant analysis," *Decision Sciences*, vol. 20, pp. 239–257, 1989.

[74] Koehler, G., "Unacceptable solutions and the hybrid discriminant model," *Decision Sciences*, vol. 20, pp. 844–848, 1989.

[75] Koehler, G., "A response to Xiao's "necessary and sufficient conditions of unacceptable solutions in LP discriminant analysls": Something is amiss," *Decision Sciences*, vol. 25, pp. 331–333, 1994.

[76] Koehler, G. and Erenguc, S., "Minimizing misclassifications in linear discriminant analysis," *Decision Sciences*, vol. 21, pp. 63–85, 1990.

[77] Kohavi, R. and John, G., "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, pp. 273–324, 1997.

[78] Koller, D. and Sahami, M., "Toward optimal feature selection," in *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 284–292, 1996.

[79] Langley, P., "Selection of relevant features in machine learning," in *Proceedings of the AAAI Fall Symposium on Relevance*, (New Orleans, LA), pp. 140–144, AAAI Press, 1994.

[80] Laskari, E., Parsopoulos, K., and Vrahatis, M., "Particle swarm optimization for integer programming," in *Proceedings of the 2002 Congress on Evolutionary Computation*, (Piscataway, NJ), pp. 1582–1587, IEEE, 2002.

[81] Lee, E., "Large-scale optimization-based classification models in medicine and biology," *Annals of Biomedical Engineering – Systems Biology, Bioinformatics, and Computational Biology*, vol. 35, no. 6, pp. 1095–1109, 2007.

[82] Lee, E., Fung, A., Brooks, J., and Zaider, M., "Automated planning volume definition in soft-tissue sarcoma adjuvant brachytherapy," *Physics in Medicine and Biology*, vol. 47, pp. 1891–1910, 2002.

[83] Lee, E., Gallagher, R., Campbell, A., and Prausnitz, M., "Prediction of ultrasound-mediated disruption of cell membranes using machine learning techniques and statistical analysis of acoustic spectra," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 1, pp. 82–89, 2004.

[84] Lee, E., Gallagher, R., and Patterson, D., "A linear programming approach to discriminant analysis with a reserved-judgment region," *INFORMS Journal on Computing*, vol. 15, no. 1, pp. 23–41, 2003.

[85] Lee, Y., Lin, Y., and Wahba, G., "Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data," *Journal of the American Statistical Association*, vol. 99, pp. 67–81, 2004.

[86] Lee, Y.-J. and Mangasarian, O., "RSVM: Reduced support vector machines," in *Proceedings of the SIAM International Conference on Data Mining*, (Chicago), April 5-7 2001.

[87] Lee, Y.-J. and Mangasarian, O., "SSVM: A smooth support vector machine for classification," *Computational Optimization and Applications*, vol. 20, no. 1, pp. 5–22, 2001.

[88] Lee, Y.-J., Mangasarian, O., and Wolberg, W., "Breast cancer survival and chemotherapy: A support vector machine analysis," in *DIMACS Series in Discrete Mathematical and Theoretical Computer Science*, vol. 55, pp. 1–10, American Mathematical Society, 2000.

[89] Lee, Y.-J., Mangasarian, O., and Wolberg, W., "Survival-time classification of breast cancer patients," *Computational Optimization and Applications*, vol. 25, pp. 151–166, 2003.

[90] Liu, H. and Yu, L., "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[91] Liu, Y., Qin, Z., Xu, Z., and He, X., "Feature selection with particle swarms," in *Lecture Notes in Computer Science: Vol. 3314. Computational and Information Science: First International Symposium*, (Berlin/Heidelberg), pp. 425–430, Springer-Verlag, 2004.

[92] Lopez, O., Becker, J., Jagust, W., Fitzpatrick, A., Carlson, M., DeKosky, S., Breitner, J., Lyketsos, C., Jones, B., Kawas, C., and Kuller, L., "Neuropsychological characteristics of mild cognitive impairment subgroups," *Journal of Neurology, Neurosurgery, and Psychiatry*, vol. 77, pp. 159–165, 2006.

[93] Loucopoulos, C. and Pavur, R., "Computational characteristics of a new mathematical programming model for the three-group discriminant problem," *Computers and Operations Research*, vol. 24, no. 2, pp. 179–191, 1997.

[94] Loucopoulos, C. and Pavur, R., "Experimental evaluation of the classificatory performance of mathematical programming approaches to the three-group discriminant problem: The case of small samples," *Annals of Operations Research*, vol. 74, pp. 191–209, 1997.

[95] Luedi, P., Hartemink, A., and Jirtle, R., "Genome-wide prediction of imprinted murine genes," *Genome Research*, vol. 15, pp. 875–884, 2005.

[96] Mangasarian, O., "Linear and nonlinear separation of patterns by linear programming," *Operations Research*, vol. 13, pp. 444–452, 1965.

[97] Mangasarian, O., "Multi-surface method of pattern separation," *IEEE Transactions on Information Theory*, vol. 14, no. 6, pp. 801–807, 1968.

[98] Mangasarian, O., "Misclassification minimization," *Journal of Global Optimization*, vol. 5, pp. 309–323, 1994.

[99] Mangasarian, O., "Machine learning via polyhedral concave minimization," in *Applied Mathematics and Parallel computing – Festschrift for Klaus Ritter* (Fischer, H., Riedmueller, B., and Schaeffler, S., eds.), (Germany), pp. 175–188, Physica-Verlag, 1996.

[100] Mangasarian, O., "Arbitrary-norm separating plane," *Operations Research Letters*, vol. 24, pp. 15–23, 1999.

[101] Mangasarian, O., "Generalized support vector machines," in *Advances in Large Margin Classifiers* (Smola, A., Bartlett, P., Schökopf, B., and Schuurmans, D., eds.), (Cambridge, MA), pp. 135–146, MIT Press, 2000.

[102] Mangasarian, O., "Data mining via support vector machines," in *System Modeling and Optimization XX* (Sachs, E. and Tichatschke, R., eds.), (Boston), pp. 91–112, Kluwer Academic Publishers, 2003.

[103] Mangasarian, O., "Support vector machine classification via parameterless robust linear programming," *Optimization Methods and Software*, vol. 20, pp. 115–125, 2005.

[104] Mangasarian, O. and Musicant, D., "Successive overrelaxation for support vector machines," *IEEE Transactions on Neural Networks*, vol. 10, pp. 1032–1037, 1999.

[105] Mangasarian, O. and Musicant, D., "Data discrimination via nonlinear generalized support vector machines," in *Complementarity: Applications, Algorithms and Extensions* (Ferris, M., Mangasarian, O., and Pang, J.-S., eds.), (Boston, MA), pp. 233–251, Kluwer Academic Publishers, 2001.

[106] Mangasarian, O. and Musicant, D., "Lagrangian support vector machines," *Journal of Machine Learning Research*, vol. 1, pp. 161–177, 2001.

[107] Mangasarian, O., Setiono, R., and Wolberg, W., "Pattern recognition via linear programming: Theory and application to medical diagnosis," in *Large-Scale Numerical Optimization* (Coleman, T. and Li, Y., eds.), (Philadelphia, Pennsylvania), pp. 22–31, SIAM, 1990.

[108] Mangasarian, O., Street, W., and Wolberg, W., "Breast cancer diagnosis and prognosis via linear programming," *Operations Research*, vol. 43, no. 4, pp. 570–577, 1995.

[109] Markowski, E. and Markowski, C., "Some difficulties and improvements in applying linear programming formulations to the discriminant problem," *Decision Sciences*, vol. 16, pp. 237–247, 1985.

[110] Mendes, R., Kennedy, J., and Neves, J., "The fully-informed particle swarm: simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.

[111] Monteiro, S. and Kosugi, Y., "Particle swarms for feature extraction of hyperspectral data," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 7, pp. 1038–1046, 2007.

[112] Narendra, P. and Fukunaga, K., "A branch and bound algorithm for feature subset selection," *IEEE Transactions on Computers*, vol. C-26, no. 9, pp. 917–922, 1977.

[113] Nelson, A. and O'Connor, M., "Mind cognitive impairment: A neuropsychological perspective," *CNS Spectrums*, vol. 13, no. 1, pp. 56–64, 2008.

[114] Pampara, G., Franken, N., and Engelbrecht, A., "Combining particle swarm optimisation with angle modulation to solve binary problems," in *2005 IEEE Congress on Evolutionary Computation*, (Piscataway, NJ), pp. 89–96, IEEE, 2005.

[115] Park, Y., Ziegler, T., Gletsu-Miller, N., Liang, Y., Yu, T., Accardi, C., and Jones, D., "Postprandial cysteine/cystine redox potential in human plasma varies with meal content of sulfur amino acids," *The Journal of Nutrition*, vol. 140, pp. 760–765, 2010.

[116] Parker, M. and Ryan, J., "Finding the minimum weight IIS cover of an infeasible system of linear inequalities," *Annals of Mathematics and Artificial Intelligence*, vol. 17, pp. 107–126, 1996.

[117] Parsopoulos, K. and Vrahatis, M., "On the computation of all global minimizers through particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 211–224, 2004.

[118] Parsopoulos, K. and Vrahatis, M., "Parameter selection and adaptation in unified particle swarm optimization," *Mathematical and Computer Modeling*, vol. 46, no. 1/2, pp. 198–213, 2007.

[119] Paterson, P., Lyon, A., Kamencic, H., Andersen, L., and Juurlink, B., "Sulfur amino acid deficiency depresses brain glutathione concentration," *Nutritional Neuroscience*, vol. 4, pp. 213–222, 2001.

[120] Pavur, R., "Dimensionality representation of linear discriminant function space for the multiple-group problem: An MIP approach," *Annals of Operations Research*, vol. 74, pp. 37–50, 1997.

[121] Pavur, R., "A comparative study of the effect of the position of outliers on classical and nontraditional approaches to the two-group classification problem," *European Journal of Operational Research*, vol. 136, pp. 603–615, 2002.

[122] Pavur, R. and Loucopoulos, C., "Evaluating the effect of gap size in a single function mathematical programming model for the three-group classification problem," *Journal of the Operational Research Society*, vol. 52, pp. 896–904, 2001.

[123] Pavur, R., Wanarat, P., and Loucopoulos, C., "Examination of the classificatory performance of MIP models with secondary goals for the two-group discriminant problem," *Annals of Operations Research*, vol. 74, pp. 173–189, 1997.

[124] Peer, E., van den Bergh, F., and Engelbrecht, A., "Using neighbourhoods with the guaranteed convergence PSO," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, (Piscataway, NJ), pp. 235–242, IEEE, 2003.

[125] Peng, H., Long, F., and Ding, C., "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[126] Petalas, Y., Parsopoulos, K., and Vrahatis, M., "Memetic particle swarm optimization," *Annals of Operations Research*, vol. 156, no. 1, pp. 99–127, 2007.

[127] Poli, R., "Analysis of the publications on the applications of particle swarm optimisation," *Journal of Artificial Evolution and Applications*, 2008. doi:10.1155/2008/685175. 10 pages.

[128] Poli, R., Kennedy, J., and Blackwell, T., "Particle swarm optimization: An overview," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[129] Pudil, P., Novovicova, J., and Kittler, J., "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, pp. 1119–1125, 1994.

[130] Rakotomamonjy, A., "Variable selection using SVM-based criteria," *Journal of Machine Learning Research*, vol. 3, pp. 1357–1370, 2003.

[131] Ressom, H., Varghese, R., Saha, D., Orvisky, E., Goldman, L., Petricoin, E., Conrads, T., Veenstra, T., Abdel-Hamid, M., Loffredo, C., and Goldman, R., "Particle swarm optimization for analysis of mass spectral serum profiles," in *Genetic and Evolutionary Computation Conference*, (New York, NY), pp. 431–438, ACM, 2005.

[132] Ressom, H., Varghese, R., Abdel-Hamid, M., Eissa, S., Saha, D., Goldman, L., Petricoin, E., Conrads, T., Veenstra, T., Loffredo, C., and Goldman, R., "Analysis of mass spectral serum profiles for biomarker selection," *Bioinformatics*, vol. 21, pp. 4039–4045, 2005.

[133] Rubin, P., "A comparison of linear programming and parametric approaches to the two-group discriminant problem," *Decision Sciences*, vol. 21, pp. 373–386, 1990.

[134] Rubin, P., "Separation failure in linear programming discriminant models," *Decision Sciences*, vol. 22, pp. 519–535, 1991.

[135] Rubin, P., "Solving mixed integer classification problems by decomposition," *Annals of Operations Research*, vol. 74, pp. 51–64, 1997.

[136] Samanta, B. and Nataraj, C., "Application of particle swarm optimization and proximal support vector machines for fault detection," *Swarm Intelligence*, vol. 3, pp. 303–325, 2009.

[137] Shen, Q., Jiang, J.-H., Jiao, C.-X., Shen, G.-L., and Yu, R.-Q., "Modified particle swarm optimization algorithm for variable selection in MLR and PLS modeling: QSAR studies of antagonism of angiotensin II antagonists," *European Journal of Pharmaceutical Sciences*, vol. 22, pp. 145–152, 2004.

[138] Shi, Y. and Eberhart, R., "A modified particle swarm optimizer," in *1998 IEEE international Conference on Evolutionary Computation*, (Piscataway, NJ), pp. 69–73, IEEE, 1998.

[139] Siedlecki, W. and Sklansky, J., "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, vol. 10, pp. 335–347, 1989.

[140] Silva, A. D. and Stam, A., "Second order mathematical programming formulations for discriminant analysis," *European Journal of Operational Research*, vol. 72, pp. 4–22, 1994.

[141] Silva, A. D. and Stam, A., "A mixed integer programming algorithm for minimizing the training sample misclassification cost in two-group classification," *Annals of Operations Research*, vol. 74, pp. 129–157, 1997.

[142] Smith, C., "Some examples of discrimination," *Annals of Eugenics*, vol. 13, pp. 272–282, 1947.

[143] Somol, P., Pudil, P., Novovicova, J., and Paclik, P., "Adaptive floating search methods in feature selection," *Pattern Recognition Letters*, vol. 20, pp. 1157–1163, 1999.

[144] Stam, A., "Nontraditional approaches to statistical classification: Some perspectives on l_p-norm methods," *Annals of Operations Research*, vol. 74, pp. 1–36, 1997.

[145] Stam, A. and Joachimsthaler, E., "Solving the classification problem in discriminant analysis via linear and nonlinear programming methods," *Decision Sciences*, vol. 20, pp. 285–293, 1989.

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

[147] Stam, A. and Ragsdale, C., "On the classification gap in mathematical-programming-based approaches to the discriminant problem," *Naval Research Logistics*, vol. 39, pp. 545–559, 1992.

[148] Stam, A. and Ungar, D., "RAGNU: A microcomputer package for two-group mathematical programming-based nonparametric classification," *European Journal of Operational Research*, vol. 86, pp. 374–388, 1995.

[149] Stearns, S., "On selecting features for pattern classifiers," in *Proceedings of the Third International Conference on Pattern Recognition*, (Coronado, California), pp. 71–75, 1976.

[150] Stuss, D. and Trites, R., "Classification of neurological status using multiple discriminant function analysis of neuropsychological test scores," *Journal of Consulting and Clinical Psychology*, vol. 45, no. 1, pp. 145–145, 1977.

[151] Sueyoshi, T., "DEA-discriminant analysis in the view of goal programming," *European Journal of Operational Research*, vol. 115, pp. 564–582, 1999.

[152] Sueyoshi, T., "Extended DEA-discriminant analysis," *European Journal of Operational Research*, vol. 131, pp. 324–351, 2001.

[153] Tabert, M., Manly, J., Liu, X., Pelton, G., Rosenblum, S., Jacobs, M., Zamora, D., Goodkind, M., Bell, K., Stern, Y., and Devanand, D., "Neuropsychological prediction of conversion to Alzheimer disease in patients with mild cognitive impairment," *Archives of General Psychiatry*, vol. 63, pp. 916–924, 2006.

[154] Tang, E., Suganthan, P., and Yao, X., "Feature selection for microarray data using least squares SVM and particle swarm optimization," in *Proceedings of the 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, (Piscataway, NJ), IEEE, 2005.

[155] Tibshirani, R., "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[156] Tor-Agbidye, J., Palmer, V., Lasarev, M., Craig, A., Blythe, L., Sabri, M., and Spencer, P., "Bioactivation of cyanide to cyanate in sulfur amino acid deficiency: Relevance to neurological disease in humans subsisting on cassava," *Toxicological Sciences*, vol. 50, pp. 228–235, 1999.

[157] Trelea, I., "The particle swarm optimization algorithm: Convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317–325, 2003.

[158] van den Bergh, F. and Engelbrecht, A., "A study of particle swarm optimization particle trajectories," *Information Sciences*, vol. 176, pp. 937–971, 2006.

[159] Vapnik, V., *The Nature of Statistical Learning Theory*. New York: Springer Verlag, 1995.

[160] Wanarat, P. and Pavur, R., "Examining the effect of second-order terms in mathematical programming approaches to the classification problem," *European Journal of Operational Research*, vol. 93, pp. 582–601, 1996.

[161] Wang, L. and Yu, J., "Fault feature selection based on modified binary PSO with mutation and its application in chemical process fault diagnosis," in *Lecture Notes in Computer Science: Vol. 3612. First International Conference on Advances in Natural Computation*, (Berlin/Heidelberg), pp. 832–840, Springer-Verlag, 2005.

[162] Xiao, B., "Necessary and sufficient conditions of unacceptable solutions in LP discriminant analysis," *Decision Sciences*, vol. 24, pp. 699–712, 1993.

[163] Xiao, B., "Decision power and solutions of LP discriminant models: Rejoinder," *Decision Sciences*, vol. 25, pp. 335–336, 1994.

[164] Xiao, B. and Feng, Y., "Alternative discriminant vectors in LP models and a regularization method," *Annals of Operations Research*, vol. 74, pp. 113–127, 1997.

[165] Yanev, N. and Balev, S., "A combinatorial approach to the classification problem," *European Journal of Operational Research*, vol. 115, pp. 339–350, 1999.

[166] Yu, L. and Liu, H., "Efficient feature selection via analysis of relevance and redundancy," *Journal of Machine Learning Research*, vol. 5, pp. 1205–1224, 2004.

[167] Zhou, X. and Mao, K., "LS bound based gene selection for DNA microarray data," *Bioinformatics*, vol. 21, no. 8, pp. 1559–1564, 2005.