SOLVING A MIXED-INTEGER PROGRAMMING FORMULATION OF A CLASSIFICATION MODEL WITH MISCLASSIFICATION LIMITS

A Thesis
Presented to
The Academic Faculty

by

J. Paul Brooks

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Industrial and Systems Engineering
Georgia Institute of Technology
December 2005

SOLVING A MIXED-INTEGER PROGRAMMING FORMULATION OF A CLASSIFICATION MODEL WITH MISCLASSIFICATION LIMITS

Approved by:

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

Dr. George Nemhauser
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Ellis Johnson
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Mark Prausnitz
School of Chemical and Biomolecular Engineering
Georgia Institute of Technology

Dr. Brani Vidakovic
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: 23 August 2005

Dedication

To my grandfather, in whose footsteps I seem to be following, albeit backwards

Acknowledgements

There are too many people to name who have played significant roles in my graduate career, and I would likely forget some names if I tried to list them all. Surely, a Ph.D. is not an individual achievement.

I would like to acknowlege my thesis advisor, Dr. Eva Lee, for her contributions to my academic maturity that enabled me to produce these results. She continually challenged me at every step to produce better work, and assured me that it was, in fact, possible to produce better work. I am also grateful for her financial support for the past five years, without which my graduate career simply would not have been possible.

I would like to thank Dr. George Nemhauser, Dr. Ellis Johnson, and Dr. Brani Vidakovic for serving on my dissertation committee and providing valuable feedback. In particular, Dr. Vidakovic's suggestion to investigate the consistency of the DAMIP added to the completeness of this work. I would also like to thank Dr. Gary Parker for his indispensible advice throughout all of the phases of my graduate career and beyond.

The ARCS (Achievement Rewards for College Scientists) Foundation provided financial and moral support throughout my graduate career. The banquets that they hosted provided an opportunity to see the members' enthusiasm for graduate research such as mine, which encouraged me to remain dedicated.

Discussions with Vijay Bharadwaj, Sid Maheshwary, and Dieter Vandenbussche have been extremely helpful.

Thank you to my parents, who have surrounded me with love and always offered their support, while at the same time granting me the independence to chose my own path.

To my wife Courtenay, thank you for your love and support (including giving me a break on chores!). You're the one that assured me that you would still be around regardless of outcomes.

There are countless others, at Georgia Tech and otherwise, who have contributed to this

project in various ways. Thank you.

Table of Contents

List of Tables

List of Figures

Summary

Discriminant analysis is concerned with the classification of observations, represented by vectors of attribute values, as belonging to one or more groups and/or with determining the identifying attributes that best define the groups. The DAMIP (discriminant analysis using mixed-integer programming) [35] is a model based on a result by Anderson [1] wherein he derives rules that maximize the total probability of correct G-group classification, subject to limits on misclassification probabilities. The DAMIP is an empirical method for estimating the parameters of the optimal classification rule, which were identified as coefficients of linear functions by Anderson.

The DAMIP is shown to be a consistent method for estimating the parameters of the optimal solution to the problem of maximizing the probability of correct classification subject to limits on misclassification. The method is shown to be $\mathcal{NP}$-complete, and an approximation is formulated as a mixed-integer program (MIP). The MIP is difficult to solve due to the formulation of constraints wherein certain variables are equal to the maximum of a set of linear functions. These constraints are conducive to an ill-conditioned coefficient matrix. The current work investigates techniques for solving instances of the DAMIP. A polynomial-time algorithm is given for two-group instances of the DAMIP. For harder problems, the conflict graph and hypergraph are employed for finding cuts in a branch-and-bound framework. Other techniques include a heuristic for finding integer feasible solutions and a tailored branching scheme. The robustness of the MIP formulation of the DAMIP is noted and empirically tested on real-world and simulated data sets.

Chapter I

Introduction

Discriminant analysis is concerned with the classification of observations as belonging to one or more groups and/or with determining the identifying attributes that best define the groups. Classification and group description are tasks that are encountered in almost any field. Medical diagnosis, assignment to market audiences, and cell behavior analysis are a few examples of an ever-increasing number of applications (e.g., see [90]). Humans use discriminating techniques in everyday situations. The automation of this process enables the incorporation of larger amounts of data on observations with known group membership, and allows for the possible detection of subtle rules for classification.

The classification models usually take the form of easily-implementable rules for classifying objects. The derivation of the rules need not be as efficient as the process for classifying unknown entities. For example, a model with the purpose of diagnosing different classes of heart disease can be built based on archived patient data. The rules need only be determined one time. Once the model is developed, the time to classify a new patient using the rules should be short enough to allow the doctor and patient to incorporate other information and take preventive measures. The rules may be modified using additional data and methods, but this process is not as time-sensitive as the diagnosis of a patient.

For most types of classification models, the implementation of the classification rules is efficient. The development of the model is more often a difficult problem, as is the case for methods employing mixed-integer programming or neural networks. The difficulties arise from problem complexity and insufficient computing power.

The DAMIP (discriminant analysis using mixed-integer programming)[35] is a classification model based on a result by Anderson [1] wherein he determined the form of optimal classification rules that maximize the total probability of correct G-group classification, subject to misclassification probability limits. The result characterizes linear functions that

dictate how entities are classified such that the probability of correct classification is maximized, and the limits on misclassification probabilities are met. The functions are algebraic expressions of classification rules. The objective of the DAMIP [35] is to determine the coefficients in the linear functions that maximize the correct classification of training entities, subject to upper limits on misclassification.

The major contributions of the DAMIP to the field of discriminant analysis include the incorporation of a reserved judgment region, the ability to discriminate between more than two populations in a single application of the model, and a consistent method for classifying observations subject to limits on misclassification rates. The reserved judgment region is an artificial group in which entities demonstrating insufficient indication of membership to any group may be placed. The placement of an observation in the reserved judgment group is a signal to collect more identifying data. The traditional linear programming- and MIP-based classification procedures are designed to discriminate between two groups [32, 48]. Through subsequent applications of the models, rules can be developed to classify observations to any number of groups [32]. The DAMIP generates rules for multiple-group classification with the solution of a single mixed-integer program.

Finding the optimal solution for the DAMIP involves finding the maxima of several sets of linear functions, subject to coupling constraints. The mixed-integer programming formulation of these constraints make the MIP extremely ill-conditioned and therefore difficult to solve. The goals of this thesis include characterizing the DAMIP as a statistically viable method for pattern classification when misclassification limits are desired, advance computational methods for solving the difficult MIP instances arising from the DAMIP models, and testing the classification performance on of the DAMIP real-world and simulated data.

1.1 Mixed-integer programming and linear programming

Mathematical programming is concerned with the solution of optimization problems with an *objective function* formulated in terms of *decision variables*, with the possibility of additional restrictions, or *constraints*, on the values that the decision variables may attain. Decision variables assume values that represent quantifiable decisions. A math program can be

expressed in the following form

$$\text{maximize } f(x)$$

subject to

$$x \in S$$

The objective function $f(x)$ is a function of the decision variables $x = \{x_1, x_2, \ldots, x_n\}$, and ordinarily describes the profit or cost associated with decisions contained in the decision variables. The set S defines the *feasible region*, or the set of values that the decision variables can have. The x variables may be restricted to assume values from a countable set of values, in which case they are *discrete*; or, they may be allowed to assume values from intervals of real numbers, in which case they are *continuous*, or a mixture of both.

In the special case that the discrete variables are required to assume integral values, the objective function is a linear function of the decision variables, and the set S is defined by linear functions of the decision variables, the math program is referred to as a *(linear) mixed-integer program*. The reader is referred to the books by Nemhauser and Wolsey [68] and Wolsey [87] for background on formulating and solving mixed-integer programs. A mixed-integer programming problem can be expressed as

$$\text{maximize } cx + dy$$

subject to

$$\begin{aligned} Ax + By &\leq b \\ x &\in \mathbb{R}^n_+ \\ y &\in \mathbb{Z}^p_+ \end{aligned}$$

where x and y are vectors of decision variables, maximize $cx + dy$ is the *objective function*, and $Ax + By \leq b$ are the constraints. We will assume, without loss of generality, that all variables are required to be non-negative. The variables y are *integer variables* and the variables x are continuous variables. In the program above, the variables are restricted to assume only nonnegative values. This practice does not reflect a loss in generality, and the restriction could have been represented by the linear constraints $x \geq 0$ and $y \geq 0$.

A special case of mixed-integer programming of interest is the case when the integer variables are further restricted to take values 0 or 1. In this case, the variables are *binary variables* and the problem is a *binary mixed-integer program*.

Binary variables are particularly useful in that they can be used to express logical requirements as constraints. For example, if x_i and x_j are binary variables, the constraint $x_i \leq x_j$ requires $x_j = 1$ whenever $x_i = 1$.

1.1.1 Feasibility and optimality

A set of values for the variables of a mixed-integer program is a *feasible point*, or lies in the *feasible region*, if all of the constraints are satisfied. A set of values is *optimal* if, among all feasible points, no other point has a better objective function value. An optimal solution is not guaranteed to exist. If the feasible region is empty, then the MIP is *infeasible*. If for every feasible point there is another feasible point with a better objective function value, then the MIP is *unbounded*.

1.1.2 Polyhedra and facets

The constraints of a mixed-integer program form a *polyhedron*. A *polyhedron* is a set of points satisfying a finite number of linear inequalities. The *dimension* of a polyhedron is the maximum number of linearly independent vectors contained in the polyhedron.

A *valid inequality* for a polyhedron is an inequality satisfied by every point in the polyhedron. The *face* corresponding to the valid inequality is the set of points in the polyhedron satisfying the valid inequality at equality. A *facet* is a face with dimension one less than the dimension one less than the polyhedron.

1.1.3 Linear programming and linear programming duality

In the special case that the set of integer variables is empty, the mixed-integer program is called a *linear program* (LP). An LP can be expressed as

$$\text{maximize } cx \quad (P)$$

subject to

$$\begin{aligned} Ax &\leq b \\ x &\in \mathbb{R}^n_+ \end{aligned}$$

The *dual* of a linear program is also a linear program. The dual of linear program (P) is

$$\text{minimize } ub \quad (D)$$

subject to

$$\begin{aligned} uA &\geq c \\ u &\in \mathbb{R}^n_+ \end{aligned}$$

The linear program (P) is called the *primal* and is the dual of the dual. Primal and dual linear programs have a special relationship. If the objective value of the primal linear program is bounded, then the dual linear program has the same optimal objective function value. If the objective value of the primal is unbounded or infeasible, then the dual linear program is infeasible. If the objective value of the dual is unbounded or infeasible, then the primal linear program is infeasible.

1.1.4 Branch-and-bound algorithms

Branch-and-bound is a common method for solving mixed-integer programs. A branch-and-bound method solves a sequence of relaxations of an optimization problem, updating the upper and lower bounds on the objective function value along the way. The bounds help to reduce the number of relaxations that must be considered. After a relaxation is solved, the solution space is *branched* upon, producing new relaxations that each contain a different partition of the solution space. For an introduction to branch-and-bound in general, see Chapter 18 of [74].

Branch-and-bound for a mixed-integer program begins by solving the *linear programming relaxation* of an integer program by allowing the integer variables to assume continuous values. If the solution has all integer variables having integral values, then the solution is optimal. Otherwise, a sequence of linear programs is solved. Integer feasible solutions

to these linear programs provide lower bounds for the optimal objective value. Typically, branches are created by selecting an integer variable with fractional value and branching on that variable. If $\lfloor x_i \rfloor < x_i < \lceil x_i \rceil$, then 2 branches can be created, one with the previous linear program plus the constraint $x_i \leq \lfloor x_i \rfloor$ and one with the previous linear program plus the constraint $x_i \geq \lceil x_i \rceil$. This practice creates a hierarchy of linear programs. The linear programs are often referred to as *nodes* of the *branch-and-bound tree*, where the LP relaxation is the *root node*. The performance of a branch-and-bound algorithm for mixed-integer programming is often measured by the number of nodes visited, or linear programs solved, before the optimal solution is found.

Branch-and-bound is essentially an intelligent enumeration scheme that seeks to avoid enumeration of all feasible solutions by using information gathered from the LP subproblems. If the objective function value of a linear programming subproblem is less than that of the best known integer feasible solution, then that node is *fathomed*, meaning that this LP does not need to be branched upon. The objective value of a feasible solution will never exceed the *incumbent* objective value because successive LP's will optimize over a smaller feasible region. An LP subproblem is also fathomed when it is infeasible.

Branching is the process by which the feasible region of the LP relaxation is further partitioned. Branching on a variable refers to selection of a fractional-valued integer variable and creating two linear programs with additional restrictions as described. The selection of the branching variable can be based on various criteria, including the most fractional or least fractional variable. *Strong branching* refers to branching based on information about how candidate variables will affect the objective function in subsequent linear program subproblems. Strong branching was first developed for solving traveling salesman problems [2].

More general branching schemes can be employed. For example, if a set of binary integer variables S has $\sum_{i \in S} x_i > |S| - 1$, then two linear programming subproblems can be created; one with the constraint $\sum_{i \in S} x_i < |S| - 1$ added and the other with $\sum_{i \in S} x_i = |S|$ added. Other generalizations include creating more or less than two new subproblems.

Node selection refers to the process of selecting an unsolved LP subproblem to solve.

Methods for node selection include selecting the LP subproblem whose *parent node* has the best objective value and selecting the LP subproblem with the best estimate of the objective value when integer infeasibilities are removed.

1.1.5 Branch-and-cut algorithms

Branch-and-cut for mixed-integer programs is a generalization of branch-and-bound. Branch-and-cut algorithms allow for the addition of valid inequalities for the mixed-integer program that are violated by the optimal solution of an LP subproblem. The *separation problem* is the problem of finding a violated valid inequality. These *cuts* are added to the LP subproblem, the subproblem is re-solved, and the process is either repeated or the problem is branched upon. Variations of branch-and-cut include using different methods for finding violated inequalities and varying the number of rounds of cut generation.

1.2 Graphs, hypergraphs, and mixed-integer programming

The reader is referred to Chapter 1 of *Graph Theory* by Bollobàs [13] for the fundamentals of graph theory and to *Graphs and Hypergraphs* by Berge [10] for background on hypergraphs. A *graph* $G = (V, E)$ consists of a set of *nodes*, or *vertices*, V and a set of *edges* E. Edges are unordered pairs of elements of V. A node v is *adjacent* to another node w if the edge $(v, w) \in E$. Graphs can be represented pictorially with the nodes drawn as dots and the edges drawn as lines between the dots for adjacent nodes.

A *clique* K is a set of nodes such that there exists an edge between every pair of nodes. A pictorial representation of a clique is given in Figure 1.

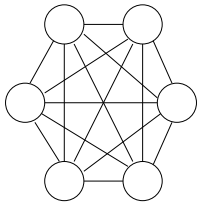


Figure 1: An example of a clique.

A *cycle* is an alternating sequence of nodes and edges beginning and ending with the same node, and with distinct nodes and edges otherwise. The *size* of a cycle is the number

of distinct nodes that it contains. An *odd cycle* is a cycle with an odd size. A *chord* of a cycle is an edge not in the cycle that contains 2 nodes in the cycle. An *odd hole* is a chordless odd cycle. A visual representation of an odd hole of size 5 is shown in Figure 2.

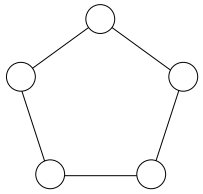


Figure 2: An example of an odd hole.

The *chromatic number* of a graph is the minimum number of colors needed to assign a color to each node of the graph such that no two adjacent nodes have the same color. An *induced subgraph* of a graph is a subset of the nodes of the graph and all of the edges between those nodes in the graph. A graph is *perfect* if every induced subgraph has the property that the chromatic number is equal to the size of a maximal clique.

A *hypergraph* $G = (V, E)$ is a set of nodes V and a set of *hyperedges* E. Hyperedges will hereafter be referred to as edges. Edges in a hypergraph are elements of the power set of V. A *uniform n-hypergraph* is a hypergraph with all edges having the same cardinality. Therefore, a graph is a uniform 2-hypergraph. We will only consider uniform n-hypergraphs, and we will refer to them as "n-hypergraphs", or simply "hypergraphs".

We extend the notion of adjacency to hypergraphs as follows: A node v is *adjacent* to a set of nodes C if either $C \cup v$ is contained in an edge of the hypergraph, or every subset of n nodes containing v and nodes from C is an edge of the hypergraph. A *hyperclique* K of size m in a n-hypergraph where $m \geq n$ is a set of nodes such that all $\binom{m}{n}$ edges are present in the hypergraph.

1.2.1 Conflict graphs and cuts

A conflict graph is a tool for representing relationships between pairs of binary variables in a math program. For each binary variable, there is an associated node in the conflict graph. For every pair of variables x_i and x_j for which $x_i + x_j \leq 1$ is valid for the set of feasible solutions, there exists an edge between the associated nodes i and j in the conflict graph.

Optimizing over the inequalities of the conflict graph is equivalent to solving a *node packing problem*. For a graph $G = (V, E)$, a node packing is a set of nodes such that no two are adjacent (see [68]). For a mixed-integer program, the polytope defined by the relationships in the conflict graph contains the projection of the feasible region onto the space of binary integer variables. Therefore, valid inequalities for the node packing polytope are valid for the polytope defined by the conflict graph, and are valid for the associated mixed-integer program. These valid inequalities can be used in a branch-and-cut framework.

Two well-known classes of valid inequalities for the node packing polytope are *maximal clique inequalities* and *odd hole inequalities*. For a maximal clique K in a graph, the following inequality is valid and facet-defining for the node packing polytope [33, 73].

$$\sum_{i \in K} x_i \leq 1$$

If the graph of a node packing problem is perfect, then the maximal clique inequalities is the set of all facets of the node packing polytope [62].

If H is the set of nodes of an odd hole of a graph, then

$$\sum_{i \in H} x_i \leq \frac{|H| - 1}{2}$$

is a valid inequality for the node packing polytope [73].

Further early work on set packing polyhedra, of which node packing is a special case, can be found in [72]. Conflict graphs and their use in solving combinatorial optimization problems are described in [40, 54, 12, 14, 3, 5, 15, 43, 50, 67].

1.2.2 Hypergraphs, the independent set polytope, and cuts

The notion of a conflict graph can easily be extended to a *conflict hypergraph* where the nodes again correspond to binary variables in a math program. An edge in the conflict hypergraph consisting of a set of nodes C corresponds to the following relationship

$$\sum_{i \in C} x_i \leq |C| - 1$$

These types of inequalities are known as *independent set inequalities*. Facets of the independent set polytope were explored in [58], including the following extension of maximal clique inequalities. Suppose that K is a set of nodes such that all $\binom{|K|}{n}$ edges are present in the n-hypergraph. Then the following inequality is valid for the associated independent set polytope

$$\sum_{i \in K} x_i \leq |K| - 1$$

If none of the $\binom{|K|}{n}$ edges is implied by the $2, 3, \ldots, n-1$ hypergraphs, then the inequality is facet-defining [58].

Further work on conflict hypergraphs and their use in solving combinatorial optimization problems is in [78, 29, 66, 53, 55].

1.3 Algorithm complexity and $\mathcal{NP}$-completeness

An *algorithm* is a step-by-step procedure that accepts input and computes an appropriate output. An algorithm's *c*omplexity is commonly measured by the number of elementary operations required in the limit as the size of the input increases for a worst-case instance. The asymptotic running time of an algorithm is represented as a function of the size of its input. The reader is referred to Chapter 2 of [24] for an introduction to asymptotic notation. Suppose n is the size of the input for an algorithm, and $f(n)$ is the running time. An algorithm has complexity $O(g(n))$ if there exists constants c and n_0 such that $f(n) \leq cg(n)$ for all $n \geq n_0$.

We will be concerned with the complexity of algorithms that seek to determine if solutions satisfying certain criteria exist, or *feasibility problems*. Every optimization problem can be stated as a feasibility problem. Let $\mathcal{P}$ be the set of all problems such that are solvable by an algorithm in *polynomial time*, or $O(g(n))$ where $g(n)$ is a polynomial. Let $\mathcal{NP}$ be the set of all problems that are verifiable by an algorithm in polynomial time. In other words, if a potential solution is given, then determining the feasibility of the solution can be performed in polynomial time.

A problem P_1 is *reducible* in polynomial-time to another problem P_2 if there exists a polynomial-time transformation of an instance of P_1 to an instance of P_2. Therefore, if there exists no polynomial-time algorithm for P_1, then there exists no polynomial-time algorithm for P_2. Equivalently, if there is a polynomial-time algorithm for P_2, then there exists a polynomial time algorithm for P_1 because reducing the instance of P_1 to an instance of P_2 and then running the algorithm for P_2 together takes polynomial time.

A problem is $\mathcal{NP}$*-hard* if every problem in $\mathcal{NP}$ can be reduced to it. Intuitively, an $\mathcal{NP}$-hard problem is at least as "hard" as every other problem in $\mathcal{NP}$. A problem is $\mathcal{NP}$*-complete* if it is in $\mathcal{NP}$ and is $\mathcal{NP}$-hard. Therefore, $\mathcal{NP}$-complete problems are

the "hardest" problems in $\mathcal{NP}$. There are no known polynomial-time algorithms for $\mathcal{NP}$-complete problems. Should one exist, then every $\mathcal{NP}$-complete problem could be solved in polynomial time. An extensive list of $\mathcal{NP}$-complete problems and reductions is contained in [36]. The reader is also referred to [24] and [68] for a further introduction to complexity.

1.4 Pattern recognition, discriminant analysis, and statistical pattern classification

Cognitive science is the science of learning, knowing, and reasoning. *Pattern recognition* is a broad field within *cognitive science* that is concerned with the process of recognizing, identifying, and categorizing input information. These areas intersect with computer science, particularly in the closely related areas of *artificial intelligence*, *machine learning*, and *statistical pattern recognition*. Artificial intelligence is associated with constructing machines and systems that reflect human abilities in cognition. Machine learning refers to how these machines and systems replicate the learning process, which is often achieved by seeking and discovering patterns in data, or statistical pattern recognition.

Discriminant analysis is the process of discriminating between categories or populations. Associated with discriminant analysis as a statistical tool are the tasks of determining the features that best discriminate between populations and the process of classifying new objects based on these features. The former is often called *feature selection* and the latter is referred to as *statistical pattern classification*. This work will be largely concerned with the development of a viable statistical pattern classifier.

As with many fields, the recent advances in computing power have led to a sharp increase in the interest and application of discriminant analysis techniques. The reader is referred to Duda et al. [27] for an introduction to various techniques for pattern classification, and to Zopounidis et al. [90] for examples of applications of pattern classification.

1.4.1 Supervised learning, training, and cross validation

An *entity* or *observation* is essentially a data point as commonly understood in statistics. In the framework of statistical pattern classification, an entity is a set of quantitative measurements (or qualitative measurements expressed quantitatively) of *attributes* for a particular

object. As an example, in medical diagnosis an entity could be the various blood chemistry levels of a patient. With each entity is associated one or more *groups* (or *populations*, *classes*, *categories*) to which it belongs. To continue with the example, the groups could be the various classes of heart disease. Statistical classification seeks to determine rules for associating entities with the groups to which they belong. Ideally, these associations align with the associations that human reasoning would produce based on information gathered on objects and their apparent categories.

Supervised learning is the process of developing classification rules based on entities for which the classification is already known. Note that the process implies that the populations are already well-defined. *Unsupervised learning* is the process of discovering patterns from unlabeled entities and thereby discovering and describing the underlying populations. Models derived using supervised learning can be used for both functions of discriminant analysis - feature selection and classification. The model that we consider is a method for supervised learning, so we assume that populations are previously defined.

The set of entities with known classification that is used to develop classification rules is the *training set*. The training set may be partitioned so that some entities are withheld during the model-developing process, also known as the *training* of the model. The withheld entities are a *test set* that is used to determine the validity of the model, a process known as *cross validation*. Entities from the test set are subjected to the rules of classification to measure the performance of the rules on entities with unknown group membership.

Validation of classification models is often performed using m-fold cross validation where the data with known classification is partitioned into m *folds* of approximately equal size. The classification model is trained m times, with the m^{th} fold withheld during each run for testing. The performance of the model is evaluated by the classification accuracy on the m test folds, and can be represented using a *classification matrix* or *confusion matrix*.

The classification matrix is a square matrix with the number of rows and columns equal to the number of groups. The ij^{th} entry of the classification matrix contains the number or proportion of test entities from group i that were classified by the model as belonging to group j. Therefore, the number or proportion of correctly classified entities are contained

in the diagonal elements of the classification matrix, and the number or proportion of misclassified entities are in the off-diagonal entries.

1.4.2 Bayesian inference and classification

The popularity of *Bayesian inference* has risen drastically over the past several decades, perhaps in part due to its suitability for statistical learning. The reader is referred to O'Hagan's volume [71] for a thorough treatment of Bayesian inference. Bayesian inference is usually contrasted against *classical inference*, though in practice they often imply the same methodology.

The Bayesian method relies on a *subjective* view of probability, as opposed to the *frequentist* view upon which classical inference is based [71]. A subjective probability describes a degree of belief in a *proposition* held by the investigator based on some information. A frequency probability describes the likelihood of an *event* given an infinite number of trials.

In Bayesian statistics, inferences are based on the *posterior distribution*. The posterior distribution is the product of the *prior probability* and the *likelihood function*. The prior probability distribution represents the initial degree of belief in a proposition, often before empirical data is considered. The likelihood function describes the likelihood that the behavior is exhibited, given that the proposition is true. The posterior distribution describes the likelihood that the proposition is true, given the observed behavior.

Suppose we have a proposition or random variable θ about which we would like to make inferences, and data x. Application of Bayes' Theorem gives

$$dF(\theta|x) = \frac{dF(\theta)dF(x|\theta)}{dF(x)}$$

For ease of conceptualization, assuming that every distribution has a density function, the identity can be rewritten as

$$f(\theta|x) = \frac{f(\theta)f(x|\theta)}{f(x)}$$

For classification, a prior probability function $\pi(g)$ describes the likelihood that an entity is allocated to group g regardless of its exhibited feature values x. A group density function $f(x|g)$ describes the likelihood that an entity exhibits certain measurable attribute values,

given that it belongs to population g. The posterior distribution for a group $P(g|x)$ is given by the product of the prior probability and group density function, normalized over the groups to obtain a unit probability over all groups. The observation x is allocated to the group $h = \arg\max_{g \in \mathcal{G}} P(g|x) = \arg\max_{g \in \mathcal{G}} \frac{\pi(g) f(x|g)}{\sum_{j \in \mathcal{G}} \pi(j) f(x|j)}$.

1.4.3 Discriminant functions

Most classification methods can be described in terms of *discriminant functions*. A discriminant function takes as input an observation and returns information about the classification of the observation. For data from a set of groups $\mathcal{G}$, an observation x is assigned to group h if $h = \arg\max_{g \in \mathcal{G}} l_g(x)$ where the functions l_g are the discriminant functions. Classification methods restrict the form of the discriminant functions, and training data is used to determine the values of parameters that define the functions.

The optimal classifier in the Bayesian framework can be described in terms of discriminant functions. Let $\pi_g = \pi(g)$ be the prior probability that an observation is allocated to group g and let $f_g(x) = f(x|g)$ be the likelihood that data x is drawn from population g. If we wish to minimize the probability of misclassification given x, then the optimal allocation for an entity is to the group $h = \arg\max_{g \in \mathcal{G}} P(g|x) = \arg\max_{g \in \mathcal{G}} \frac{\pi_g f_g(x)}{\sum_{j \in \mathcal{G}} \pi_j f_j(x)}$. Under the Bayesian framework,

$$P(g|x) = \frac{\pi_g f(x|g)}{f(x)} = \frac{\pi_g f(x|g)}{\sum_{j \in \mathcal{G}} \pi_j f(x|j)}$$

The discriminant functions can be $l_g(x) = P(g|x)$ for $g \in \mathcal{G}$. The same classification rule is given by $l_g(x) = \pi_g f(x|g)$ and $l_g(x) = \log f(x|g) + \log \pi_g$. The problem then becomes finding the form of the prior functions and likelihood functions that match the data.

If the data are multivariate normal with equal covariance matrices ($f(x|g) \sim N(\mu_g, \Sigma)$), then a linear discriminant function is optimal:

$$\begin{aligned} l_g(x) &= \log f(x|g) + \log \pi_g \\ &= -1/2(x - \mu_g)^T \Sigma^{-1} (x - \mu_g) - 1/2 \log |\Sigma_g| - d/2 \log 2\pi + \log \pi_g \\ &= w_g^T x + w_{g0} \end{aligned}$$

where d is the number of attributes, $w_g = \Sigma^{-1}\mu_g$, and $w_{g0} = -1/2\mu_g^T\Sigma^{-1}\mu_g + \log \pi_g + x^T\Sigma^{-1}x - d/2 \log 2\pi$. Note that the last two terms of w_{g0} are constant for all g and need not be calculated. When there are 2 groups ($\mathcal{G} = \{1, 2\}$) and the priors are equal ($\pi_1 = \pi_2$), the discriminant rule is equivalent to Fisher's linear discriminant [31]. Fisher's linear discriminant can also be derived, as it was by Fisher, by choosing w so that $\frac{(w^T\mu_1 - w^T\mu_2)^2}{w^T\Sigma w}$ is maximized.

If the data are multivariate normal with unequal covariance matrices ($f(x|g) \sim N(\mu_g, \Sigma_g)$), then a quadratic discriminant function is optimal:

$$\begin{aligned} l_g(x) &= \log f(x|g) + \log \pi_g \\ &= -1/2(x - \mu_g)^T\Sigma_g^{-1}(x - \mu_i) - 1/2 \log |\Sigma_g| - d/2 \log 2\pi + \log \pi_g \\ &= x^T W_g x + w_g^T x + w_{g0} \end{aligned}$$

where $W_g = -1/2\Sigma_g^{-1}$, $w_g = \Sigma_g^{-1}\mu_g$, and $w_{g0} = -1/2\mu_g^T\Sigma_g^{-1}\mu_g - 1/2 \log |\Sigma_g| + \log \pi_g - d/2 \log 2\pi$.

These linear and quadratic discriminant functions are often applied to data sets that are not multivariate normal or continuous (see [77], pages 234-235) by using approximations for the means and covariances. Regardless, these models are *parametric* in that they incorporate assumptions about the distribution of the data. Fisher's linear discriminant is *non-parametric* because no assumptions were made about the underlying distribution of the data. Thus, for a special case, a parametric and non-parametric model coincide to produce the same discriminant rule. The linear discriminant function derived above is also called the *homoscedastic model*, and the quadratic discriminant function is called the *heteroscedastic model*. The exact form of discriminant functions in the Bayesian framework can be derived for other distributions [27].

Some classification methods are alternate methods for finding coefficients for linear discriminant functions. In other words, they seek coefficients w_g and constant w_{g0} such that $l_g(x) = w_g x + w_{g0}$ is an optimal set of discriminant functions. The criteria for optimality is different for different methods. Linear discriminant functions project the data onto a linear subspace and then discriminate between entities in that subspace. For example, Fisher's linear discriminant projects two-group data on an optimal line, and discriminates on that

line. A good linear subspace may not exist for data with overlapping distributions between groups and therefore the data will not be classified accurately using these methods. The hyperplanes defined by the discriminant functions form boundaries between the group regions. A large portion of the literature concerning the use of math programming models for classification describe methods for finding coefficients of linear discriminant functions [90].

Other classification methods seek to determine parameters to establish quadratic discriminant functions. The general form of a quadratic discriminant function is $l_g(x) = x^T W_g x + w_g^T x + w_{g0}$. The boundaries defining the group regions can assume any hyperquadric form, as can the Bayes decision rules for arbitrary multivariate normal distributions [27].

1.4.4 Math programming methods

Math programming methods for statistical pattern classification emerged in the 1960's and gained popularity in the 1980's which has grown drastically since. Most of the math programming approaches are non-parametric which has been cited as an advantage when analyzing contaminated data sets over methods that require assumptions about the distribution of the data [80]. Most of the literature about math programming methods is concerned with either using math programming to determine the coefficients of linear discriminant functions or with *support vector machines*.

The use of linear programs to determine the coefficients of linear discriminant functions has been widely studied [32, 63, 48, 39]. The methods determine the coefficients for different objectives, including minimizing the sum of the distances to the separating hyperplane, minimizing the maximum distance of an observation to the hyperplane, and minimizing other measures of badness of fit or maximizing measures of goodness of fit.

Other math programming approaches determine the coefficients to linear discriminant functions, but using mixed-integer programs [51, 6, 22, 89]. The flexibility of mixed-integer programming allows for the minimization of misclassified observations or minimization of misclassification costs. Others [81, 80] have considered nonlinear objectives.

Support vector machines were introduced (by name) in the early 90's and have gained in

popularity since. The reader is referred to [82] for an extensive treatment of support vector machines. Consider the two-group discrimination problem. For observations x^j, nonlinear functions $\phi(\cdot)$ are chosen that map the data to a higher dimension such that the observations are (more) separable by a linear hyperplane in the higher-dimensional space. There always exists a higher dimensional space where the transformed observations are linearly separable [82]. A hyperplane is sought that minimizes training error and maximizes the distance between the groups. The hyperplane is often obtained by solving a quadratic program of the following form

$$\min \frac{1}{2} w^T w + C \sum_j F(\xi_j)$$

subject to

$$\begin{aligned} y_j(w \cdot \phi(x^j) + b) &\geq 1 - \xi_j \quad \forall\, j \\ \xi^j &\geq 0 \qquad\quad \forall\, j \end{aligned}$$

where w and b are variables that define the hyperplane, $y_j \in \{-1, 1\}$ corresponds to the classification of entity x^j, C is a penalty parameter, and F is a monotonic convex function [25]. The nonlinear mappings $\phi(\cdot)$ are called *kernel functions* and are usually decided upon before training. The first term in the objective seeks to maximize the distance or margin between the groups and the second term seeks to reduce the training error.

In practice, the dual problem is solved which requires a small portion of the data, or the *support vectors*. The support vectors represent the hardest data to classify, and define the optimal hyperplane. The use of support vectors in finding optimal hyperplanes for two-group separable problems is based on the work of Vapnik and Lerner [86], Vapnik and Chervonenkis [85], and Mangasarian [63, 64].

Many math programming methods are focused on two-group analysis only [80, 90], and performance is often compared to Fisher's linear discriminant, or Smith's quadratic discriminant [79]. It has been noted that these methods can be used for multiple group analysis by finding $G(G-1)/2$ discriminants for each pair of groups ("one-against-one") or by finding G discriminants for each group versus the remaining data ("one-against-all"), but these approaches can lead to ambiguous classification rules (see [27], page 218).

Math programming methods developed for multiple group analysis are in [61, 37, 39, 35,

58]. The latter two of these are variations on the DAMIP. The methods in [61, 37, 35] are mixed-integer programming approaches and the methods in [39, 58] use linear programming. Multiple group formulations for support vector machines have been proposed and tested [82, 59], but are still considered computationally intensive [44]. The "one-against-one" and "one-against-all" methods with support vector machines have been successfully applied [65, 44].

1.4.5 Other methods

While most classification methods can be described in terms of discriminant functions, some methods are not trained in the paradigm of determining coefficients or parameters for functions of a pre-defined form. These methods include *classification and regression trees (CART)*, *nearest-neighbor* methods, and *neural networks*.

Classification and regression trees [17] are nonparametric approaches to prediction. Classification trees seek to develop classification rules based on successive binary partitions of observations based on attribute values. Regression trees also employ rules consisting of binary partitions, but are used to predict continuous responses.

The rules generated by classification trees are easily viewable by plotting them in a tree-like structure from which the name arises. An example is shown in Figure 3. A test entity may be classified using rules in a tree plot by first comparing the entity's data with the root node of the tree. If the root node condition is satisfied by the data for a particular entity, the left branch is followed to another node; otherwise, the right branch is followed to another node. The data from the observation is compared to conditions at subsequent nodes until a leaf node is reached.

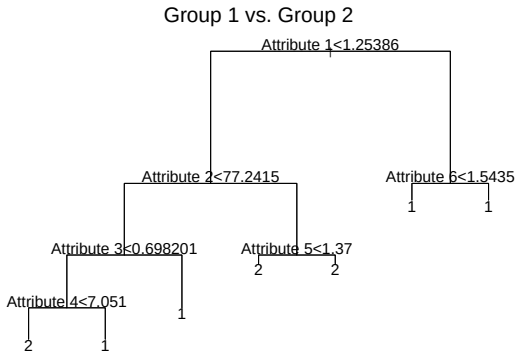


Figure 3: An example of a classification tree for a classification problem with 2 groups.

Nearest-neighbor methods begin by establishing a set of labeled prototype observations. The nearest-neighbor classification rule assigns test entities to groups according to the group membership of the nearest prototype. Different measures of distance may be used. The k-nearest-neighbor rule assigns entities to groups according to the group membership of the k nearest prototypes.

Neural networks are classification models that can also be interpreted in terms of discriminant functions, though they are used in a way that does not require finding an analytic form for the functions [27]. Neural networks are trained by considering one observation at a time, modifying the classification procedure slightly with each iteration.

1.4.6 Constrained discrimination rules

Constrained or *partial discrimination rules* are classification rules that do not necessarily force the allocation of every test observation. In general, a *reserved judgment region* is introduced for observations that do not seem to belong to any of the existing groups. Allocation to the reserved judgment region is a signal to collect more information about an entity. The region is well suited for classification problems for which some attributes are costly to measure.

Rao [76] was the first to consider constrained discrimination rules. Anderson [1] determined the form of constrained discrimination rules between k populations that maximize the total probability of correct classification. These rules are further discussed in the next section. The DAMIP is based on Anderson's result. Other methods that have included a reserved judgment region are described in [34, 23, 41, 20, 38, 42, 70, 75].

1.5 The DAMIP model

The DAMIP is an empirical model based on a result by Anderson [1]. Several MIP formulations of the DAMIP were proposed by Gallagher et al. [35]. The computational difficulty led to the development of a quicker linear approximation [58].

1.5.1 Classification to G populations

The Bayesian method of classification assigns entities to groups according to the largest posterior probability. In practice, this method can result in large misclassification probabilities. Anderson [1] showed that with the introduction of a reserved judgment region (an artificial group) and with the use of modified posterior "probabilities", the probability of correct classification can be maximized with constraints on the misclassification probabilities. He also showed that this maximum occurs for deterministic classification rules. The modified posterior probabilities are not actually probabilities, but the terminology is used because they are an indication of the likelihood of events; namely, the classification of entities to groups.

Neyman and Pearson [69] produced results concerning the maximization of the probability of correct allocation to one population subject to pre-specified misclassification probabilities for G other populations. Anderson [1] generalized the result and solved the problem of maximizing the probability of correct classification between G populations, subject to upper bounds on misclassification probabilities.

Suppose a sample point $x \in \mathbb{R}^m$ is given so that x is a vector containing m measurable attributes for an entity and must be allocated to one of G groups in $\mathcal{G}$. In Bayesian analysis, the probability that an entity belongs to a group is initially assumed to follow a prior distribution $\pi = \{\pi_g : g = 1, 2, \ldots, G\}$. The prior distribution reflects the confidence that the investigator has in the proposition that an observation belongs to each group, prior to further analysis of the data. Data is collected to derive conditional probability densities for each group. The conditional densities $\{f_g(\cdot) : g = 1, 2, \ldots, G\}$ describe the likelihood that an entity would exhibit some range of values for the measurable attributes, given that the entity belongs to population g. For an observation x and group g, $f_g(x) = f(x|g)$. The posterior probability that an observation x belongs to group g is

$$P(g|x) = \frac{\pi_g f_g(x)}{\sum_{i=1}^{G} \pi_i f_i(x)}$$

Let $\{\phi_g(x) : g = 1, 2, \ldots, G\}$ be the critical functions so that $\phi_g(x)$ is the probability

that an entity displaying attribute values x will be allocated to group g. To minimize the probability of error, observations should be allocated to the groups for which they have the largest posterior probability. In other words, $\phi_h(x) = 1$ where

$$h = \arg\max_g \frac{\pi_g f_g(x)}{\sum_{i=1}^{G} \pi_i f_i(x)} = \arg\max_g \pi_g f_g(x)$$

This rule is the *Bayes decision rule* for minimizing the probability of error (see [27], pages 22-25).

Rao [76] and later Anderson [1] considered the problem of maximizing U, but with limits on the probability of classifying an entity belonging to group h to a different group g. Anderson showed that an allocation rule maximizing U is

$$\phi_h(x) = \begin{cases} 1 & \text{if } h = \arg\max_g L_g(x) \\ 0 & \text{o.w} \end{cases}$$

where $L_g(x)$ is the modified posterior probability for group g, defined as

$$\begin{aligned} L_g(x) &= \pi_g f_g(x) - \sum_{i \in \mathcal{G} \setminus g}^{G} \lambda_{ig} f_i(x) \\ L_0(x) &= 0 \end{aligned}$$

Anderson introduces group 0, the suspended judgment region, to place entities for which there is not sufficient information for classification. Placement in the reserved judgment region corresponds to attributes that do not fit any population, and is a signal to collect more information. The reserved judgment region is necessary for helping to reduce the misclassification probabilities; non-classification is preferred over misclassification.

Note that the modified posterior probabilities may assume negative values for certain values of the λ_{ih} variables. Modified posterior probabilities are not probabilities, but the terminology is used because they are an indication of the likelihood of events.

Note that the modified posterior probabilities are discriminant functions and can be written as $l_g(x) = L_g(f(x))$ for $g \in \mathcal{G}$ where $f(x) = (f_1(x), f_2(x), \ldots, f_G(x))$ and $l_0(x) = L_0(x) = 0$. In practice, the prior probabilities and likelihood function values are determined

and then the λ_{ih} coefficients are determined in a two step process. Any method for determining the likelihood function values, including any method using discriminant functions, can be used. An empirical model to determine the optimal rules can be seen as a transformation of the data to a G-dimensional space, perhaps via discriminant functions, and a subsequent linear transformation.

1.5.2 Empirical models

Gallagher et al. [35] were the first to provide a computational framework for applying Anderson's results to discriminant analysis. They formulated three DAMIP models for calculating the λ_{ih}'s. The conditional group densities and prior probabilities are calculated based on input data. The input to the DAMIP includes approximations of prior probabilities and approximate evaluations of the group density functions evaluated at each training point. The DAMIP finds the optimal linear coefficients to derive modified posterior probabilities for optimal allocation subject to the desired limits on misclassification probabilities.

The model uses the proportion of correctly classified training entities as an approximation of U, and the proportion of misclassified training entities as estimates for the probability of misclassification. The λ_{ih}'s are calculated using a training set of data and are then used to classify entities of unknown group membership. The training data is a set of points $\{x^{gj} : g \in \mathcal{G},\ j \in \mathcal{N}_g\}$, where $\mathcal{G} = \{1, 2, \ldots G\}$ and $\mathcal{N}_g = \{1, 2, \ldots, n_g\}$, representing the values of measurable attributes for entity j of group g. The model takes estimates of the prior probabilities $\hat{\pi}_h$, estimates of the conditional group densities evaluated at the points representing each training entity $\hat{f}_h(x^{gj})$, and limits on the proportion of misclassified entities for each combination of groups $\alpha_{hg} \in [0, 1]$ as input. Model 1 is a nonlinear mixed-integer program of the following form

$$\max \sum_{g \in \mathcal{G}} \sum_{j \in \mathcal{N}_g} u_{ggj}$$

subject to

$$\begin{aligned}
L_{hgj} &= \hat{\pi}_h \hat{f}_h(x^{gj}) - \sum_{i \in \mathcal{G} \setminus h} \lambda_{ih} \hat{f}_i(x^{gj}) && h,\ g \in \mathcal{G},\ j \in \mathcal{N}_g \\
y_{gj} &= \max\{0, L_{hgj} : h = 1, \ldots, \mathcal{G}\} && g \in \mathcal{G},\ j \in \mathcal{N}_g \\
y_{gj} - L_{ggj} &\leq M(1 - u_{ggj}) && g \in \mathcal{G},\ j \in \mathcal{N}_g \\
y_{gj} - L_{hgj} &\geq \epsilon(1 - u_{hgj}) && h,\ g \in \mathcal{G},\ j \in \mathcal{N}_g,\ h \neq g \\
\sum_{j \in \mathcal{N}_g} u_{hgj} &\leq \lfloor \alpha_{hg} n_g \rfloor && h,\ g \in \mathcal{G},\ h \neq g
\end{aligned}$$

$$-\infty < L_{hgj} < \infty,\ y_{gj} \geq 0,\ \lambda_{ih} \geq 0,\ u_{hgj} \in \{0, 1\}$$

where

$$u_{hgj} = \begin{cases} 1 & \text{if entity } gj \text{ is allocated to group } h \text{ based on the values of the } \lambda_{ih}\text{'s} \\ 0 & \text{o.w.} \end{cases}$$

The nonlinearity arises in the piecewise-linear max constraints. Under mild assumptions, the model obtains a solution in which no entities are misclassified [35]. In other words, every training entity is either correctly classified or placed in the reserved judgment group. Model 2 is a variation of model 1 in which the max constraints are replaced by constraints of the form $y_{gj} \geq L_{hgj}$ and the objective function is modified, placing a penalty on the value of the y_{gj}'s so that they tend to satisfy the max constraint:

$$\max \sum_{g \in \mathcal{G}} \sum_{j \in \mathcal{N}_g} (\beta u_{ggj} - \gamma y_{gj})$$

Model 3 is a mixed-integer program that is equivalent to model 1. The max constraints are replaced with the following constraints and variables

$$
\begin{aligned}
y_{gj} &\geq L_{hgj} && h,\ g \in \mathcal{G},\ j \in \mathcal{N}_g \\
\tilde{y}_{hgj} - L_{hgj} &\leq M(1 - v_{ghj}) && h,\ g \in \mathcal{G},\ j \in \mathcal{N}_g \\
\tilde{y}_{hgj} &\leq \hat{\pi}_h \hat{f}_h(x^{gj}) v_{hgj} && h,\ g \in \mathcal{G},\ j \in \mathcal{N}_g \\
\sum_{h \in \mathcal{G}} v_{hgj} &\leq 1 && g \in \mathcal{G},\ j \in \mathcal{N}_g \\
\sum_{h \in \mathcal{G}} \tilde{y}_{hgj} &= y_{gj} && g \in \mathcal{G},\ j \in \mathcal{N}_g
\end{aligned}
$$

$$
\tilde{y}_{hgj} \geq 0,\ v_{hgj} \in \{0, 1\}
$$

The three models compete effectively with standard methods on well-known data sets, but are computationally intensive [35].

A linear programming model (DALP) based on the DAMIP was developed by Lee et al. [58] which solves rapidly and produces similar results. The DALP has been successfully applied to the prediction of ultrasound-mediated disruption of cell membranes [57], automated planning volume definition in soft-tissue sarcoma adjuvant brachytherapy [56], and genomic pattern recognition in human cancer [30].

For simplicity of notation, the "hat" accents will be omitted. The prior probabilities and conditional group densities are assumed to be known, and we will not consider techniques for their estimation.

1.6 Consistency

1.6.1 The Bayes decision rule and consistency

As indicated in Section 1.4.2, the Bayes decision rule for classification is to allocate observations to the group h for which $f(h|x)$ is largest. This rule is optimal because it minimizes the probability of misclassification, or equivalently, maximizes the probability of correct classification. The following development extends the treatment in [26] of the two-group case to multiple groups.

Let $(X, Y) \in \mathbb{R}^d \times \{1, 2, \ldots, G\}$ be random variables where G is the number of groups and let μ be the probability measure for X. The random variable Y is a discrete random variable defined by a conditional distribution $f(h|x) = P\{Y = h|X = x\}$ for $h = 1, 2, \ldots, G$.

A function $\phi : \mathbb{R}^d \to \{1, 2, \ldots, G\}$ is a classifier. The probability of correct classification for the classifier is $P\{\phi(X) = Y\}$.

Let $\phi^*(x)$ be the *Bayes decision rule*, the function that assigns x to the group h for which $P\{Y = h|X = x\}$ is maximum, or equivalently, $\phi^*(x) = \arg\max_h f(h|x)$. Then for any classifier $\phi(x)$,

$$\begin{aligned}
&P\{\phi^*(X) = Y|X = x\} - P\{\phi(X) = Y|X = x\} \\
&= \sum_{h=1}^{G} P\{Y = h, \phi^*(X) = h|X = x\} - \sum_{h=1}^{G} P\{Y = h, \phi(X) = h|X = x\} \\
&= \sum_{h=1}^{G} I_{\{\phi^*(x)=h\}} P\{Y = h|X = x\} - \sum_{h=1}^{G} I_{\{\phi(x)=h\}} P\{Y = h|X = x\} \\
&= \sum_{h=1}^{G} f(h|x) I_{\{\phi^*(x)=h\}} - \sum_{h=1}^{G} f(h|x) I_{\{\phi(x)=h\}} \\
&\geq 0
\end{aligned}$$

Integrating the first and last lines with respect to $\mu(dx)$ gives

$$\int (P\{\phi^*(X) = Y|X = x\} - P\{\phi(X) = Y|X = x\})\mu(dx) = P\{\phi^*(X) = Y\} - P\{\phi(X) = Y\} \geq 0$$

Therefore, the Bayes decision function $\phi^*(x)$ is optimal for the problem

$$\max_{\phi} P\{\phi(X) = Y\}$$

Let $\phi_n(X, D_n)$ be a classification rule selected based on an i.i.d. data set $D_n = \{(X_i, Y_i)\}_{i=1}^n$ with size n. We will call a decision rule *consistent* if, as the sample size increases, the expected probability of correct classification converges to the probability of correct classification associated with the Bayes rule $P\{\phi^*(X) = Y\}$, or

$$EP\{\phi_n(X, D_n) = Y|D_n\} \to P\{\phi^*(X) = Y\}, \text{ as } n \to \infty$$

The rule is *strongly consistent* if

$$\lim_{n\to\infty} P\{\phi_n(X, D_n) = Y|D_n\} = P\{\phi^*(X) = Y\}, \text{ with probability 1}$$

The rule is *universally consistent* if the probability of correct classification converges to the probability of correct classification associated with Bayes rule for all possible distributions of X and Y.

(Ordinarily, the Bayes decision rule is defined as that which minimizes the probability of error, and the Bayes error is $P\{\phi^*(X) \neq Y|X = x\}$. A rule is called *consistent* if, as the sample size increases, the expected probability of error approaches the Bayes error. A rule is *universally consistent* if the probability of error converges to the Bayes error for all distributions of X and Y. The notion of a *strongly consistent* rule is analogously defined. For purposes of analyzing the DAMIP, the first definitions will be more useful.)

1.6.2 Vapnik-Chervonenkis Theory

Vapnik and Chervonenkis developed much of the theory concerning the selection of a classifier from a class $\mathcal{C}$ of classifiers based on empirical performance [83, 84, 82]. In particular, they investigated the selection of a classifier based on minimizing *empirical error*. The empirical error of a classifier is the difference between the error of the classifier $P\{\phi(X) \neq Y\}$ and the infimum of the error over the class $\mathcal{C}$ of classifiers. The *approximation error* is the difference between the infimum of the error over the class $\mathcal{C}$ and the Bayes error. There is a trade-off between the size of $\mathcal{C}$ and the size of the approximation error. Additionally, the empirical error is often easier to minimize when $|\mathcal{C}|$ is small.

As with the discussion of the Bayes decision rule in the previous section, the results from Vapnik-Chervonenkis Theory (VC Theory) will be discussed in terms of maximizing empirical benefit rather than minimizing empirical risk. Accordingly, associated with the optimal classification rule will be the Bayes benefit.

In practice, a classifier is selected from a class based on a sample of size n for which the *empirical benefit* is maximized. Let the empirical benefit of a classifier ϕ be

$$B_n(\phi) = \frac{\sum_{j=1}^{n} I_{\{\phi(X_j)=Y_j\}}}{n}$$

for a sample of size n. The difference between the Bayes benefit and the empirically optimal benefit of a classifier can be written as

$$(B(\phi^*) - \sup_{g\in\mathcal{C}} B(\phi)) + (\sup_{g\in\mathcal{C}} B(\phi) - B(\phi_n))$$

where the first difference is the approximation error and the second difference is the empirical error. "Error" in this context refers to the difference in benefit between the benefit of an

empirical classifier and a Bayes classifier. In the VC framework, we wish to describe how the empirical benefit converges to the supremum of the benefit possible within $\mathcal{C}$ as the sample size n increases. The rate of convergence can be bounded above by factors which depend on the characteristics of functions in $\mathcal{C}$. These factors describe the number of partitions of n data points that are possible using functions from $\mathcal{C}$.

When applied to empirical benefit maximization, the result on the convergence of frequencies to their probabilities in [83] implies that

$$P\left\{\sup_{\phi\in\mathcal{C}}|B_n(\phi)-B(\phi)|>\epsilon\right\}\leq 8\mathcal{S}(\mathcal{C},n)e^{-n\epsilon^2/8}$$

and therefore

$$P\left\{\sup_{\phi\in\mathcal{C}}B(\phi)-B(\phi_n)>\epsilon\right\}\leq 8\mathcal{S}(\mathcal{C},n)e^{-n\epsilon^2/8}$$

where $\mathcal{S}(\mathcal{C},n)$ is the *shatter coefficient* for the class $\mathcal{C}$. If $\mathcal{C}$ is the class of functions that define a single halfspace $\{x : ax \leq b\}$ in $\mathbb{R}_d$, then the shatter coefficient for $\mathcal{C}$ is bounded by $\mathcal{S}(\mathcal{C},n)\leq 2(n-1)^d+2$ (see Corollary 13.1 of [26]). If $\mathcal{C}$ is the class of functions that define the intersection of sets of t halfspaces, then the shatter coefficient is bounded by $\mathcal{S}(\mathcal{C},n)\leq(2(n-1)^d+2)^t$ (using Theorem 13.5 in [26]).

The result due to Vapnik and Chervonenkis regarding the convergence of the empirical benefit was initially presented in more general terms that describe the convergence of any empirical risk functional

$$R(\alpha)=\frac{1}{n}\sum_{i=1}^{n}Q((X_i,Y_i),\alpha)$$

where $Q((X,Y),\alpha),\ \alpha\in\Lambda$ is a set of indicator functions. If $F(x,y)$ is the probability measure on the space (X,Y), then

$$P\left\{\sup_{\alpha\in\Lambda}\left|Q((x,y),\alpha)dF(x,y)-\frac{1}{n}\sum_{i=1}^{n}Q((x_i,y_i),\alpha)\right|>\epsilon\right\}\leq 8\mathcal{S}(\Lambda,n)e^{-n\epsilon^2/8}$$

where $\mathcal{S}(\Lambda,n)$ is the shatter coefficient of the set of indicator functions [83].

1.7 Robustness and stability

Robustness is a term with separate but related definitions across many fields within science. Perhaps a unifying definition would be the sensitivity of results or conclusions to changes

in the assumptions upon which an experiment depends. In traditional statistical inference, Bayesian inference, combinatorial optimization, and computer science, the term "robustness" has been applied in different ways. Each of these definitions can be employed in the evaluation of a pattern classifier that uses mixed-integer programming.

1.7.1 Traditional statistical inference

In traditional statistical inference, robustness is often defined as the sensitivity of inferences to *outliers* or noisy data [77]. An outlier is essentially an unusual observation in a particular data set. *Noise* is any uncertainty introduced due to random disturbances in nature or the data collection process. Robustness is also used more broadly by statisticians to describe the sensitivity of a model to changes in assumptions about model parameters or to changes in the characteristics of input data.

These definitions are easily applied to pattern classification. Classification models can be considered robust if their rules are relatively insensitive to outliers in the input data or noisy input data. They can also be robust if they are able to classify new data that contains outliers or noisy data. In the other sense, classification models are said to be robust if they are insensitive to changes in model parameters or perform well on a wide variety of data sets. For example, a parametric model might be designed for data from normally distributed data, but is robust for data derived from distributions with thicker tails.

A little mentioned, but seemingly important, concept in pattern classification is that of *stability*. Stability is the sensitivity of the derived classification rules to small changes in the input data. Breiman [16, 18, 19] and Li and Belford [60] have studied methods for treating the instability of classification trees.

1.7.2 Bayesian inference

Bayesian statisticians focus on inference robustness, which is closely related to the second definition for traditional statistical inference. Within Bayesian inference, this approach implies studying the sensitivity of inferences to uncertainty or misspecification in prior probability functions and/or likelihood functions.

This notion of robustness applies only to classification models constructed within a

Bayesian framework - those for which inferences about group membership are made based on a posterior distribution.

1.7.3 Combinatorial optimization

Robustness is a rather new term in the field of combinatorial optimization, though *sensitivity analysis* is probably as old as the field itself. Ben-Tal and Nemirovski [7, 8, 9] introduced the term to describe the solution of convex optimization problems with well-defined uncertainty in the input data. Kouvelis and Yu [52], Bertsimas and Sim [11], and Atamturk [4] have applied the concept to discrete optimization.

Sensitivity analysis is a methodology for characterizing the stability of an already-obtained optimal solution. The product of sensitivity analysis is a characterization of the degree of perturbation in input data that would change the optimal solution. *Robust optimization*, in contrast, is more forward-thinking and seeks to guarantee a desired amount of stability in the solution of a problem. A *robust solution* will continue to satisfy constraints after perturbation of the input data.

1.7.4 Computer science

A computer science definition of robustness is a quality of systems that hold up well under exceptional circumstances. When applied to classification, this definition seems to coincide with the traditional statistical notion of performance on noisy data or in the presence of outliers. This definition is also closely related to the goals of robust optimization.

1.7.5 Conventions used in this dissertation

We seek to test the DAMIP under various notions of robustness and stability. Because the DAMIP is a statistical tool, developed within a Bayesian framework, involves the solution of a math program, many of the different definitions apply. Some of these definitions collapse to assume the same meaning in the context of the DAMIP.

We will take a broad definition of robustness - the ability of a classification model to perform well in the presence of contaminated data and under a variety of assumptions about model parameters and problem data. We define stability as the sensitivity of classification

rules to changes in input data. For the DAMIP, this sensitivity is equivalent to the stability of optimal solutions. We seek to determine methods of guaranteeing stable solutions or rules, in line with the goals of robust optimization and stable classification.

1.8 Outline of this dissertation

Chapters 2-4 are concerned with the solution of a mixed-integer programming formulation of the DAMIP. Chapter 2 develops a polynomial-time combinatorial algorithm for the two-group case. The algorithm does not use mixed-integer programming, but rather checks a polynomial number of possible solutions with the guarantee of finding an optimal solution.

Chapter 3 contains theoretical results concerning the solution of the mixed-integer programming formulation of the DAMIP for G groups. The solution of the mixed-integer program is shown to be NP-complete. The dimension of the polytope is characterized. The structure of the DAMIP is exploited to find edges of the conflict graph and conflict hypergraphs. Upper bounds on the values of some parameters are derived.

The consistency, robustness, and stability of the DAMIP is discussed in Chapter 4. A general form of the DAMIP is shown to be strongly universally consistent. The robustness and stability of solutions and classification rules generated by the DAMIP is discussed.

Computational methods are described in Chapter 5. The methods include a heuristic for finding integer feasible solutions, a specialized branching scheme, and cutting planes derived from the conflict graph and conflict 3-hypergraph. An algorithm for finding maximal hypercliques is presented as an extension of a maximal clique algorithm in [21].

Chapter 6 contains the results of various computational tests. The performance of CPLEX on mixed-integer programs of the DAMIP is compared to enhanced code with the techniques of Chapter 4 implemented. The relative contribution of the various computational strategies is assessed. The classification accuracy of the DAMIP is compared to standard methods including linear discriminant functions, quadratic discriminant functions, classification trees, and support vector machines. The methods are tested with both real-world and simulated data. Additional simulations explore the effects of various training conditions on classification accuracy.

Chapter 7 presents a summary of conclusions, contributions, and future research.

chapter 1

Chapter II

Two-Group Discrimination with the DAMIP

In this chapter, an $O(N^2)$ algorithm for solving the 2-group DAMIP is developed, where N is the number of training observations. An algorithm is first developed for the case when no misclassification constraints are present. The algorithm is subsequently extended for the case when misclassification constraints are present.

The input to the 2-group DAMIP is

- Estimates of the prior probabilities π_h describing the likelihood that an unknown entity belongs to group $h \in \{1, 2\}$.
- A set of entities $\mathcal{N}_h$ for each group $h \in \{1, 2\}$ such that the entities in $\mathcal{N}_h$ are known to belong to group h. Let $n_h = |N_h|$ and $N = n_1 + n_2$.
- Estimates of the conditional group density functions $f_h(\cdot)$ evaluated at the points representing each of the entities x^{gj}.
- Misclassification limits $\alpha_1 = \lfloor \alpha_{12} n_1 \rfloor$ and $\alpha_2 = \lfloor \alpha_{21} n_2 \rfloor$ that represent the highest numbers of entities from group 1 and group 2, respectively, that can be allocated to group 2 and 1, respectively.

The variables are

$$u_{hgj} = \begin{cases} 1 & \text{if entity } gj \text{ is allocated to group } h \\ 0 & \text{o.w.} \end{cases}$$

$$L_{hgj} = \text{modified posterior probability for entity } gj \text{ to be allocated to group } h$$

$$y_{gj} = \text{maximum modified posterior probability for entity } gj$$

$$\lambda_{12} = \text{linear coefficient determining the modified posterior probability}$$

$$\lambda_{21} = \text{linear coefficient determining the modified posterior probability}$$

A formulation of the problem is as follows

$$\max \sum_{g \in \mathcal{G}} \sum_{j \in \mathcal{N}_g} u_{ggj}$$

subject to

$$\begin{aligned}
L_{1gj} &= \pi_1 f_1(x^{gj}) - f_2(x^{gj})\lambda_{21} && g \in \{1,2\},\ j \in \mathcal{N}_g \\
L_{2gj} &= \pi_2 f_2(x^{gj}) - f_1(x^{gj})\lambda_{11} && g \in \{1,2\},\ j \in \mathcal{N}_g \\
u_{hgj} &= \begin{cases} 1 & \text{if } h = \arg\max\{0, L_{h'gj} : h' = 1, \ldots, G\} \\ 0 & \text{o.w.} \end{cases} && g, h \in \{1,2\},\ j \in \mathcal{N}_g \\
\sum_{j \in \mathcal{N}_g} u_{hgj} &\leq \lfloor \alpha_{hg} n_g \rfloor && h,\ g \in \{1,2\},\ h \neq g
\end{aligned}$$

$$-\infty < L_{hgj} < \infty,\ y_{gj} \geq 0,\ \lambda_{ih} \geq 0$$

The following proposition gives conditions on the input data for pairs of entities from different groups such that there does not exist a solution to the DAMIP where both entities are correctly or incorrectly classified simultaneously. Figure 4 shows how a pair of entities would be classified. For this particular pair of observations, they can be simultaneously correctly classified, but cannot both be misclassified.

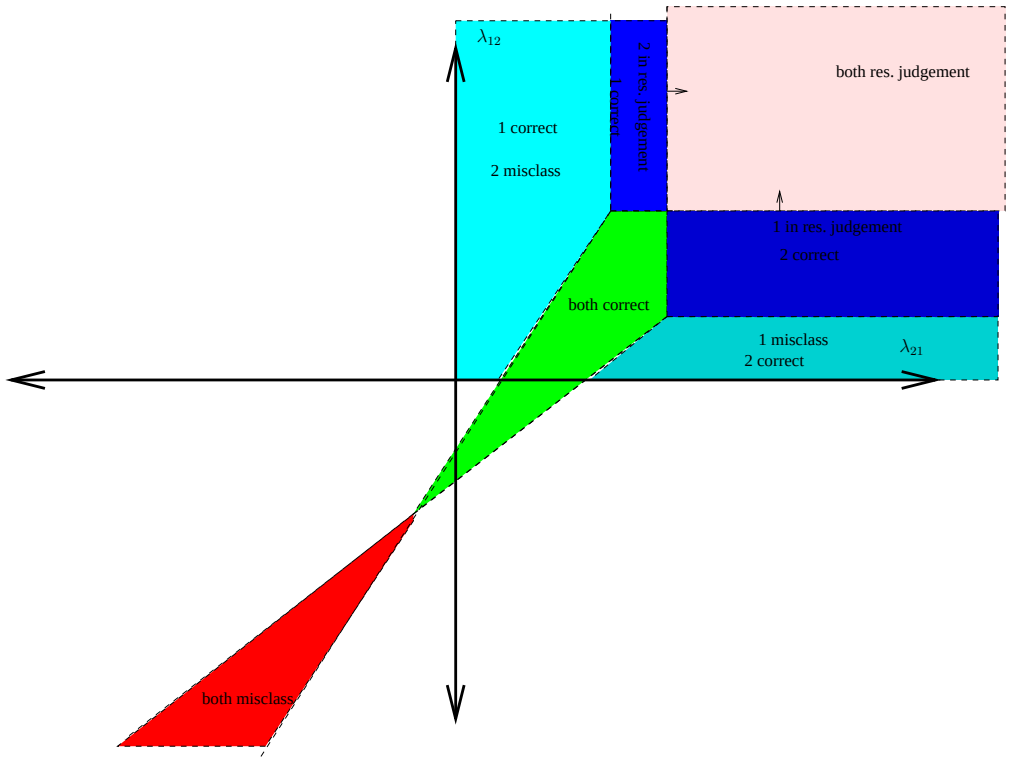


Figure 4: A representation of outcomes for a pair of entities based on the values of λ_{12} and λ_{21}. This particular pair of entities can be correctly classified when the λ_{ih} values fall in the green region. There are no positive values for the λ_{ih}'s such that both entities are misclassified.

Proposition 2.0.1. *If for two entities with different group membership,* $f_1(x^{1j})f_2(x^{2k}) - f_2(x^{1j})f_1(x^{2k}) > 0$*, then there exist values for* λ_{21} *and* λ_{12} *for which both* $u_{11j} = 1$ *and* $u_{22k} = 1$*. In other words, a solution exists with both entities correctly classified. Furthermore, there does not exist a solution with both* $u_{21j} = 1$ *and* $u_{12k} = 1$ *where both entities are misclassified.*

Conversely, if $f_1(x^{1j})f_2(x^{2k}) - f_2(x^{1j})f_1(x^{2k}) < 0$*, then values for the* λ*'s exist such that both entities are misclassified, but there are no* λ*'s for which both entities are correctly classified.*

Proof. Suppose that $f_1(x^{1j})f_2(x^{2k}) - f_2(x^{1j})f_1(x^{2k}) > 0$.

Case 1: Suppose $\pi_1 f_1(x^{1j}) \geq \pi_2 f_2(x^{1j})$. Let $\lambda_{12} = 0$ and

$$\frac{\pi_1 f_1(x^{2k})}{f_2(x^{2k})} - \pi_2 < \lambda_{21} \leq \frac{\pi_1 f_1(x^{1j})}{f_2(x^{1j})} - \pi_2$$

Note that such a value for λ_{21} is feasible because

$$\begin{aligned} \frac{\pi_1 f_1(x^{1j})}{f_2(x^{1j})} - \pi_2 - (\frac{\pi_1 f_1(x^{2k})}{f_2(x^{2k})} - \pi_2) &= \frac{\pi_1(f_1(x^{1j})f_2(x^{2k}) - f_1(x^{2k})f_2(x^{1j}))}{f_2(x^{2k})f_2(x^{1j})} \\ &> 0 \end{aligned}$$

and $\frac{\pi_1 f_1(x^{1j})}{f_2(x^{1j})} - \pi_2 \geq 0$ by the assumption for Case 1.

Using these λ's, the modified posterior probabilities are

$$\begin{aligned} L_{11j} &= \pi_1 f_1(x^{1j}) - f_2(x^{1j})\lambda_{21} > \pi_1 f_1(x^{1j}) - f_2(x^{1j})(\tfrac{\pi_1 f_1(x^{1j})}{f_2(x^{1j})} - \pi_2) = \pi_2 f_2(x^{1j}) \\ L_{21j} &= \pi_2 f_2(x^{1j}) - f_1(x^{1j})\lambda_{12} = \pi_2 f_2(x^{1j}) \\ L_{22k} &= \pi_2 f_2(x^{2k}) - f_1(x^{2k})\lambda_{12} = \pi_2 f_2(x^{2k}) \\ L_{12k} &= \pi_1 f_1(x^{2k}) - f_2(x^{2k})\lambda_{21} < \pi_1 f_1(x^{2k}) - f_2(x^{2k})(\tfrac{\pi_1 f_1(x^{2k})}{f_2(x^{2k})} - \pi_2) = \pi_2 f_2(x^{2k}) \end{aligned}$$

Therefore, $L_{11j} > L_{21j}$ and $L_{22k} > L_{12k}$ so that both entities are correctly classified.

Case 2: Now suppose $\pi_1 f_1(x^{1j}) < \pi_2 f_2(x^{1j})$. Let $\lambda_{21} = 0$ and

$$\frac{\pi_2 f_2(x^{1j})}{f_1(x^{1j})} - \pi_1 < \lambda_{12} < \frac{\pi_2 f_2(x^{2k})}{f_1(x^{2k})} - \pi_1$$

A symmetric argument to that of Case I demonstrates that $\lambda_{21} = 0$ and the given interval for λ_{12} are solutions such that both observations are correctly classified.

For the second part of the proposition, suppose that $u_{21j} = 1$ and $u_{12k} = 1$. Then

$$\begin{aligned} y_{1j} = \pi_1 f_1(x^{1j}) - \lambda_{21} f_2(x^{1j}) &\leq \pi_2 f_2(x^{1j}) - \lambda_{12} f_1(x^{1j}) \\ y_{2k} = \pi_2 f_2(x^{2k}) - \lambda_{12} f_1(x^{2k}) &\leq \pi_1 f_1(x^{2k}) - \lambda_{21} f_2(x^{2k}) \end{aligned}$$

These inequalities are equivalent to

$$\begin{aligned} \lambda_{21} &\geq \frac{\pi_1 f_1(x^{1j}) - \pi_2 f_2(x^{1j}) + \lambda_{12} f_1(x^{1j})}{f_2(x^{1j})} \\ \lambda_{12} &\geq \frac{\pi_2 f_2(x^{2k}) - \pi_1 f_1(x^{2k}) + \lambda_{21} f_2(x^{2k})}{f_1(x^{2k})} \end{aligned}$$

Therefore,

$$\begin{aligned} \lambda_{12} &\geq \frac{\pi_2 f_2(x^{2k}) - \pi_1 f_1(x^{2k}) + \lambda_{21} f_2(x^{2k})}{f_1(x^{2k})} \\ &\geq \frac{\pi_2 f_2(x^{2k}) - \pi_1 f_1(x^{2k}) + \frac{\pi_1 f_1(x^{1j}) - \pi_2 f_2(x^{1j}) + \lambda_{12} f_1(x^{1j})}{f_2(x^{1j})} f_2(x^{2k})}{f_1(x^{2k})} \\ &= \frac{\pi_1 (f_1(x^{1j}) f_2(x^{2k}) - f_2(x^{1j}) f_1(x^{2k})) + \lambda_{12} f_1(x^{1j}) f_2(x^{2k})}{f_2(x^{1j}) f_1(x^{2k})} \end{aligned}$$

Rearranging,

$$\lambda_{12}(f_2(x^{1j}) f_1(x^{2k}) - f_1(x^{1j}) f_2(x^{2k})) \geq \pi_1 (f_1(x^{1j}) f_2(x^{2k}) - f_2(x^{1j}) f_1(x^{2k}))$$

which is possible only if $f_1(x^{1j}) f_2(x^{2k}) - f_2(x^{1j}) f_1(x^{2k}) < 0$. Thus, both entities cannot be misclassified. A symmetric proof proves the converse case. □

The next proposition and corollary give necessary and sufficient conditions for training observations from 2 groups to be separated by the DAMIP without error.

Proposition 2.0.2. *In the two-group case, a solution exists with all entities correctly classified (and none misclassified) if and only if* $f_1(x^{1j})f_2(x^{2k}) - f_2(x^{1j})f_1(x^{2k}) > 0$ *for every pair of entities* j *from group 1 and* k *from group 2.*

Proof. If a solution exists with all entities correctly classified and none misclassified, then $f_1(x^{1j})f_2(x^{2k}) - f_2(x^{1j})f_1(x^{2k}) \geq 0$ for every j, k from Proposition 2.0.1. If $f_1(x^{1j})f_2(x^{2k}) - f_2(x^{1j})f_1(x^{2k}) = 0$ for some pair of entities j and k, then j and k are correctly classified if

$$\begin{aligned} y_{1j} = \pi_1 f_1(x^{1j}) - \lambda_{21} f_2(x^{1j}) &> \pi_2 f_2(x^{1j}) - \lambda_{12} f_1(x^{1j}) \\ y_{2k} = \pi_2 f_2(x^{2k}) - \lambda_{12} f_1(x^{2k}) &> \pi_1 f_1(x^{2k}) - \lambda_{21} f_2(x^{2k}) \end{aligned}$$

These inequalities are equivalent to

$$\begin{aligned} \lambda_{21} &< \frac{\pi_1 f_1(x^{1j}) - \pi_2 f_2(x^{1j}) + \lambda_{12} f_1(x^{1j})}{f_2(x^{1j})} \\ \lambda_{12} &< \frac{\pi_2 f_2(x^{2k}) - \pi_1 f_1(x^{2k}) + \lambda_{21} f_2(x^{2k})}{f_1(x^{2k})} \end{aligned}$$

Therefore, they are correctly classified if

$$\begin{aligned} \lambda_{12} &< \frac{\pi_2 f_2(x^{2k}) - \pi_1 f_1(x^{2k}) + \lambda_{21} f_2(x^{2k})}{f_1(x^{2k})} \\ &< \frac{\pi_2 f_2(x^{2k}) - \pi_1 f_1(x^{2k}) + \frac{\pi_1 f_1(x^{1j}) - \pi_2 f_2(x^{1j}) + \lambda_{12} f_1(x^{1j})}{f_2(x^{1j})} f_2(x^{2k})}{f_1(x^{2k})} \\ &= \frac{\pi_2 f_2(x^{2k}) f_2(x^{1j}) - \pi_1 f_1(x^{2k}) f_2(x^{1j}) + \pi_1 f_1(x^{1j}) f_2(x^{2k}) - \pi_2 f_2(x^{1j}) f_2(x^{2k}) + \lambda_{12} f_1(x^{1j}) f_2(x^{2k})}{f_1(x^{2k}) f_2(x^{1j})} \\ &= \lambda_{12} \end{aligned}$$

which is a contradiction. Therefore, if a solution exists with all entities correctly classified, then $f_1(x^{1j})f_2(x^{2k}) - f_2(x^{1j})f_1(x^{2k}) > 0$ for every pair of entities j and k.

If $f_1(x^{1j})f_2(x^{2k}) - f_2(x^{1j})f_1(x^{2k}) > 0$ for every j and k, then either

1. $\pi_1 f_1(x^{1j}) > \pi_2 f_2(x^{1j})$ for every j, or

2. $\pi_2 f_2(x^{2k}) > \pi_1 f_1(x^{2k})$ for every k

To demonstrate that at least one case must hold true, suppose that both conditions are false. Then $\pi_1 f_1(x^{1j}) < \pi_2 f_2(x^{1j})$ for some j and $\pi_2 f_2(x^{2k}) < \pi_1 f_1(x^{2k})$ for some k. Thus

$$\begin{aligned} f_1(x^{1j})f_2(x^{2k}) - f_2(x^{1j})f_1(x^{2k}) &< \frac{\pi_2}{\pi_1} f_2(x^{1j}) \frac{\pi_1}{\pi_2} f_1(x^{2k}) - f_2(x^{1j})f_1(x^{2k}) \\ &= f_2(x^{1j})f_1(x^{2k}) - f_2(x^{1j})f_1(x^{2k}) \\ &= 0 \end{aligned}$$

This result contradicts the assumption. Therefore, at least one of the conditions holds. Without loss of generality, suppose $\pi_1 f_1(x^{1j}) > \pi_2 f_2(x^{1j})$ for every j in group 1. Then set $\lambda_{12} = 0$ and let

$$\max_k \frac{\pi_1 f_1(x^{2k})}{f_2(x^{2k})} - \pi_2 < \lambda_{21} \leq \min_j \frac{\pi_1 f_1(x^{1j})}{f_2(x^{1j})} - \pi_2$$

The value for λ_{21} is feasible as in the proof of Proposition 2.0.1 because $f_1(x^{1j})f_2(x^{2k}) - f_2(x^{1j})f_1(x^{2k}) > 0$ for all j and k and $\frac{\pi_1 f_1(x^{1j})}{f_2(x^{1j})} - \pi_2 \geq 0$ for every j. Using these values for the λ's, $L_{11j} > L_{21j}$ and $L_{11j} \geq 0$ for every j, and $L_{22k} > L_{12k}$ and $L_{22k} \geq 0$ for every k. Thus, every entity is correctly classified, and none are misclassified. □

Corollary 2.0.3. *In the two-group case, a solution exists with all entities correctly classified (and none misclassified) if and only if the Bayesian classification rule correctly classifies every entity from at least one of the groups.*

Proof. The condition for the Bayesian classification to correctly classify all entities from at least one of the groups is equivalent to the statement that

1. $\pi_1 f_1(x^{1j}) > \pi_2 f_2(x^{1j})$ for every j, or

2. $\pi_2 f_2(x^{2k}) > \pi_1 f_1(x^{2k})$ for every k

This condition is equivalent to the condition $f_1(x^{1j})f_2(x^{2k}) - f_2(x^{1j})f_1(x^{2k}) > 0$ for every j, k, and is therefore necessary and sufficient for the correct classification of all entities. □

2.1 Two-group DAMIP without misclassification constraints

The next two propositions show that an optimal solution to the 2-group DAMIP without misclassification constraints occurs along the line $\lambda_{12} = 0$ or $\lambda_{21} = 0$. They lay the foundation for the algorithm for 2-group DAMIP without misclassification constraints.

Proposition 2.1.1. *In the two-group case, at least one solution exists such that*

1. *The number of correctly classified entities is maximized, and*
2. *Either $\lambda_{12} = 0$ or $\lambda_{21} = 0$.*

Proof. Suppose the assertion is false. Consider a solution where the maximum number of entities are correctly classified. For this solution $\lambda_{12} > 0$ and $\lambda_{21} > 0$.

Case I: Let $j^* = \arg\min_j \{f_1(x^{1j}) : j \in \mathcal{G}\}$. Suppose $\pi_1 f_1(x^{1j^*}) \leq \pi_2 f_2(x^{1j^*})$. Then let $\lambda_{21}^{new} = 0$ and

$$\lambda_{12}^{new} = \frac{\pi_2 f_2(x^{1j^*}) - \pi_1 f_1(x^{1j^*})}{f_1(x^{1j^*})}$$

For entities from group 1 that were correctly classified from before,

$$\begin{aligned} L_{11j} - L_{21j} &= \pi_1 f_1(x^{1j}) - f_2(x^{1j})\lambda_{21}^{new} - \pi_2 f_2(x^{1j}) + f_1(x^{1j})\lambda_{12}^{new} \\ &= \pi_1 f_1(x^{1j}) - \pi_2 f_2(x^{1j}) + f_1(x^{1j})\frac{\pi_2 f_2(x^{1j^*})}{-}\pi_1 f_1(x^{1j^*})f_1(x^{1j^*}) \\ &= \frac{\pi_2(f_1(x^{1j})f_2(x^{1j^*}) - f_2(x^{1j})f_1(x^{1j^*}))}{f_1(x^{1j^*})} \\ &> 0 \end{aligned}$$

The last inequality is due to the definition of j^*. Also note that $L_{11j} > 0$, so that all such entities j are correctly classified. For all entities from group 2 that were previously correctly classified,

$$\begin{aligned} L_{22k} - L_{12k} &= \pi_2 f_2(x^{2k}) - f_1(x^{2k})\lambda_{12}^{new} - \pi_1 f_1(x^{2k}) + f_2(x^{2k})\lambda_{21}^{new} \\ &= \frac{\pi_2(f_2(x^{2k})f_1(x^{1j^*}) - f_1(x^{2k})f_2(x^{1j^*}))}{f_1(x^{1j^*})} \\ &> 0 \end{aligned}$$

The last inequality is due to Proposition 2.0.1 and the fact that entities k and j^* were previously correctly classified. Therefore, all entities that were previously correctly classified are correctly classified with the new λ values.

Case II: Let $k^* = \arg\min_k\{f_2(x^{2k}) : k \in \mathcal{G}\}$. Suppose $\pi_2 f_2(x^{2k^*}) \leq \pi_1 f_1(x^{2k^*})$. Then let $\lambda_{12}^{new} = 0$ and

$$\lambda_{21}^{new} = \frac{\pi_1 f_2(x^{1k^*}) - \pi_2 f_2(x^{2k^*})}{f_2(x^{2k^*})}$$

By an argument symmetric to that for Case I, all entities that were previously correctly classified are correctly classified with the new λ values.

Case III: Suppose for $j^* = \arg\min_j\{f_1(x^{1j}) : j \in \mathcal{G}\}$ and $k^* = \arg\min_k\{f_2(x^{2k}) : k \in \mathcal{G}\}$ that $\pi_1 f_1(x^{1j^*}) \geq \pi_2 f_2(x^{1j^*})$ and $\pi_2 f_2(x^{2k^*}) \geq \pi_1 f_1(x^{2k^*})$. Then the Bayesian classification rule correctly classifies all entities by setting $\lambda_{12} = \lambda_{21} = 0$.

□

Proposition 2.1.2. *In the two-group case, for any solution that maximizes the number of correctly classified entities, there are no entities placed in the reserved judgment region.*

Proof. Suppose that the assertion is false, so that there exists a solution with at least one entity placed in the reserved judgment class and the number of correctly classified entities is maximized. Suppose also, without loss of generality, that an entity from group 1 is placed

in the reserved judgment region and that $j^* = \arg\min_j\{f_1(x^{1j}) : j \in \mathcal{G}$ and j is classified in the reserved judgment region$\}$. Then

$$\lambda_{21} > \frac{\pi_1 f_1(x^{1j^*})}{f_2(x^{1j^*})} \text{ and } \lambda_{12} > \frac{\pi_2 f_2(x^{1j^*})}{f_1(x^{1j^*})}$$

Case I: Suppose $\pi_1 f_1(x^{1j^*}) \leq \pi_2 f_2(x^{1j^*})$. Then let

$$\lambda_{21}^{new} = 0 \text{ and } \lambda_{12}^{new} = \frac{\pi_2 f_2(x^{1j^*}) - \pi_1 f_1(x^{1j^*})}{f_1(x^{1j^*})}$$

For entities that were previously correctly classified from group 1, $L_{11j} = \pi_1 f_1(x^{1j}) - f_2(x^{1j})\lambda_{21} \geq 0$, and therefore.

$$\frac{\pi_1 f_1(x^{1j^*})}{f_2(x^{1j^*})} < \lambda_{21} \leq \frac{\pi_1 f_1(x^{1j})}{f_2(x^{1j})}$$

which implies

$$\frac{\pi_1 f_1(x^{1j})}{f_2(x^{1j})} - \frac{\pi_1 f_1(x^{1j^*})}{f_2(x^{1j^*})} > 0$$

$$\Rightarrow f_1(x^{1j}) f_2(x^{1j^*}) - f_1(x^{1j^*}) f_2(x^{1j}) > 0$$

Then,

$$\begin{aligned} L_{11j} - L_{21j} &= \pi_1 f_1(x^{1j}) - f_2(x^{1j})\lambda_{21}^{new} - \pi_2 f_2(x^{1j}) + f_1(x^{1j})\lambda_{12}^{new} \\ &= \pi_1 f_1(x^{1j}) - \pi_2 f_2(x^{1j}) + f_1(x^{1j})\left(\frac{\pi_2 f_2(x^{1j^*}) - \pi_1 f_1(x^{1j^*})}{f_1(x^{1j^*})}\right) \\ &= \frac{\pi_2(f_2(x^{1j^*}) f_1(x^{1j}) - f_2(x^{1j}) f_1(x^{1j^*}))}{f_1(x^{1j^*})} \\ &> 0 \end{aligned}$$

Also note that $L_{11j} > 0$ so that all entities from group 1 that were previously correctly classified are still correctly classified. For entities from group 2 that were correctly classified $L_{22k} = \pi_2 f_2(x^{2k}) - f_1(x^{2k})\lambda_{12}$, so that

$$\frac{\pi_2 f_2(x^{1j^*})}{f_1(x^{1j^*})} < \lambda_{12} < \frac{\pi_2 f_2(x^{2k})}{f_1(x^{2k})}$$

$$\Rightarrow f_2(x^{2k}) f_1(x^{1j^*}) - f_2(x^{1j^*}) f_1(x^{2k}) > 0$$

Using this relation and λ_{12}^{new} and λ_{21}^{new}, $L_{22k} > 0$ and $L_{22k} > L_{12k}$ so that entities that were correctly classified before are correctly classified with the new λ's. For j^*, note that

$$\begin{aligned} L_{11j^*} - L_{21j^*} &= \pi_1 f_1(x^{1j^*}) - f_2(x^{1j^*})\lambda_{21}^{new} - \pi_2 f_2(x^{1j^*}) + f_1(x^{1j^*})\lambda_{12}^{new} \\ &= 0 \end{aligned}$$

Note that $L_{11j^*} > 0$. Because the inequalities were strict for the other entities, the value of λ_{12}^{new} can be increased slightly so that entity j^* is correctly classified. Therefore, all entities that were correctly classified are correctly classified with the new values for the λ's. In addition, another entity j^* is correctly classified, which contradicts the assumption that the solution is optimal.

Case II: Suppose $\pi_1 f_1(x^{1j^*}) > \pi_2 f_2(x^{1j^*})$. Then let

$$\lambda_{12}^{new} = 0 \text{ and } \lambda_{21}^{new} = \frac{\pi_1 f_1(x^{1j^*}) - \pi_2 f_2(x^{1j^*})}{f_2(x^{1j^*})}$$

Then for entities previously correctly classified from group 1, $f_1(x^{1j}) f_2(x^{1j^*}) - f_1(x^{1j^*}) f_2(x^{1j}) > 0$, $L_{11j} > 0$, and $L_{11j} > L_{21j}$. For entities that were correctly classified from group 2, $f_2(x^{2k}) f_1(x^{1j^*}) - f_1(x^{2k}) f_2(x^{1j^*}) > 0$, $L_{22k} > 0$, and $L_{22k} > L_{12k}$. For j^*, $L_{11j^*} = L_{21j^*}$. Therefore, a solution exists with more entities correctly classified, a contradiction.

□

The following algorithm iterates through possible solutions to the 2-group DAMIP without misclassification constraints by setting $\lambda_{12} = 0$ or $\lambda_{21} = 0$.

Algorithm 2.1.3. *1. Let* $S = \{s : \pi_1 P\{\textit{entity } s \textit{ assigned to group 1}\} > \pi_2 P\{\textit{entity } s \textit{ assigned to group 2}\}\}$. *Sort* S *in ascending order by* $P\{s \textit{ assigned to group 1}\}$ *(and reassign indices). For each* $s \in S$, *calculate*

$$\lambda_{12}^{s} = \frac{\pi_2 P\{s \textit{ assigned to group 2}\} - \pi_1 P\{s \textit{ assigned to group 1}\}}{P\{s \textit{ assigned to group 1}\}}$$

Let $T = \{t : \pi_1 P\{t \textit{ assigned to group 1}\} < \pi_2 P\{t \textit{ assigned to group 2}\}\}$. *Sort* T *ascending order by* $P\{t \textit{ assigned to group 2}\}$ *(and reassign indices). For each* $t \in T$, *calculate*

$$\lambda_{21}^{t} = \frac{\pi_1 P\{t \textit{ assigned to group 1}\} - \pi_2 P\{t \textit{ assigned to group 2}\}}{P\{t \textit{ assigned to group 2}\}}$$

Set $\lambda_{12}^{0} = \lambda_{21}^{0} = 0$.

2. *Set* $\lambda_{12}^{*} = \lambda_{21}^{*} = 0$.

3. *If every entity from group 1 has* $\pi_1 f_1(x^{1j}) \geq \pi_2 f_2(x^{1j})$ *and every entity from group 2 has* $\pi_2 f_2(x^{2k}) \geq \pi_1 f_1(x^{2k})$, *then set optimal = number of entities and stop.*

4. *optimal* $= 0$

5. *for* $(s = 1;\ s \leq |S|;\ ++s)$

   - *currnumcorrect = 0*
   - *Choose a value for* λ_{12} *such that* $\lambda_{12}^{s-1} < \lambda_{12} < \lambda_{12}^{s}$.
   - *for each (entity j from group 1)*
     - *if* $(f_1(x^{1j}) \geq P\{s \textit{ assigned to group 1}\})$, $++$*currnumcorrect*
   - *for each (entity k from group 2)*
     - *if* $(f_2(x^{2k}) \geq P\{s - 1 \textit{ assigned to group 2}\})$, $++$*currnumcorrect*

- *if (optimal* $<$ *currnumcorrect)*
  - *optimal* $=$ *currnumcorrect*
  - $\lambda_{12}^* = \lambda_{12}$

6. *for* $(t = 1;\ t \leq |T|;\ ++t)$
   - *currnumcorrect = 0*
   - *Choose a value for* λ_{21} *such that* $\lambda_{21}^{t-1} < \lambda_{21} < \lambda_{21}^{t}$.
   - *for each (entity j from group 1)*
     - *if* $(f_1(x^{1j}) \geq P\{t-1$ *assigned to group 1*$\})$, $++$*currnumcorrect*
   - *for each (entity k from group 2)*
     - *if* $(f_2(x^{2k}) \geq P\{t$ *assigned to group 2*$\})$, $++$*currnumcorrect*
   - *if (optimal* $<$ *currnumcorrect)*
     - *optimal* $=$ *currnumcorrect*
     - $\lambda_{21}^* = \lambda_{21},\ \lambda_{12}^* = 0$

The next proposition shows that the algorithm terminates in $O(N^2)$ operations.

Proposition 2.1.4. *If* $f_1(x^{1j})f_2(x^{2k}) - f_2(x^{1j})f_1(x^{2k}) \neq 0$ *for all* j *and* k*, then Algorithm 2.1.3 finds a solution such that the maximum number of entities is correctly classified. The algorithm terminates in* $O(N^2)$ *time, where* N *is the number of entities.*

Proof. According to Proposition 2.1.1, there exists a solution such that the number of correctly classified entities is maximized and either $\lambda_{12} = 0$ or $\lambda_{21} = 0$. At these points, no entities are placed in the reserved judgment region, as shown in Proposition 2.1.2. Step 1 of the algorithm calculates the endpoints of intervals for the λ's over which an entity is correctly classified as defined by $L_{11j} \geq L_{21j}$ and $L_{22k} \geq L_{12k}$. The algorithm chooses points in the interior of the endpoints to avoid values for which an entity is both correctly classified and misclassified. The optimal λ's are chosen by checking which entities are correctly classified over an interval.

The sorting of the entities in step 1 takes $O(N \log N)$ using standard sorting methods. A total of N intervals for the $\lambda' s$ is checked, and for each interval, the classification state of each entity is checked. Therefore, the algorithm finds an optimal solution in $O(N^2)$ time. □

2.2 Two-group DAMIP with misclassification constraints

We begin by presenting an algorithm for the two-group DAMIP with misclassification constraints, which is an extension of the algorithm for the case with no misclassification constraints. The subsequent analysis proves that the algorithm is correct and runs in $O(N^2)$ time.

Algorithm 2.2.1. *Given α_1, α_2 the tolerance for misclassified entities for groups 1 and 2, respectively.*

1. *Let $S = \{s : \pi_1 P\{\text{entity } s \text{ assigned to group 1}\} > \pi_2 P\{\text{entity } s \text{ assigned to group 2}\}\}$. Sort S in ascending order by $P\{s \text{ assigned to group 1}\}$ (and reassign indices).*

   Let $T = \{t : \pi_1 P\{t \text{ assigned to group 1}\} < \pi_2 P\{t \text{ assigned to group 2}\}\}$. Sort T ascending order by $P\{t \text{ assigned to group 2}\}$ (and reassign indices).

2. *Set optimal = 0, $\lambda_{12}^* = \lambda_{21}^* = 0$*

3. *If every entity from group 1 has $\pi_1 f_1(x^{1j}) \geq \pi_2 f_2(x^{1j})$ and every entity from group 2 has $\pi_2 f_2(x^{2k}) \geq \pi_1 f_1(x^{2k})$, then set optimal = number of entities, $\lambda_{12}^* = \lambda_{21}^* = 0$ and stop.*

4. *for ($s = 1$; $s \leq |S|$; $++s$)*

   (a) *Set $currnumcorrect_1 = currnumcorrect_2 = curnummisclass_1 = curnummisclass_2 = currresjudgment_1 = currresjudgment_2 = saved_1 = saved_2 = 0$*

   (b) *for each (entity j from group 1)*

   - *if ($f_1(x^{1j}) \geq P\{s \text{ assigned to group 1}\}$), $++currnumcorrect_1$*
   - *else, $++currnummisclass_1$*

(c) *for each (entity k from group 2)*

- *if* ($f_2(x^{2k}) \geq P\{s-1$ *assigned to group 2*$\}$), $++currnumcorrect_2$
- *else,* $++currnummisclass_2$

(d) *for* ($s^{'} = s$; $s^{'} \leq |S|$; $++s^{'}$)

- *if* ($saved_1 = currnummisclass_1 - \alpha_1$), *set* $j^* =$ *entity* $s^{'}$ *and* $s^{'} = |S| + 1$
- *if (entity* $s^{'}$ *is in group 1),* $++saved_1$*; else,* $++currresjudgment_2$

(e) *for* ($s^{'} = s - 1$; $s^{'} > 0$; $--s^{'}$)

- *if* ($saved_2 = currnummisclass_2 - \alpha_2$), *set* $k^* =$ *entity* $s^{'}$ *and* $s^{'} = 0$
- *if (entity* $s^{'}$ *is in group 2),* $++saved_2$*; else,* $++currresjudgment_1$

(f) *for* ($t^{'} = 1$; $t^{'} \leq |T|$; $++t^{'}$)

- *if* ($saved_2 = currnummisclass_2 - \alpha_2$), *set* $k^* =$ *entity* $t^{'}$ *and* $t^{'} = |T| + 1$
- *if (entity* $t^{'}$ *is in group 2),* $++saved_2$*; else,* $++currresjudgment_1$

(g) *if* (($saved_1 = currnummisclass_1 - \alpha_1$) *and* ($saved_2 = currnummisclass_2 - \alpha_2$) *and* ($optimal < currnumcorrect_1 + currnumcorrect_2 - currresjudgment_1 - currresjudgment_2$))

- $optimal = currnumcorrect_1 + currnumcorrect_2 - currresjudgment_1 - currresjudgment_2$
- *Set* $\lambda_{12}^* = \frac{\pi_2 f_2(x^{1j^*})}{f_1(x^{1j^*})} + \epsilon$, $\lambda_{21}^* = \frac{\pi_1 f_1(x^{2k^*})}{f_2(x^{2k^*})} + \epsilon$

5. *for* ($t = 1$; $t \leq |T|$; $++t$)

(a) *Set* $currnumcorrect_1 = currnumcorrect_2 = curnummisclass_1 = curnummmisclass_2 = currresjudgment_1 = currresjudgment_2 = saved_1 = saved_2 = 0$

(b) *for each (entity j from group 1)*

- *if* ($f_1(x^{1j}) \geq P\{t-1$ *assigned to group 1*$\}$), $++currnumcorrect_1$
- *else,* $++currnummisclass_1$

(c) *for each (entity k from group 2)*

- *if* ($f_2(x^{2k}) \geq P\{t$ *assigned to group 2*$\}$), $++currnumcorrect_2$
- *else,* $++currnummisclass_2$

(d) for ($t^{'} = t - 1$; $t^{'} > 0$; $--t^{'}$)

- *if* ($saved_1 = currnummisclass_1 - \alpha_1$), *set* $j^* =$ *entity* $t^{'}$ *and* $t^{'} = 0$
- *if (entity* $t^{'}$ *is in group 1),* $++saved_1$*; else,* $++currresjudgment_2$

(e) for ($s^{'} = 1$; $s^{'} \leq |S|$; $++s^{'}$)

- *if* ($saved_2 = currnummisclass_2 - \alpha_2$), *set* $k^* =$ *entity* $s^{'}$ *and* $s^{'} = |S| + 1$
- *if (entity* $s^{'}$ *is in group 1),* $++saved_1$*; else,* $++currresjudgment_2$

(f) for ($t^{'} = t$; $t^{'} \leq |T|$; $++t^{'}$)

- *if* ($saved_2 = currnummisclass_2 - \alpha_2$), *set* $k^* =$ *entity* $t^{'}$ *and* $t^{'} = 0$
- *if (entity* $t^{'}$ *is in group 2),* $++saved_2$*; else,* $++currresjudgment_1$

(g) if (($saved_1 = currnummisclass_1 - \alpha_1$) *and* ($saved_2 = currnummisclass_2 - \alpha_2$) *and* ($optimal < currnumcorrect_1 + currnumcorrect_2 - currresjudgment_1 - currresjudgment_2$))

- $optimal = currnumcorrect_1 + currnumcorrect_2 - currresjudgment_1 - currresjudgment_2$
- *Set* $\lambda^*_{12} = \frac{\pi_2 f_2(x^{1j^*})}{f_1(x^{1j^*})} + \epsilon$, $\lambda^*_{21} = \frac{\pi_1 f_1(x^{2k^*})}{f_2(x^{2k^*})} + \epsilon$

Note: Setting the λ's involves adding an ϵ to the RHS value, where ϵ is chosen sufficiently small so that the classification of at most one observation changes.

The algorithm iterates through solutions restricted to $\lambda_{12} = 0$ or $\lambda_{21} = 0$ and then shifts the restricted λ from zero until the misclassification constraints are satisfied. The next two lemmas characterize the behavior of the classification as the restricted λ's are shifted. The first lemma gives conditions under which entities are placed in the reserved judgment region, and the second lemma gives conditions under which observations are correctly classified and misclassified. Figure 5 can be useful for visualizing these lemmas.

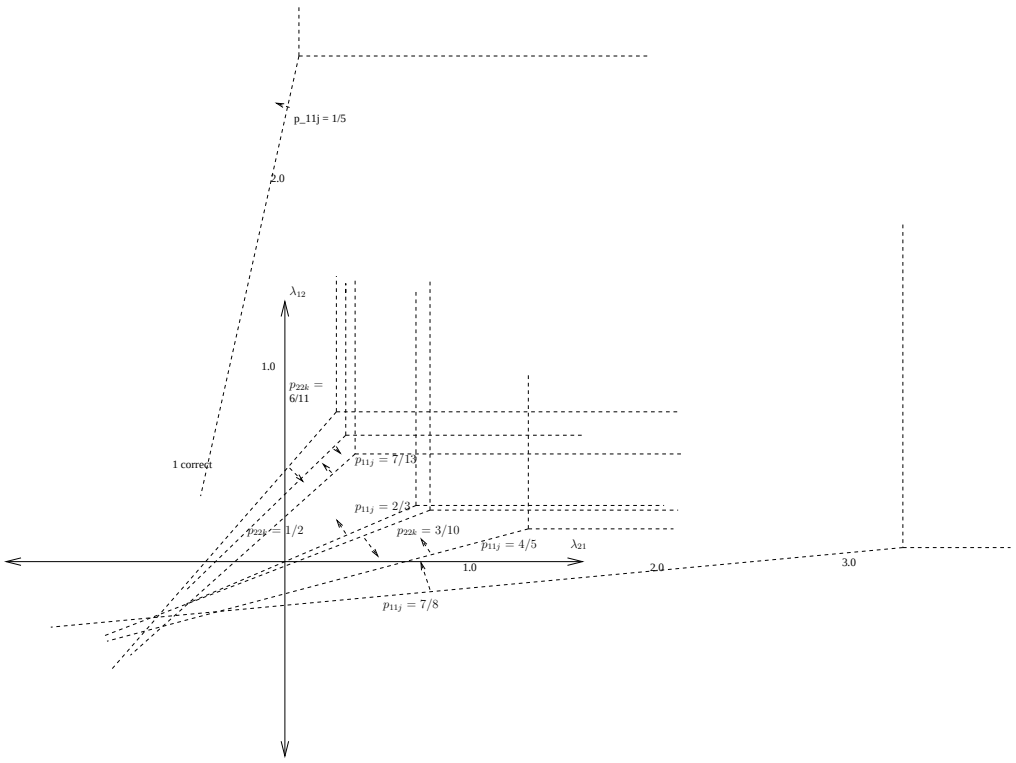


Figure 5: A graphical representation of several entities in the λ_{12}-λ_{21} space. As in Figure 4, input data define regions in which entities are correctly classified, misclassified, or placed in the reserved judgment region. The arrows point to the region for which an entity is correctly classified. For a nonzero λ_{12} (λ_{21}), increasing λ_{21} (λ_{12}) from zero successively correctly classifies entities from group 2 (group 1), misclassifies entities from group 1 (group 2), and eventually places entities in the reserved judgment region.

Lemma 2.2.2. *Suppose $f_1(x^{1j'}) > f_1(x^{2k'})$ for entities j' from group 1 and k' from group 2, and both are placed in the reserved judgment region. Then entities from group 1 with $f_1(x^{2k'}) < f_1(x^{1j}) < f_1(x^{1j'})$ and entities from group 2 with $f_2(x^{1j'}) < f_2(x^{2k}) < f_2(x^{2k'})$ are also placed in the reserved judgment region.*

If $f_1(x^{1j'}) < f_1(x^{2k'})$ and both entities are placed in the reserved judgment region, then entities from group 1 with $f_1(x^{1j'}) < f_1(x^{1j}) < f_1(x^{2k'})$ and entities from group 2 with $f_2(x^{2k'}) < f_2(x^{2k}) < f_2(x^{1j'})$ are also placed in the reserved judgment region.

Proof. Suppose $f_1(x^{1j'}) > f_1(x^{2k'})$ and j' and k' are placed in the reserved judgment region. Then $f_2(x^{1j'}) < f_2(x^{2k'})$ and

$$\begin{aligned} \lambda_{21} &> \frac{\pi_1 f_1(x^{1j'})}{f_2(x^{1j'})} > \frac{\pi_1 f_1(x^{2k'})}{f_2(x^{2k'})} \\ \lambda_{12} &> \frac{\pi_2 f_2(x^{2k'})}{f_1(x^{2k'})} > \frac{\pi_2 f_2(x^{1j'})}{f_1(x^{1j'})} \end{aligned}$$

For entities from group 1 with $f_1(x^{2k'}) < f_1(x^{1j}) < f_1(x^{1j'})$,

$$\begin{aligned} \lambda_{21} &> \frac{\pi_1 f_1(x^{1j'})}{f_2(x^{1j'})} > \frac{\pi_1 f_1(x^{1j})}{f_2(x^{1j})} > \frac{\pi_1 f_1(x^{2k'})}{f_2(x^{2k'})} \\ \lambda_{12} &> \frac{\pi_2 f_2(x^{2k'})}{f_1(x^{2k'})} > \frac{\pi_2 f_2(x^{1j})}{f_1(x^{1j})} > \frac{\pi_2 f_2(x^{1j'})}{f_1(x^{1j'})} \end{aligned}$$

These constraints on λ_{21} and λ_{12} force $L_{11j} < 0$ and $L_{21j} < 0$, placing entities j in the reserved judgment region. And for entities from group 2 with $f_2(x^{1j'}) < f_2(x^{2k}) < f_2(x^{2k'})$,

$$\begin{aligned} \lambda_{21} &> \frac{\pi_1 f_1(x^{1j'})}{f_2(x^{1j'})} > \frac{\pi_1 f_1(x^{2k})}{f_2(x^{2k})} > \frac{\pi_1 f_1(x^{2k'})}{f_2(x^{2k'})} \\ \lambda_{12} &> \frac{\pi_2 f_2(x^{2k'})}{f_1(x^{2k'})} > \frac{\pi_2 f_2(x^{2k})}{f_1(x^{2k})} > \frac{\pi_2 f_2(x^{1j'})}{f_1(x^{1j'})} \end{aligned}$$

These constraints on λ_{21} and λ_{12} force $L_{22k} < 0$ and $L_{12k} < 0$, placing entities k in the reserved judgment region. A similar argument shows that the assertion holds when $f_1(x^{1j'}) < f_1(x^{2k'})$. □

Lemma 2.2.3. *If for two entities with the same group membership, the entity with the lower correct classification probability is correctly classified, then the entity with the higher correct classification probability is also correctly classified. If the entity with the higher correct*

classification probability is misclassified, then the entity with lower correct classification probability is also misclassified.

For example, if $f_1(x^{1j_1}) > f_1(x^{1j_2})$, and entity j_2 is correctly classified, then entity j_1 is correctly classified. If entity j_1 is misclassified, then entity j_2 is correctly classified.

Proof. Suppose, without loss of generality, that j_1 and j_2 are in group 1 and $f_1(x^{1j_1}) > f_1(x^{1j_2})$ and entity j_2 is correctly classified. Then,

$$\pi_1 f_1(x^{1j_2}) - f_2(x^{1j_2})\lambda_{21} \geq \pi_2 f_2(x^{1j_2}) - f_1(x^{1j_2})\lambda_{12}$$

But $\pi_1 f_1(x^{1j_1}) - f_2(x^{1j_1})\lambda_{21} > \pi_1 f_1(x^{1j_2}) - f_2(x^{1j_2})\lambda_{21}$ and $\pi_2 f_2(x^{1j_2}) - f_1(x^{1j_2})\lambda_{12} > \pi_2 f_2(x^{1j_1}) - f_1(x^{1j_1})\lambda_{12}$, so that

$$\pi_1 f_1(x^{1j_1}) - f_2(x^{1j_1})\lambda_{21} \geq \pi_2 f_2(x^{1j_1}) - f_1(x^{1j_1})\lambda_{12}$$

so that $L_{11j_1} \geq L_{21j_1}$. Also,

$$\lambda_{21} < \frac{\pi_1 f_1(x^{1j_2})}{f_2(x^{1j_2})} < \frac{\pi_1 f_1(x^{1j_1})}{f_2(x^{1j_1})}$$

so that $L_{11j_1} \geq 0$. Therefore, entity j_1 is correctly classified. A symmetric argument shows that if entity j_1 is misclassified, then entity j_2 is misclassified. □

The following proposition follows from the lemmas and algorithm above, and is the main result of this chapter. It states that the 2-group DAMIP can be solved in $O(N^2)$ time.

Proposition 2.2.4. *Algorithm 2.2.1 finds an optimal solution to the problem of maximizing the number of correctly classified entities such that the number of misclassified entities from group 1 is no more than α_1 and the number of misclassified entities from group 2 is no more than α_2. The algorithm runs in $O(N^2)$ time, where N is the number of entities.*

If a tolerance α is given for the total number of misclassified entities, an optimal solution can be found in $O(N^3)$ time.

Proof. The algorithm begins by finding a solution with one of the λ's set to 0. The minimum number of entities from groups 1 and 2 are placed in the reserved judgment by increasing the λ's in order to reduce the number of misclassified entities to the tolerance level.

For each group i, the loops (d), (e), and (f) search for the currnummisclass$_i - \alpha_i$ entities from group i that are misclassified. Due to Lemma 2.2.2, finding the misclassified entities with the largest correct classification probabilities minimizes the number of entities placed in the reserved judgment region that were previously correctly classified. These are the entities that are "saved". Further, due to Lemma 2.2.3, the number of entities that remain correctly classified is maximized because the λ's are kept at the lowest levels possible.

Steps 4 and 5 together iterate through each entity. The inner loops (b), (c), (d), (e), (f) separately check each entity at most once. Therefore the complexity of the algorithm is $O(N^2)$.

If a tolerance α is given for the total number of misclassified entities, then the algorithm can be run for every possible value of $\alpha_1 = 1 - \alpha_2$, which is N times. Therefore, an optimal solution can be found in $O(N^3)$ time. □

chapter 2

Chapter III

Formulating and Solving the Mixed-Integer Programming Formulation of the DAMIP

The input to the mixed-integer programming formulation of the DAMIP is

- A set of groups $\mathcal{G}$.
- Estimates of prior probabilities π_h describing the likelihood that an entity with unknown attribute values belongs to group $h \in \mathcal{G}$.
- A set of entities $\mathcal{N}_h$ for each group $h \in \mathcal{G}$ such that the entities in $\mathcal{N}_h$ are known to belong to group h.
- The conditional group density functions evaluated at the points representing each of the entities. The conditional group density function for observation j with known group membership g for allocation to group h is written as $f_h(x^{gj})$.
- Misclassification limits $\alpha_{hg} \in [0, 1]$ that represent the highest proportion of entities from group h that can be allocated to group g.

The variables are

$$u_{hgj} = \begin{cases} 1 & \text{if entity } x^{gj} \text{ is allocated to group } h \\ 0 & \text{o.w.} \end{cases}$$

$L_{hgj} =$ modified posterior probability for entity x^{gj} to be allocated to group h

$y_{gj} =$ maximum modified posterior probability for entity x^{gj}

λ_{ih} = linear coefficient determining the modified posterior probabilities

The objective of the optimization model is to select λ_{ih}'s such that the number of correctly classified training entities is maximized subject to limits on the number of misclassified entities. Note that the λ_{ih} variables completely determine the values of all other variables.

3.1 Formulations

Consider the following formulation of the mixed-integer program associated with the DAMIP which is a generalization of Model 1 of Gallagher et. al [35].

$$\max \sum_{g \in \mathcal{G}} \sum_{j \in \mathcal{N}_g} u_{ggj}$$

subject to

$$\begin{aligned}
L_{hgj} &= \pi_h f_h(x^{gj}) - \sum_{\substack{i \in \mathcal{G} \\ i \neq h}} f_i(x^{gj}) \lambda_{ih} && g, h \in \mathcal{G},\ j \in \mathcal{N}_g \\
u_{hgj} &= \begin{cases} 1 & \text{if } h = \arg\max\{0, L_{h'gj} : h' = 1, \ldots, G\} \\ 0 & \text{o.w.} \end{cases} && g, h \in \mathcal{G},\ j \in \mathcal{N}_g \\
\sum_{j \in \mathcal{N}_g} u_{hgj} &\leq \lfloor \alpha_{hg} n_g \rfloor && h,\ g \in \mathcal{G},\ h \neq g
\end{aligned}$$

$$-\infty < L_{hgj} < \infty,\ y_{gj} \geq 0,\ \lambda_{ih} \geq 0$$

where $\mathcal{G} = G$ and $|\mathcal{N}_g| = n_g$.

The model explicitly allocates an observation x^{gj} to a group h based on the maximum modified posterior probability $\max_h L_{hgj}$. A nonlinear mixed-integer programming formulation of the above math program is

$$\text{maximize} \sum_{g \in \mathcal{G}} \sum_{j \in \mathcal{N}_g} u_{ggj}$$

subject to

$$\begin{aligned}
L_{hgj} &= \pi_h f_h(x) - \sum_{\substack{i \in \mathcal{G} \\ i \neq h}} f_i(x) \lambda_{ih} && g,\ h \in \mathcal{G},\ j \in \mathcal{N}_g && \\
y_{gj} &= \max\{0, L_{hgj} : h = 1, \ldots, G\} && g \in \mathcal{G},\ j \in \mathcal{N}_g && (1) \\
y_{gj} - L_{hgj} &\leq M(1 - u_{hgj}) && g,\ h \in \mathcal{G},\ j \in \mathcal{N}_g && (2) \\
y_{gj} &\leq M(1 - u_{0gj}) && g \in \mathcal{G},\ j \in \mathcal{N}_g && (3) \\
y_{gj} - L_{hgj} &\geq \epsilon(1 - u_{hgj}) && g,\ h \in \mathcal{G},\ j \in \mathcal{N}_g && (4) \\
\sum_{0, h \in \mathcal{G}} u_{hgj} &= 1 && g \in \mathcal{G},\ j \in \mathcal{N}_g && (5) \\
\sum_{j \in \mathcal{N}_g} u_{hgj} &\leq \lfloor \alpha_{hg} n_g \rfloor && h,\ g \in \mathcal{G},\ h \neq g && (6)
\end{aligned}$$

$$-\infty < L_{hgj} < \infty,\ y_{gj} \geq 0,\ \lambda_{ih} \geq 0,\ u_{hgj} \in \{0, 1\}$$

Note that the second formulation is not as precise as the first formulation due to the introduction of the ϵ constants (4). The relative difference in modified posterior probability between groups to which an entity is allocated and groups to which it is not allocated is bounded below by ϵ. In other words, $u_{hgj} = 0$ implies $y_{gj} \geq L_{hgj} + \epsilon$ which is an approximation of $y_{gj} > L_{hgj}$.

These ϵ constants, though a source of imprecision, are at the same time the source of stability of the classification rule derived by the DAMIP (see Section 4.2). Adjusting the ϵ values will produce classification rules that are robust in the sense that there are buffers between regions of likelihood function values that define the groups. Adding a set of constraints $y_{gj} \geq \epsilon u_{hgj}$ for each h, g, j further enhances the stability of solutions produced by the DAMIP by placing a buffer between the reserved judgment region and the regions corresponding to the groups.

The nonlinearity of the mixed-integer program can be removed by simply removing the *max constraints* (1) because they are redundant to the formulation.

Proposition 3.1.1. *The max constraints (1) are redundant.*

Proof. Suppose that the constraints (1) are omitted and a solution $x = [L, y, \lambda, u]$ is obtained. Suppose further that one of the *max constraints* (1) is violated for some entity x^{gj} (the constraints can be violated for additional entities as well). Then $y > \max\{0, L_{hgj} : h = 1, \ldots, G\}$. The big M constraints (2) and (3) force $u_{hgj} = 0$ for $h = 0, 1, \ldots, G$. These values for the u_{hgj}'s violates the constraint (5) $u_{0gj} + \sum_{h \in \mathcal{G}} u_{hgj} = 1$. Therefore, the max constraints will never be violated by an otherwise feasible solution, and they are redundant. □

The objective function and constraints (2) and (4) ensure that $u_{h'gj} = 1$ if and only if $y_{gj} = L_{h'gj} = \max\{0, L_{hgj} : h = 1, \ldots, G\}$. Constraints (2) and (4) also dictate that $u_{hgj} = 0$ if and only if $y_{gj} \geq L_{hgj} + \epsilon$. The objective and constraint (3) force the condition $u_{0gj} = 1$ if and only if $y_{gj} = 0$ and $L_{hgj} < 0$. Constraint (5) forces the classification of each entity to exactly one group.

The constraints (5) $u_{0gj} + \sum_{h \in \mathcal{G}} u_{hgj} = 1$ provide that an observation be allocated to exactly one group. The reserved judgment region is dedicated to observations for which no information about group membership is given by the L_{hgj}'s (i.e., $L_{hgj} < 0 \; \forall \; h$).

The constraints (6) are the misclassification constraints. Limits can be placed on the number of misclassified training entities with the intention of reducing the probability of misclassification of test observations. In general, tighter limits will encourage the placement of more entities in the reserved judgment region and reduces the number of correctly classified entities.

The removal of the nonlinearity and stability considerations lead us to the following model which will be the subject of the remainder of this work.

$$\text{maximize} \sum_{g\in\mathcal{G}} \sum_{j\in\mathcal{N}_g} u_{ggj}$$

subject to

$$\begin{aligned}
L_{hgj} &= \pi_h f_h(x^{gj}) - \sum_{\substack{i\in\mathcal{G}\\ i\neq h}} f_i(x^{gj})\lambda_{ih} && g,\ h\in\mathcal{G},\ j\in\mathcal{N}_g \\
y_{gj} - L_{hgj} &\leq M(1-u_{hgj}) && g,\ h\in\mathcal{G},\ j\in\mathcal{N}_g \\
y_{gj} &\leq M(1-u_{0gj}) && g\in\mathcal{G},\ j\in\mathcal{N}_g \\
y_{gj} &\geq \epsilon u_{hgj} && g,\ h\in\mathcal{G},\ j\in\mathcal{N}_g \\
y_{gj} - L_{hgj} &\geq \epsilon(1-u_{hgj}) && g,\ h\in\mathcal{G},\ j\in\mathcal{N}_g \\
\sum_{0,h\in\mathcal{G}} u_{hgj} &= 1 && g\in\mathcal{G},\ j\in\mathcal{N}_g \\
\sum_{j\in\mathcal{N}_g} u_{hgj} &\leq \lfloor \alpha_{hg} n_g \rfloor && h,\ g\in\mathcal{G},\ h\neq g
\end{aligned}$$

$$-\infty < L_{hgj} < \infty,\ y_{gj}\geq 0,\ \lambda_{ih}\geq 0,\ u_{hgj}\in\{0,1\}$$

The DAMIP is a process that involves estimations of the conditional group density function values $f_h(x^{gj})$ and subsequent solution of this mixed-integer program. Unless explicitly stated otherwise, the mixed-integer program will be referred to as the DAMIP for the remainder of this chapter. The mixed-integer program is solved for the training set only, so the correct classification of each observation is known. Observation x^{gj} refers to the j^{th} observation with correct classification to group g. The values $f_h(x^{gj})$ should be considered known constants in the mixed-integer program.

Note that the L_{hgj} variables can be substituted out (and we will assume that they are). The formulation contains the L_{hgj} for ease of reading only.

The constraints that contain M as coefficients will be referred to as *big-M constraints*; those with ϵ coefficients will be ϵ or *little* ϵ *constraints*; the constraints requiring the allocation of each observation to exactly one group will be the *required allocation constraints*; the constraints with α_{hg} coefficients will be *misclassification constraints*.

3.2 The complexity of DAMIP and related problems

3.2.1 LINEAR MAX SAT and its complexity

Previously, Johnson and Preparata [49] showed that CLOSED HEMISPHERE is $\mathcal{NP}$-complete. A statement of the problem is as follows.

CLOSED HEMISPHERE. Given a set of p linear inequalities

$$\{a^i x \leq 0 : i = 1, 2, \ldots, p, a^i \in \mathbb{Q}^d\}$$

and an integer r, does there exist a vector $x^* \in \mathbb{Q}^d$ satisfying r of the p inequalities?

Consider q sets of k linear functions in x with rational coefficients and right-hand sides. Suppose that in each set of functions, there is one function that corresponds to a *true* value and the other functions are *false*. Given values for x, a set of functions is *satisfied* if the *true* function has greater value than the other $k-1$ functions. Based on these definitions, consider the problem LINEAR MAX SAT is stated as follows.

LINEAR MAX SAT. Suppose q sets of k linear functions in x with rational data are given such that in each set there is one function that corresponds to a *true* value and the remaining functions are *false*. Does there exist a rational vector $x^* \in \mathbb{Q}^d$ such that s sets of linear functions are satisfied?

Proposition 3.2.1. *For a fixed value of k, LINEAR MAX SAT is $\mathcal{NP}$-complete.*

Proof. Given an instance of LINEAR MAX SAT and a vector $x \in \mathbb{Q}^d$, the vector can be verified by determining the values of the qk linear functions and counting the number of satisfied sets. Determining the values of the qk linear inequalities is equivalent to performing a matrix-vector multiplication, and is therefore polynomial in q and k. Determining if a set is satisfied takes $O(k-1)$ time which is the time to compare the *true* function value to the other $k-1$ function values. Counting the number of satisfied sets takes $O((k-1)q)$. The input is polynomial in q, k, and d. Thus, LINEAR MAX SAT can be verified in polynomial time, and is in $\mathcal{NP}$.

To show that LINEAR MAX SAT is $\mathcal{NP}$-hard, we reduce CLOSED HEMISPHERE to LINEAR MAX SAT. Suppose an instance of CLOSED HEMISPHERE is given with p linear

rational inequalities in d variables and an integer $r < p$. Each inequality will correspond to a set of k linear functions in the LINEAR MAX SAT problem. For each inequality, create $k-1$ linear functions such that the coefficients are distinct, positive multiples of the coefficients of the inequality and the constant term is 0. The zero function is the *true* function and the others have value *false* in each set.

A vector $x \in \mathbb{Q}^d$ satisfies r of the p rational linear inequalities if and only if the zero function is the maximum function in r of the p sets of linear functions. Therefore LINEAR MAX SAT $\leq_P$ CLOSED HEMISPHERE. □

Note that every variable x can be written as the difference of two nonnegative variables, so that it can be shown that LINEAR MAX SAT with nonnegative variables is also $\mathcal{NP}$-complete.

Also, by the same principle, LINEAR MAX SAT with functions having only positive coefficients is $\mathcal{NP}$-complete. As an illustration, suppose that function 1, $l_1(x) = \sum_{j=1}^{d} a_{1j}x_j + b_1$, is the *true* function and function 2, $l_2(x) = \sum_{j=1}^{d} a_{2j}x_j + b_2$, is a *false* function in the same set. The set of functions is satisfied only when $l_1(x) \geq l_2(x)$, or equivalently,

$$\sum_{j=1}^{d} a_{1j}x_j - \sum_{j=1}^{d} a_{2j}x_j \geq b_1 - b_2$$

$$\Rightarrow \sum_{j=1}^{d} (a_{1j} - a_{2j})x_j \geq b_1 - b_2$$

The differences $a_{1j} - a_{2j}$ and $b_1 - b_2$ can be rewritten in terms of differences between nonnegative numbers. The functions f_1 and f_2 can therefore be rewritten with nonnegative data. LINEAR MAX SAT with nonnegative variables and nonnegative data is also $\mathcal{NP}$-complete.

3.2.2 The complexity of DAMIP

DAMIP is a special case of LINEAR MAX SAT with nonnegative variables, nonnegative data, and a particular variable structure. A statement of the problem DAMIP is as follows.

DAMIP. Suppose N observations, each belonging to one of G groups; prior probabilities $\{\pi_h : h \in \mathcal{G}\}$; and conditional group density function values for each observation-group combination $\{f_h(x^{gj}) : h,\ g \in \mathcal{G}, j \in \mathcal{N}_g\}$ are given. Do there exist values for $\{\lambda_{ih} : i \in \mathcal{G}, h \in \mathcal{G}, i \neq h\}$ such that at least s observations are correctly classified according to the modified posterior probabilities?

For an observation x^{gj}, g is the group corresponding to a correct classification of the observation. With each observation-group combination, there is associated a modified posterior probability, which is a linear function in the λ_{ih}'s. The modified posterior probability for observation x^{gj} for allocation to group h is

$$L_{hgj} = \pi_h f_h(x^{gj}) - \sum_{i \neq h} f_i(x^{gj})\lambda_{ih}$$

An observation is allocated to the group h' for which $L_{h'gj} = \max\{L_{hgj} : h = 1, \ldots, G\}$. If all of the modified posterior probabilities are negative for an observation, that observation is not classified.

For each observation x^{gj}, the modified posterior probability function L_{ggj} is the analogue of the *true* function, and the functions L_{hgj}, $h \neq g$ are analogous to the *false* functions in a set of linear functions in LINEAR MAX SAT. The linear functions have a particular variable structure, as defined by the modified posterior probabilities. INDEPENDENT SET can be reduced to DAMIP to show that DAMIP is $\mathcal{NP}$-hard.

Proposition 3.2.2. *DAMIP is $\mathcal{NP}$-complete.*

Proof. Suppose an instance of DAMIP with G groups and N observations is given, along with values of the $G(G-1)$ λ_{ih} variables. The values of the NG modified posterior probability functions can be found with a matrix vector calculation and is therefore polynomial in N and G. Determining if an observation is given the correct classification (i.e., the appropriate modified posterior probability is larger than the others for that observation) takes $O(G-1)$ time. Counting the number of correctly classified observations takes $O(N(G-1))$ time. Therefore, DAMIP can be verified in polynomial time and is in $\mathcal{NP}$.

Suppose an instance of INDEPENDENT SET is given for a graph with $|V|$ nodes and

$|E|$ edges in an adjacency-list representation, and the problem is to determine if there exists an independent set of size s in the graph. The problem can be reduced to DAMIP with $G = |V| + 1$ groups and $N = |V|$ observations.

For each node in the graph $i = 1, 2, \ldots, |V|$, create an observation x^{i1} belonging to group i. Further, create one extra group z. If a node i is adjacent to no other nodes, then let $f_i(x^{i1}) = 1$ and $f_j(x^{i1}) = 0$ for $j \neq i$. Otherwise, if a node is adjacent to a set of nodes S, let $f_i(x^{i1}) = p_i$, $f_k(x^{i1}) = 2p_i$ for $k \in S$, $f_z(x^{i1}) = q$, and $f_j(x^{i1}) = 0$ for $j \notin S$ and $j \neq i$ where $p_i = \frac{1}{1+q+2|S|}$ so that

$$\sum_{j \in \mathcal{G}} f_j(x^{i1}) = f_i(x^{i1}) + \sum_{k \in S} f_k(x^{i1}) + f_z(x^{i1}) = 1$$

(Note that an appropriate value for q for the node with the highest degree will work for all other nodes.) Finally, set all of the prior probabilities π_h to $1/G$. The corresponding DAMIP problem is to determine if values exist for the $G(G-1) = (|V|+1)|V|$ λ_{ih} variables such that at least s of the observations are correctly classified according to the maximum posterior probabilities for each observation.

A solution to INDEPENDENT SET is a set of s or more nodes with no edges between them. The corresponding observations can be correctly classified in the DAMIP instance. Because of the way that the conditional probabilities are assigned, if two nodes are not adjacent to one another, then their simultaneous correct classification is always possible.

For example, suppose that nodes 1 and 2 are not adjacent to one another. Observation 11 is correctly classified if $L_{111} = \max\{L_{g11} : g = 1, 2, \ldots G\}$. For nodes g not adjacent to node 1, $L_{g11} \leq 0$ because $f_g(x^{11}) = 0$. To correctly classify observation 11, the λ_{ih} variables for nodes h adjacent to 1 should be increased until L_{111} is the maximum posterior probability. Increasing λ_{ih} variables for h adjacent to 1 only decreases the likelihood that observations are placed in group h. Because nodes 1 and 2 are not adjacent, increasing these λ_{ih} variables can only help the correct classification of observation x^{21}, because the likelihood of placing observation x^{21} in h decreases also. In short, the correct classification of observations x^{11} and x^{21} are independent of one another. Therefore, given an independent set of at least s nodes, there exist values for the λ_{ih} variables such that all observations are

correctly classified.

A solution to DAMIP is a set of values for the λ_{ih} variables such that at least s of the observations are correctly classified. The correctly classified observations correspond to an independent set of size s in the original graph.

Suppose, to the contrary, that two observations 11 and 21 are correctly classified and correspond to adjacent nodes in the graph. Then a necessary and sufficient condition for an edge in the conflict graph of the DAMIP (see Proposition 3.5.2, condition 1) reduces to

$$\pi_2 f_2(x^{21}) - \pi_1 f_1(x^{21}) - \pi_2 f_2(x^{11})\frac{f_a(x^{21})}{f_a(x^{11})} + \pi_1 f_1(x^{11})\frac{f_b(x^{21})}{f_b(x^{11})} < 0$$

where $\frac{f_a(x^{21})}{f_a(x^{11})} = \max_{g \neq 1} \frac{f_g(x^{21})}{f_g(x^{11})}$ and $\frac{f_b(x^{11})}{f_b(x^{21})} = \max_{g \neq 2} \frac{f_g(x^{11})}{f_g(x^{21})}$. By the manner in which the conditional probabilities were assigned, note that

$$\frac{f_a(x^{21})}{f_a(x^{11})} \geq \frac{f_z(x^{21})}{f_z(x^{11})} = 1$$

and

$$\frac{f_b(x^{11})}{f_b(x^{21})} \geq \frac{f_z(x^{11})}{f_z(x^{21})} = 1$$

Then the condition holds because

$$\begin{aligned}
&\pi_2 f_2(x^{21}) - \pi_1 f_1(x^{21}) - \pi_2 f_2(x^{11})\tfrac{f_a(x^{21})}{f_a(x^{11})} + \pi_1 f_1(x^{11})\tfrac{f_b(x^{21})}{f_b(x^{11})} \\
&\leq \quad (1/G)p_2 - (1/G)(2p_2) - (1/G)(2p_1) + (1/G)p_1 \\
&= \quad (1/G)(-p_1 - p_2) \\
&< \quad 0
\end{aligned}$$

Therefore, observations 11 and 21 cannot be simultaneously correctly classified, a contradiction. The set of correctly classified observations corresponds to an independent set on the original graph. □

DAMIP with misclassification limits is a problem that places limits on the number of observations that can be incorrectly allocated between each pair of groups. DAMIP can be reduced to DAMIP with misclassification limits by simply using limits of 100%. DAMIP

with misclassification limits is in $\mathcal{NP}$, as checking that the limits are satisfied takes linear time in the number of observations, which is added to the time needed to verify a potential solution to DAMIP. Therefore, DAMIP with misclassification limits is $\mathcal{NP}$-complete.

3.3 Dimension

The following proposition and corollary characterize the dimension of the DAMIP when the misclassification constraints are removed and certain restrictions are placed on the data.

Proposition 3.3.1. *Suppose that the following three conditions hold*

1. $M > \max_{g \in \mathcal{G}} \{ \max_{h \in \mathcal{G} \setminus g} \pi_h f_h(x^{gj}) - (G-2)\pi_g f_g(x^{gj}) \}$ *(see Proposition 3.6.3).*
2. *For every pair of entities* x^{mn} *and* x^{st} *and every pair of groups* a *and* b*,*
$$\frac{f_a(x^{mn})}{f_b(x^{mn})} \neq \frac{f_a(x^{st})}{f_b(x^{st})}$$
3. $f_h(x^{gj}) \neq 0$ *for all* h*,* g*, and* j

and the DAMIP has no misclassification constraints. Then there exists an ϵ *such that the equality set of the polyhedron defined by the constraints is comprised of the required allocation constraints*

$$\sum_{0, h \in \mathcal{G}} u_{hgj} = 1 \;\; g \in \mathcal{G}, \;\; j \in \mathcal{N}_g$$

Proof. Consider the null space. Let μ_{gj} be multipliers for the y_{gj}'s, let η_{ih} be multipliers for the λ_{ih}'s, and let γ_{hgj} be multipliers for the u_{hgj}'s. Then for a constant c, the null space is the set of multipliers such that

$$\sum_{gj} \mu_{gj} y_{gj} + \sum_{ih} \eta_{ih} \lambda_{ih} + \sum_{hgj} \gamma_{hgj} u_{hgj} + c = 0 \;\; (1)$$

for all feasible points. Consider the following feasible point that places all observations in the reserved judgment region.

$$y_{gj} = 0 \;\forall\; g, \;\; j; \;\; \lambda_{ih} = \max_{gj} \frac{\pi_h f_h(x^{gj})}{f_i(x^{gj})}; \;\; u_{0gj} = 1 \;\forall\; g, \;\; j; u_{hgj} = 0, \;\; h \neq 0 \;\; (a)$$

For this point,

$$\sum_{ih} \eta_{ih} \left(\max_{gj} \frac{\pi_h f_h(x^{gj})}{f_i(x^{gj})} \right) + \sum_{gj} \gamma_{0gj} u_{0gj} + c = 0 \quad (2)$$

For each λ_{ih} in turn, create new feasible points by adding a quantity $\delta > 0$ such that all entities are still placed in the reserved judgment region. For example, for λ_{21}, the new feasible point is

$$y_{gj} = 0 \ \forall \ g, \ j; \ \lambda_{21} = \max_{gj} \frac{\pi_1 f_1(x^{gj})}{f_2(x^{gj})} + \delta; \ \lambda_{ih} = \max_{gj} \frac{\pi_h f_h(x^{gj})}{f_i(x^{gj})}, \ ih \neq 21;$$

$$u_{0gj} = 1 \ \forall \ g, \ j; \ u_{hgj} = 0, \ h \neq 0$$

The point implies that

$$\eta_{21} \left(\max_{gj} \frac{\pi_h f_h(x^{gj})}{f_i(x^{gj})} + \delta \right) + \sum_{ih \neq 21} \eta_{ih} \max_{gj} \frac{\pi_h f_h(x^{gj})}{f_i(x^{gj})} + \sum_{gj} \gamma_{0gj} u_{0gj} + c = 0$$

The difference between this equation and (2) implies that

$$\delta \eta_{21} = 0 \Rightarrow \eta_{21} = 0$$

The same procedure can be repeated for each λ_{ih}, so that $\eta_{ih} = 0$ for all i, h. From the point (a), we have that

$$\sum_{gj} \gamma_{0gj} + c = 0$$

Consider again point (a). Create a new feasible point by setting $\lambda_{i1} = 0, \ i \neq 2$, and reducing λ_{21} until an entity, say x^{mn}, has $y_{mn} > 0$. Then, decrease λ_{21} by a small positive quantity such that $y_{gj} = 0$ for $gj \neq mn$ and all observations remain in the reserved judgment region. The difference in the points implies that $\mu_{mn} = 0$. Such values exist for λ_{21} because of the conditions on the data.

Continue to decrease λ_{21} in pairs of decrements such that the classification of observations does not change and the number of observations with $y_{gj} = 0$ remains constant. The differences in the pairs of points imply that $\mu_{gj} = 0$ for all observations x^{gj}.

Return again to point (a). Set $\lambda_{i1} = 0$, $i \neq 2$ and reduce λ_{21} until an entity is placed in group 1. Suppose the entity is x^{mn}. Then the new point is

$$y_{gj} = \zeta \ \forall \ g, \ j;$$

$$\lambda_{21} = \frac{\pi_1 f_1(x^{mn})}{f_2(x^{mn})} - \frac{\zeta}{f_2(x^{mn})}, \ \lambda_{i1} = 0, \ i \neq 2,$$

$$\lambda_{ih} = \min_{gj} \frac{\pi_h f_h(x^{gj})}{f_i(x^{gj})}, \ h \neq 1;$$

$$u_{1mn} = 1, \ u_{0mn} = 0$$

$$u_{0gj} = 1 \ gj \neq mn$$

$$u_{hgj} = 0, \ gj \neq mn \ (b)$$

for some $\zeta > \epsilon$. Because the μ_{gj}'s and η_{ih}'s are all 0 and $\sum_{gj} \gamma_{0gj} + c = 0$, this point implies that

$$\sum_{gj} \gamma_{0gj} - \gamma_{0mn} + \gamma_{1mn} + c = -\gamma_{0mn} + \gamma_{1mn} = 0$$

Now take point (b) and create a new feasible point by decreasing λ_{21} until a second entity is placed in group 1. Suppose the second entity is x^{st}. Then the new point has values

$$u_{1mn} = 1, \ u_{1st} = 1, \ u_{0mn} = 0, \ u_{0st} = 0$$

$$u_{0gj} = 1 \ gj \notin \{mn, \ st\}$$

$$u_{hgj} = 0, \ gj \notin \{mn, \ st\}$$

This new point implies

$$\sum_{gj} \gamma_{0gj} - \gamma_{0mn} - \gamma_{0st} + \gamma_{1mn} + \gamma_{1st} + c = -\gamma_{0st} + \gamma_{1st} = 0$$

Continue decreasing λ_{21} until each entity is placed in group 1 and modifying λ_{21} by a small factor to show that for each entity, $\gamma_{0gj} = \gamma_{1gj}$.

Return again to point (a) and create a new set of feasible points by setting $\lambda_{ih} = 0$, $i \neq 1$ and decreasing λ_{1h} until an entity is placed in group h. For each point in turn, decrease the λ_{1h}'s in the same way that λ_{21} was decreased in order to subsequently place entities in group h. Note that for all of these points to be feasible, ϵ must be chosen small

enough so that every observation can be allocated to every group in some feasible solution. A necessary and sufficient condition is to require $\epsilon \leq \min\{\pi_h f_h(x^{gj}) : h,\ g \in \mathcal{G}, j \in \mathcal{N}_g\}$.

Implied by these points are the following

$$\gamma_{0gj} = \gamma_{hgj},\ \forall\ h, g, j$$

These equations, coupled with the fact that $\sum_{gj} \gamma_{0gj} + c = 0$, imply that $\sum_{gj} \gamma_{hgj} + c = 0$ for all h. Each of these conditions on the multipliers are multiples of the required allocation constraints. Therefore, the required allocation constraints completely define the equality set. □

Corollary 3.3.2. *Suppose that the conditions of Proposition 3.3.1 hold. Then the dimension of the polytope defined by the constraints of the DAMIP is*

$$N + G(G-1) + (G+1)N - N = G(G-1) + N(G+1)$$

where N is the number of observations.

Proof. The DAMIP has N y_{gj} variables, $G(G-1)$ λ_{ih}, and $(G+1)N$ u_{hgj} variables. The equality set consists of the N required allocation constraints, which are clearly linearly independent. The dimension of the polytope is the dimension of the equality set subtracted from the number of variables. □

When the misclassification constraints are added to the DAMIP, the equality sets for the models can vary drastically depending on the input data. For example, the misclassification constraints could dictate that every entity be placed in the reserved judgment region. Some of the implications of the misclassification constraints are explored in Sections 3.4.2 and 3.4.3.

Let $R = \{(g, j) :\ u_{0gj} = 1$ in every feasible solution $\}$, and $S = \{(h, g, j)$: there exists no feasible solution with $u_{hgj} = 1\}$. Then the equality set can contain equalities of the following forms

$$
\begin{aligned}
u_{0gj} &= 1 \quad (g,j) \in R \\
y_{gj} &= 0 \quad (g,j) \in R \\
u_{hgj} &= 0 \quad h \in \mathcal{G},\ (g,j) \in R \\
u_{hgj} &= 0 \quad (h,g,j) \in S
\end{aligned}
$$

The equality set can also contain equalities of the form

$$
\sum_{(h,g,j) \in Q_1} u_{hgj} = \sum_{(h,g,j) \in Q_2} u_{hgj}
$$

for sets of entity-group combinations Q_1 and Q_2.

If the conditions for full-dimensionality given in Proposition 3.3.1 are not true, then the equality set can change. For example, if two entities x^{mn} and x^{st} have the same input data, then $u_{hmn} = u_{hst}$ for all h for those entities.

3.4 Finding the conflict graph and fixing variables

Consider the conflict graph for the DAMIP. The required allocation constraints provide that edges between nodes corresponding to u_{0gj} and u_{hgj} exist in the conflict graph for every group h and entity x^{gj}. For simplicity, let u_{hgj} represent both the integer variable in the IP formulation of the DAMIP and its corresponding node in the conflict graph. Let $\bar{u}_{hgj}$ be the node corresponding to the complement of u_{hgj}.

3.4.1 Generating the conflict graph

Consider two entities x^{mn} and x^{st} and groups a and b. Entity x^{mn} is placed in group a and x^{st} is placed in group b if and only if

$$
\begin{aligned}
L_{amn} - L_{hmn} &\geq 0 \quad \forall\ h \neq a \\
L_{bst} - L_{hst} &\geq 0 \quad \forall\ h \neq b \\
L_{amn} &\geq 0 \\
L_{bst} &\geq 0
\end{aligned}
$$

Therefore, $u_{amn} = u_{bst} = 1$ is infeasible if this system of $2G$ linear inequalities is infeasible. Written in terms of the λ_{ih}'s, the system becomes

$$\begin{aligned}
\sum_{i \in \mathcal{G} \setminus h} f_i(x^{mn})\lambda_{ih} - \sum_{i \in \mathcal{G} \setminus a} f_a(x^{mn})\lambda_{ia} &\geq \pi_h f_h(x^{mn}) - \pi_a f_a(x^{mn}) \quad \forall\ h \neq a \\
\sum_{i \in \mathcal{G} \setminus h} f_i(x^{st})\lambda_{ih} - \sum_{i \in \mathcal{G} \setminus b} f_b(x^{st})\lambda_{ib} &\geq \pi_h f_h(x^{st}) - \pi_b f_b(x^{st}) \quad \forall\ h \neq b \\
\sum_{i \in \mathcal{G} \setminus a} f_i(x^{mn})\lambda_{ia} &\leq \pi_a f_a(x^{mn}) \\
\sum_{i \in \mathcal{G} \setminus b} f_i(x^{st})\lambda_{ib} &\leq \pi_b f_b(x^{st})
\end{aligned}$$

where all of the λ_{ih}'s are restricted to greater than or equal to zero. Note that in the first two sets of inequalities, for $h \neq \{a, b\}$, the λ_{ih}'s can be increased arbitrarily to find values satisfying the inequalities in which they appear. Therefore, the only inequalities needed are those with only λ_{ia} and λ_{ib} variables. For any combination of 2 entities and 2 groups, an edge of the conflict graph can be derived by determining that the following system of 4 constraints and $2(G-1)$ variables is infeasible.

$$\begin{aligned}
\sum_{i \in \mathcal{G} \setminus b} f_i(x^{mn})\lambda_{ib} - \sum_{i \in \mathcal{G} \setminus a} f_a(x^{mn})\lambda_{ia} &\geq \pi_b f_b(x^{mn}) - \pi_a f_a(x^{mn}) \\
\sum_{i \in \mathcal{G} \setminus a} f_i(x^{st})\lambda_{ia} - \sum_{i \in \mathcal{G} \setminus b} f_b(x^{st})\lambda_{ib} &\geq \pi_a f_a(x^{st}) - \pi_b f_b(x^{st}) \\
\sum_{i \in \mathcal{G} \setminus a} f_i(x^{mn})\lambda_{ia} &\leq \pi_a f_a(x^{mn}) \\
\sum_{i \in \mathcal{G} \setminus b} f_i(x^{st})\lambda_{ib} &\leq \pi_b f_b(x^{st})
\end{aligned}$$

All λ_{ih}'s are restricted to be nonnegative. Similarly, for entities x^{mn} and x^{st} and a group a, edge (u_{amn}, u_{0st}) can be derived by determining that the following system is infeasible

$$\begin{aligned}
\sum_{i \in \mathcal{G} \setminus h} f_i(x^{mn})\lambda_{ih} - \sum_{i \in \mathcal{G} \setminus a} f_a(x^{mn})\lambda_{ia} &\geq \pi_h f_h(x^{mn}) - \pi_a f_a(x^{mn}) \quad \forall\ h \neq a \\
\sum_{i \in \mathcal{G} \setminus a} f_i(x^{mn})\lambda_{ia} &\leq \pi_a f_a(x^{mn}) \\
\sum_{i \in \mathcal{G} \setminus h} f_i(x^{st})\lambda_{ih} &> \pi_h f_h(x^{st}) \quad \forall\ h
\end{aligned}$$

which can be determined by considering the following system of 2 inequalities and $(G-1)$ variables.

$$\begin{aligned}
\sum_{i \in \mathcal{G} \setminus a} f_i(x^{mn})\lambda_{ia} &\leq \pi_a f_a(x^{mn}) \\
\sum_{i \in \mathcal{G} \setminus a} f_i(x^{st})\lambda_{ia} &> \pi_a f_a(x^{st})
\end{aligned}$$

Again, all λ_{ih}'s are required to be nonnegative. Note that these systems are actually conservative estimates when solving the DAMIP in practice. The sets of inequalities do not take advantage of the fact that the DAMIP actually gives classification rules with an ϵ buffer between groups. See Section 4.2 for more on the stability of solutions to the DAMIP. Methods for determining if these systems of inequalities are infeasible are developed further in the next section.

Note that the cases where $mn = st$ and/or $a = b$ do not need to be considered. The edges (u_{amn}, u_{bmn}) for $a \neq b$ are implied by the required allocation constraints, so that the case $mn = st$ does not need to be considered. For $a = b$, there exist non-negative values for the λ_{ih}'s such that two entities are placed in the same group, assuming that the misclassification constraints do not imply otherwise.

Aside from consideration of the misclassification constraints, the edges derived from the solution of the 2 entity 2 group systems is sufficient to derive all edges of the form (u_{amn}, u_{bst}) and (u_{amn}, u_{0st}) in the conflict graph for DAMIP. A feasible solution to the 2 entity 2 group solutions provides values for the λ_{ih}'s that can be extended to a feasible solution for the DAMIP because the λ_{ih}'s determine the values of all other variables.

3.4.2 Implications of other inequalities

Suppose that systems of inequalities corresponding to all possible pairs of entities and groups have been deemed feasible or infeasible. The required allocation constraints can be used to further describe the conflict graph. Specifically, they can be used to derive implications of the form $u_{amn} \leq u_{ast}$ or $u_{amn} + \bar{u}_{ast} \leq 1$.

Suppose that for entities x^{mn} and x^{st} and a group a, the edges (u_{amn}, u_{hst}) for all $h \in \{\mathcal{G} \backslash a, 0\}$ are in the conflict graph. Then, by the required allocation constraints, entity x^{st} is placed in group a whenever entity x^{mn} is placed in group a. Therefore, under such conditions, edge $(u_{amn}, \bar{u}_{ast})$ can be placed in the conflict graph.

3.4.3 Using the conflict graph to fix variables

Now suppose that all edges of the form $(u_{amn}, \bar{u}_{ast})$ have been derived. Suppose that for entity x^{mn} and group a, there are J edges in the conflict graph of the from $(u_{amn}, \bar{u}_{amj})$

so that if entity x^{mn} is placed in a, then J other entities are also placed in a. Due to the misclassification constraints, if $J \geq \alpha_{ma} n_a$, placing entity x^{mn} in group a is infeasible. Therefore, $u_{amn} = 0$ for all feasible solutions.

3.4.4 Using the conflict graph to solve the DAMIP

The inequalities implied by the conflict graph are a relaxation of the DAMIP. Valid inequalities for the conflict graph polytope are valid for the DAMIP. Let P^{CG} bet the conflict graph polytope.

Note that P^{CG} is a set packing polytope. If K is a maximal clique, the clique constraint $\sum_{(h,g,j):u_{hgj} \in K} u_{hgj} \leq 1$ is a facet for the convex hull of integer solutions in P^{CG} [73]. Maximal clique inequalities are also facets for the convex hull of integer solutions to the DAMIP without misclassification constraints, as shown in the following proposition.

Proposition 3.4.1. *A maximal clique inequality derived from the conflict graph is facet-defining for the full-dimensional (Proposition 3.3.1) DAMIP without misclassification constraints for sufficiently small ϵ.*

Proof. Let P be the polytope representing the DAMIP without misclassification constraints, and let $F = \{x \in P : \sum_{u_{hgj} \in K} u_{hgj} = 1\}$ for some maximal clique K in the conflict graph. Suppose for multipliers μ_{gj}, η_{ih}, and γ_{hgj} and a constant c that

$$\sum_{gj} \mu_{gj} y_{gj} + \sum_{ih} \eta_{ih} \lambda_{ih} + \sum_{hgj} \gamma_{hgj} u_{hgj} + c = 0 \ (1)$$

for all $x \in F$. Consider the following feasible point contained in F where all entities are placed in group a.

$$y_{gj} = \pi_a f_a(x^{gj});\ \lambda_{ia} = 0,\ \lambda_{ih} = \max_{gj} \frac{\pi_h f_h(x^{gj})}{f_i(x^{gj})} + \delta;\ u_{agj} = 1 \ (a)$$

where $u_{amn} \in K$. Note that such a point is feasible for F defined by any maximal clique K because, without misclassification constraints, there exist values for the λ_{ih}'s such that any two entities can be placed in the same group. Therefore, at most one entity in K has group a as its allocated group. The point (a) implies that

$$\sum_{gj} \pi_a f_a(x^{gj})\mu_{gj} + \left(\sum_{\substack{h \neq a \\ i \neq h}} \max_{gj} \frac{\pi_h f_h(x^{gj})}{f_i(x^{gj})} + \delta \right) \eta_{ih} + \sum_{gj} \gamma_{agj} + c = 0 \quad (2)$$

Now for each λ_{ih} in turn where $h \neq a$, subtract a quantity $\xi > 0$ such that all entities are still placed in group a. Such a quantity exists due to the condition on the $f_i(x^{gj})$'s in Proposition 3.3.1. The only difference between the new solution and (a) is the λ_{ih} with ξ subtracted. This fact, taken with equation (2), imply that $\xi\eta_{ih} = \eta_{ih} = 0$ for $h \neq a$ and $i \neq h$. Similarly, consider the feasible point contained in F where all entities are placed in group b and some entity in K has b as its allocated group. For each λ_{ih} where $h \neq b$, subtract a quantity ξ to create a point that still places all entities in group b. These points imply that $\xi\eta_{ih} = \eta_{ih} = 0$ for $h \neq b$ and $i \neq h$. Therefore, $\eta_{ih} = 0$ for all h and $i \neq h$.

Consider an entity x^{kl} and group $h \neq a$ such that $u_{hkl} = u_{amn} = 1$ holds in a feasible solution contained in F where $u_{amn} \in K$. If there are other entities x^{st} in such a solution with $u_{hst} = 1$, decrease the λ_{ia}'s and increase the λ_{ih}'s until only one entity is placed in group h. Call this entity x^{kl}. For observation x^{kl}, there is a unique hyperplane that separates the region where $u_{hkl} = 1$ and the region where $u_{akl} = 1$. The hyperplane is

$$L_{hkl} - L_{akl} = \pi_h f_h(x^{kl}) - \sum_{i \neq h} f_i(x^{kl})\lambda_{ih} - \pi_a f_a(x^{kl}) + \sum_{i \neq a} f_i(x^{kl})\lambda_{ia}$$

and x^{kl} is placed in group h if $L_{hkl} - L_{akl} \geq \epsilon$ and group a if $L_{hkl} - L_{akl} \leq -\epsilon$. The feasibility of a point where $u_{hkl} = u_{amn} = 1$ and a point where $u_{akl} = u_{amn} = 1$ may imply additional restrictions on ϵ. Let point (b) be a feasible point with $u_{hkl} = u_{amn} = 1$ sufficiently close to the separating hyperplane so that adjusting a λ_{ih} by ζ results in a feasible point with $u_{akl} = u_{amn} = 1$, $y_{kl} = \zeta f_i(x^{kl})$, and all other variable values remain constant. Let point (c) be a point achieved by adjusting a λ_{ih} by ζ. The difference in (b) and (c) imply

$$\zeta f_i(x^{kl})\mu_{kl} - \gamma_{hkl} + \gamma_{akl} = 0$$

Further increase the same λ_{ih} by $\delta > 0$ such that all entities maintain their current classification. The new point implies the following

$$(\delta + \zeta) f_i(x^{kl}) \mu_{kl} - \gamma_{hkl} + \gamma_{akl} = 0$$

The two equations together force $\mu_{kl} = 0$ and $\gamma_{hkl} = \gamma_{akl}$. Repeat the procedure for each entity x^{gj} and group h (including $h = 0$) such that $u_{hgj} = u_{amn} = 1$ holds in a feasible solution contained in F. Adjusting the appropriate λ_{ih}'s demonstrates that $\mu_{kl} = 0$ and $\gamma_{hgj} = \gamma_{agj}$ for entities x^{gj} and groups h where $u_{hgj} = u_{amn} = 1$ is possible.

Repeat the procedure for all entities x^{gj} and groups h where $u_{hgj} = u_{bst} = 1$ is possible where $u_{bst} \in K$. For every $u_{hgj} \notin K$, $\mu_{gj} = 0$ and $\gamma_{hgj} = \gamma_{agj}$. Note that because K is a maximal clique, for every node not in K, there exists a solution with $u_{hgj} = u_{bst} = 1$ for some $u_{bst} \in K$. This fact implies that $\mu_{gj} = 0$ for every node not in K.

Consider again point (a). Increase λ_{1a} by an amount $\sigma > 0$ such that the current classification of the observations does not change. The new point implies that

$$(\pi_a f_a(x^{mn}) - \sigma f_1(x^{mn}))\mu_{mn} + \sum_{gj} \gamma_{agj} + c = 0$$

The difference between this equation and point (a) requires that $\mu_{mn} = 0$. A similar argument shows that $\mu_{bst} = 0$ for all $u_{bst} \in K$. For each $u_{bst} \in K$, $\sum_{gj} \gamma_{bgj} + c = 0$.

Consider two nodes u_{amn} and u_{bst} that are in K. Adjust the λ_{ih}'s until a point is obtained such that $u_{amn} = 1$ and such that adjusting a λ_{ia} or λ_{ib} slightly forces $u_{amn} = 0 = 1 - u_{bst}$. The difference between this point and one obtained after the slight adjustment implies that $\gamma_{amn} + \gamma_{ast} = \gamma_{bmn} + \gamma_{bst}$. Repeat this procedure for every pair of nodes u_{amn} and u_{bst} in K, so that the equality holds for all pairs.

Now consider an entity and group $h \neq a$ such that $u_{hxy} + u_{amn} \leq 1$ for all feasible solutions contained in F. Construct a point such that $u_{hxy} = 1 = 1 - u_{amn}$ and such that adjusting a λ_{ia} or λ_{ih} slightly forces $u_{hxy} = 0 = 1 - u_{amn}$. If in the adjustment the classification of another entity not in K changes, relabel u_{hxy}. Note that for the point to remain feasible through the adjustment, there must exist $u_{bst} \in K$ such that $u_{bst} = 1$ for the initial point and $u_{bst} = 0$ in the final point. With the choice of λ_{ih}'s adjusted, $u_{ast} = 1$ at the final point. The difference between the initial and final points implies that

$$\gamma_{hxy} + \gamma_{bst} + \gamma_{hmn} = \gamma_{axy} + \gamma_{ast} + \gamma_{amn}$$

$$\Rightarrow \gamma_{hxy} = \gamma_{axy}$$

Repeat this procedure for all entities x^{gj} and groups h (including $h = 0$) where $u_{hgj} + u_{bst} \leq 1$ is true for $u_{bst} \in K$. For all group-entity combinations not in K, $\gamma_{hgj} = \gamma_{agj}$. Similarly, $\gamma_{0gj} = \gamma_{agj}$ for all such nodes not in K.

To summarize, $\eta_{ih} = 0$ for all i and h where $i \neq h$; $\mu_{gj} = 0$ for all gj; and $\gamma_{hxy} = \gamma_{axy}$ where $u_{amn} \in K$, $h \neq a$, and $xy \neq mn$. For each $u_{bst} \in K$, $\sum_{gj} \gamma_{bgj} = -c$. For u_{amn}, $u_{bst} \in K$, $\gamma_{amn} - \gamma_{bmn} = \gamma_{bst} - \gamma_{ast}$.

Let $\alpha = \gamma_{amn} - \gamma_{bmn}$ for some $u_{amn} \in K$. Let β be a vector of length N with elements β_{gj} corresponding to each observation and with values

$$\beta_{gj} = \gamma_{agj} \text{ for } gj \neq mn$$

$$\beta_{mn} = \gamma_{bmn}$$

where $u_{amn},\ u_{bst} \in K$.

Recall that the equality set of the DAMIP is the required allocation constraints. If F is defined by $\pi x = \pi_0$, and the equality set of the DAMIP is $A^{=}x = b^{=}$ then $\alpha\pi + \beta A^{=}$ is a vector with each element corresponding to a variable of the DAMIP. The values of $\alpha\pi + \beta A^{=}$ are zero for the elements corresponding to the y_{gj} variables and the λ_{ih} variables because they have zero coefficients in both π and $A^{=}$. The value of the element corresponding to the u_{hgj} variables are

$$\alpha + \beta_{gj} = \gamma_{amn} - \gamma_{bmn} + \gamma_{bmn} = \gamma_{amn} \text{ for all } u_{amn} \in K$$

$$\beta_{gj} = \gamma_{agj} = \gamma_{hgj} \text{ for } gj \text{ with } u_{agj} \notin K, \text{ for all } h$$

so that $\alpha\pi + \beta A^{=}$ is equal to the vector of multipliers defining the null space of F in equation

(1). Also,

$$\begin{aligned}
\alpha\pi_0 + \beta b^{=} &= \alpha + \sum_{gj} \beta_{gj} \\
&= \alpha + \beta_{mn} + \sum_{gj \neq mn} \beta_{gj} \\
&= \gamma_{amn} - \gamma_{bmn} + \gamma_{bmn} + \sum_{gj \neq mn} \beta_{gj} \\
&= \gamma_{amn} + \sum_{gj \neq mn} \gamma_{agj} \\
&= \sum_{gj} \gamma_{agj} \\
&= -c
\end{aligned}$$

where $u_{amn} \in K$. Thus, the choice of α and β demonstrate that the multipliers and constant in (1) are linear combinations of the facet-defining equality and the equality set for the DAMIP. By Theorem 3.6 on page 91 of Nemhauser and Wolsey [68], F is a facet of P. □

If the conflict graph is perfect, then the maximal clique constraints contain the facets for P^{CG}. Even if the graph is perfect, enumerating all of the maximal cliques can require an exponential number of operations in terms of the size of a graph.

In general, the conflict graph for the DAMIP is not perfect. Consider the following input for a 3 group problem with entities x^{11}, x^{21}, and x^{31} and groups 1, 2, and 3

$$\pi_1 = 7/20,\ \pi_2 = 7/20,\ \pi_3 = 3/10$$

$$\begin{aligned}
f_1(x^{11}) = 1/2,\ f_2(x^{11}) = 9/20,\ f_3(x^{11}) = 1/20 \\
f_1(x^{21}) = 2/5,\ f_2(x^{21}) = 11/20,\ f_3(x^{21}) = 1/20 \\
f_1(x^{31}) = 14/25,\ f_2(x^{31}) = 2/5,\ f_3(x^{31}) = 1/25
\end{aligned}$$

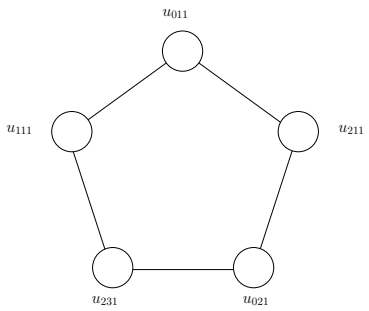


Figure 6: A conflict graph with an odd hole.

The data generate the conflict graph in Figure 6 which contains an odd hole. (Assume that the misclassification constraints do not imply any other edges or that any variables be fixed.) Graphs containing odd holes are not perfect, so the maximal clique constraints are not sufficient to describe the convex hull of integer feasible solutions of P. Odd hole inequalities are valid, but not necessarily facet-defining for the convex hull of integer solutions. If H is the set of nodes of an odd hole on the conflict graph, then $\sum_{(h,g,j)\in H} u_{hgj} \leq \frac{|H|-1}{2}$ is a valid inequality. These inequalities can be strengthened by lifting.

3.5 Finding the conflict hypergraph

The nodes of a conflict hypergraph are the same as those of the conflict graph. For an n-hypergraph, define an edge to be a subset of the nodes of size n. An edge E of the conflict n-hypergraph corresponds to an independent set constraint of the form

$$\sum_{(h,g,j):u_{hgj}\in H} u_{hgj} \leq n-1$$

Thus, the conflict graph is equivalent to the conflict 2-hypergraph. The results of this section are generalizations and extensions of the results for conflict graphs. The conflict n-hypergraph can contain structures such as maximal cliques, odd holes, and webs that are defined in a manner analogous to structures of the conflict graph []. These structures correspond to valid inequalities for the original problem. As an example, consider the maximal hyperclique inequality derived by Easton et. al []. A hyperclique $K_{m,n}$ in an n-hypergraph is a set of m vertices such that the induced subhypergraph of $K_{m,n}$ contains all $\binom{m}{n}$ edges. For a maximal hyperclique $K_{m,n}$, the following is a facet of the conflict hypergraph, provided that none of the edges between the nodes of the 2-, $\ldots$, $n-1$-hypergraphs are present.

$$\sum_{(h,g,j):u_{hgj}\in K_{m,n}} u_{hgj} \leq n-1$$

(Note that any edge of the conflict $(n-1)$-hypergraph can be extended to an edge of the conflict n-graph by adding any node not in the edge of the conflict $(n-1)$-hypergraph.)

These maximal hyperclique inequalities are not implied by the maximal clique inequalities of the conflict graph under certain conditions [28]. They contain more variables than

the maximal clique inequalities, and can be stronger inequalities than those derived from the conflict graph.

3.5.1 Necessary conditions for edges in the hypergraph

Consider generating the n-hypergraph of the DAMIP. If the misclassification constraints are removed, there exist values for λ_{ih} such that any two entities can be placed in the same group. For the remainder of the section, the misclassification constraints are removed from consideration. For simplicity of notation, let x^j refer to an observation, and let h_j be the group to which j is potentially assigned (for now, the group to which entity x^j belongs is not needed). Also, let $u_{h_j j} = 1$ if observation j is allocated to h_j and 0 otherwise. Let $u_{h_j j}$ represent the corresponding node in the conflict hypergraph. In a fashion similar to the generation of the conflict graph, for n entities x^j and nodes $u_{h_j j}$, a hyperedge in the conflict n-hypergraph exists between the nodes if and only if the following system is infeasible

$$\begin{aligned} \sum_{i \neq g} f_i(x^j)\lambda_{ig} - \sum_{i \neq h_j} f_i(x^j)\lambda_{ih_j} &\geq \pi_g f_g(x^j) - \pi_{h_j} f_{h_j}(x^j) \quad \forall j,\ g \neq h_j \quad (1) \\ \sum_{i \neq h_j} f_i(x^j)\lambda_{ih_j} &\leq \pi_{h_j} f_{h_j}(x^j) \qquad\qquad \forall j \qquad\qquad (2) \end{aligned}$$

The above system is actually over-constrained; the inequalities (1) that are needed are the g's for which u_{gk} is a node in the potential hyperedge for some entity x^k where $k \neq j$. If one of the nodes under consideration for the hyperedge corresponds to a reserved judgment variable u_{0j}, then let $h_j = 0$ and $\lambda_{i0} = 0$ and $f_0(x^j) = 0$. The second constraint (2) is not present for nodes/variables corresponding to the allocation of an observation to the reserved judgment region. Therefore, a hyperedge exists if and only if the following system is infeasible

$$\begin{aligned} \sum_{i \neq h_k} f_i(x^j)\lambda_{ih_k} - \sum_{i \neq h_j} f_i(x^j)\lambda_{ih_j} &\geq \pi_{h_k} f_{h_k}(x^j) - \pi_{h_j} f_{h_j}(x^j) \quad \forall j,\ h_k \neq h_j,\ k \neq j \quad (1) \\ \sum_{i \neq h_j} f_i(x^j)\lambda_{ih_j} &\leq \pi_{h_j} f_{h_j}(x^j) \qquad\qquad \forall j : h_j \neq 0 \qquad\qquad (2) \end{aligned}$$

The following two propositions give necessary conditions for the system of inequalities to be infeasible.

Proposition 3.5.1. *Consider the system of inequalities involved in deriving a potential edge of a hypergraph, where no reserved judgment nodes are under consideration for the edge. If the system is infeasible, then* $\pi_{h_k} f_{h_k}(x^j) > \pi_{h_j} f_{h_j}(x^j)$ *for some entity* x^j *and some entity* x^k *where* $k \neq j$*, and* $h_k \neq h_j$*.*

Proof. Consider the linear program formed by the system of inequalities for entities x^j and groups h_j

$$\text{maximize} \sum_{\substack{i,h \in G \\ i \neq h}} 0\lambda_{ih}$$

subject to

$$\begin{aligned}
\sum_{i \neq h_k} f_i(x^j)\lambda_{ih_k} - \sum_{i \neq h_j} f_i(x^j)\lambda_{ih_j} &\geq \pi_{h_k} f_{h_k}(x^j) - \pi_{h_j} f_{h_j}(x^j) \quad \forall\ j,\ h_k \neq h_j\ k \neq j \quad (1) \\
\sum_{i \neq h_j} f_i(x^j)\lambda_{ih_j} &\leq \pi_{h_j} f_{h_j}(x^j) \quad \forall\ j \quad (2) \\
\lambda_{ih_j} &\geq 0 \quad \forall\ h_j,\ i \neq h_j
\end{aligned}$$

Note that indices for entities x^j and x^k are for nodes under consideration only. Let η_{gj} and γ_j be the dual variables for constraint sets (1) and (2), respectively. The dual of the linear program is

$$\text{minimize} \sum_j \sum_{g \neq h_j} (\pi_g f_g(x^j) - \pi_{h_j} f_{h_j}(x^j))\eta_{gj} + \sum_j \pi_{h_j} f_{h_j}(x^j)\gamma_j$$

subject to

$$-\sum_{j:h_j=h} \sum_{h_k \neq h} f_i(x^j)\eta_{h_k j} + \sum_{j:h_j \neq h} f_i(x^j)\eta_{hj} + \sum_{j:h_j=h} f_i(x^j)\gamma_j \geq 0 \forall\ i,\ h \neq i$$

$$\eta_{gj} \leq 0,\ \gamma_j \geq 0$$

Negating the η_{gj}'s so that all variables are nonnegative gives

$$\text{minimize} \sum_j \sum_{g \neq h_j} (\pi_{h_j} f_{h_j}(x^j) - \pi_g f_g(x^j))\eta_{gj} + \sum_j \pi_{h_j} f_{h_j}(x^j)\gamma_j$$

subject to

$$\sum_{j:h_j=h} \sum_{h_k \neq h} f_i(x^j)\eta_{h_k j} - \sum_{j:h_j \neq h} f_i(x^j)\eta_{hj} + \sum_{j:h_j=h} f_i(x^j)\gamma_j \geq 0\ \forall\ i,\ h \neq i$$

$$\eta_{gj} \geq 0,\ \gamma_j \geq 0$$

The dual is always feasible because all constraints are satisfied when the dual variables are set to zero. Therefore, the primal is infeasible if and only if the dual is unbounded. The necessary condition derives immediately from the objective function of the dual, which requires that at least one coefficient of the η_{gj}'s in the dual is negative for the dual to be unbounded. □

Proposition 3.5.2. *Consider the system of inequalities involved in deriving a potential edge of a hypergraph where each entity under consideration belongs to a unique group. If the system is infeasible, then* $f_{i_{kj}}(x^j) f_{h_k}(x^k) - f_{i_{kj}}(x^k) f_{h_k}(x^j) < 0$ *for some node* $u_{h_k k}$ *and some entity* x^j *under consideration, where* i_{kj} *is chosen such that* $\frac{f_{i_{kj}}(x^j)}{f_{i_{kj}}(x^k)} = \max_{i \neq h_k} \frac{f_i(x^j)}{f_i(x^k)}$.

Proof. Consider the following linear program formed by the system of inequalities for entities x^j and groups h_j

$$\text{maximize} \sum_{\substack{i,h \in G \\ i \neq h}} 0\lambda_{ih}$$

subject to

$$\begin{aligned}
\sum_{i \neq h_k} f_i(x^j)\lambda_{ih_k} - \sum_{i \neq h_j} f_i(x^j)\lambda_{ih_j} &\geq \pi_{h_k} f_{h_k}(x^j) - \pi_{h_j} f_{h_j}(x^j) \quad \forall\ j,\ h_k \neq h_j\ k \neq j \quad (1) \\
\sum_{i \neq h_j} f_i(x^j)\lambda_{ih_j} &\leq \pi_{h_j} f_{h_j}(x^j) \qquad\qquad \forall\ j : h_j \neq 0 \quad (2) \\
\lambda_{ih_j} &\geq 0 \qquad\qquad \forall\ h_j,\ i \neq h_j
\end{aligned}$$

Note that indices for entities x^j and x^k are for nodes under consideration only. Let η_{gj} and γ_j be the dual variables for constraint sets (1) and (2), respectively. The dual of the linear program is

$$\text{minimize} \sum_{j:h_j \neq 0} \sum_{g \neq h_j} (\pi_g f_g(x^j) - \pi_{h_j} f_{h_j}(x^j))\eta_{gj} + \sum_{j:h_j=0} \sum_{g} \pi_g f_g(x^j)\eta_{gj} + \sum_{j:h_j \neq 0} \pi_{h_j} f_{h_j}(x^j)\gamma_j$$

subject to

$$\sum_{k \neq j} f_i(x^k)\eta_{h_j k} - \sum_{g \neq h_j} f_i(x^j)\eta_{gj} + f_i(x^j)\gamma_j \geq 0 \quad \forall\ j,\ i \neq h_j$$

$$\eta_{gj} \leq 0,\ \gamma_j \geq 0$$

Negating the η_{gj}'s so that all variables are nonnegative gives

$$\text{minimize} \sum_{j} \sum_{g \neq h_j} (\pi_{h_j} f_{h_j}(x^j) - \pi_g f_g(x^j))\eta_{gj} - \sum_{j:h_j=0} \sum_{g} \pi_g f_g(x^j)\eta_{gj} + \sum_{j} \pi_{h_j} f_{h_j}(x^j)\gamma_j$$

subject to

$$-\sum_{k \neq j} f_i(x^k)\eta_{h_j k} + \sum_{g \neq h_j} f_i(x^j)\eta_{gj} + f_i(x^j)\gamma_j \geq 0 \quad \forall\ j,\ i \neq h_j$$

$$\eta_{gj} \geq 0,\ \gamma_j \geq 0$$

The dual is always feasible because all constraints are satisfied when the dual variables are set to zero. Therefore, the primal is infeasible if and only if the dual is unbounded. For each entity x^j with $h_j \neq 0$, and $i \neq h_j$,

$$\gamma_j \geq \sum_{k \neq j} \frac{f_i(x^k)}{f_i(x^j)} \eta_{h_j k} - \sum_{g \neq h_j} \eta_{gj}$$

in a feasible solution. At any feasible point, the objective function can be decreased by setting the inequalities to equality above for all j and some $i \neq h_j$. Therefore, we may assume that the right-hand side is a formula for γ_j and $i \neq h_j$ such that the right-hand side is maximized. The objective function becomes

$$\begin{aligned}
&\sum_{j:h_j\neq 0}\sum_{g\neq h_j}(\pi_{h_j}f_{h_j}(x^j)-\pi_g f_g(x^j))\eta_{gj}-\sum_{j:h_j=0}\sum_g \pi_g f_g(x^j)\eta_{gj}\\
&\qquad\qquad +\sum_{j:h_j\neq 0}\pi_{h_j}f_{h_j}(x^j)\left(\sum_{k\neq j}\tfrac{f_i(x^k)}{f_i(x^j)}\eta_{h_jk}-\sum_{g\neq h_j}\eta_{gj}\right)\\
=\;& -\sum_{j:h_j\neq 0}\sum_{g\neq h_j}\pi_g f_g(x^j)\eta_{gj}-\sum_{j:h_j=0}\sum_g \pi_g f_g(x^j)\eta_{gj}+\sum_{j:h_j\neq 0}\pi_{h_j}f_{h_j}(x^j)\sum_{k\neq j}\tfrac{f_i(x^k)}{f_i(x^j)}\eta_{h_jk}\\
=\;& -\sum_{j:h_j\neq 0}\sum_{g\neq h_j}\pi_g f_g(x^j)\eta_{gj}-\sum_{j:h_j=0}\sum_g \pi_g f_g(x^j)\eta_{gj}+\sum_j\sum_{g\neq h_j}\pi_g f_g(x^k)\tfrac{f_i(x^j)}{f_i(x^k)}\eta_{gj}\\
=\;& \sum_j\sum_{g\neq h_j}\left(\pi_g f_g(x^k)\tfrac{f_i(x^j)}{f_i(x^k)}-\pi_g f_g(x^j)\right)\eta_{gj}
\end{aligned}$$

Note that in the first two expressions, $i \neq h_j$ and for the last two expressions, $i \neq h_k$ due to the change of variables. The dual is unbounded only if at least one of the coefficients on the η_{gj}'s is less than zero. The coefficient of η_{gj} is less than zero if $\frac{f_i(x^j)}{f_i(x^k)} < \frac{f_g(x^j)}{f_g(x^k)}$ for all $i \neq g$ which is equivalent to the condition as given in the proposition. □

Corollary 3.5.3. *If an edge (u_{hj}, u_{gk}) exists in the conflict graph, then*

1. $\pi_g f_g(x^j) > \pi_h f_h(x^j)$ *or* $\pi_h f_h(x^k) > \pi_g f_g(x^k)$, *and*
2. $f_{i_1}(x^k)f_h(x^j) - f_{i_1}(x^j)f_h(x^k) < 0$ *or* $f_{i_2}(x^j)f_g(x^k) - f_{i_2}(x^k)f_g(x^j) < 0$ *where* $\frac{f_{i_1}(x^k)}{f_{i_1}(x^j)} = \max_{i\neq h}\frac{f_i(x^k)}{f_i(x^j)}$ *and* $\frac{f_{i_2}(x^j)}{f_{i_2}(x^k)} = \max_{i\neq g}\frac{f_i(x^j)}{f_i(x^k)}$

The first necessary condition is intuitive, because if $\pi_{h_j}f_{h_j}(x^j) > \pi_g f_g(x^j)$ for every entity x^j, then $\lambda_{ih} = 0$ for all i, h is feasible. The second necessary condition is reminiscent of the necessary and sufficient condition of the 2-group problem of Proposition 2.0.1. Both conditions follow directly from the preceding propositions.

The next two propositions give necessary and sufficient conditions for the system of inequalities to be infeasible when generating the conflict graph.

3.5.2 Necessary and sufficient conditions for edges in the conflict graph

Proposition 3.5.4. *An edge (u_{hj}, u_{gk}) is in the conflict graph if and only if*

1. $\pi_g f_g(x^k) - \pi_h f_h(x^k) - \pi_g f_g(x^j)\frac{f_{i_2}(x^k)}{f_{i_2}(x^j)} + \pi_h f_h(x^j)\frac{f_{i_1}(x^k)}{f_{i_1}(x^j)} < 0$, *or*

2. $\pi_h f_h(x^j) - \pi_g f_g(x^j) - \pi_h f_h(x^k)\frac{f_{i_1}(x^j)}{f_{i_1}(x^k)} + \pi_g f_g(x^k)\frac{f_{i_2}(x^j)}{f_{i_2}(x^k)} < 0$, *or*

3. *both 1. and 2. hold*

where i_1 and i_2 are such that $\frac{f_{i_1}(x^k)}{f_{i_1}(x^j)} = \max_{i \neq h} \frac{f_i(x^k)}{f_i(x^j)}$ and $\frac{f_{i_2}(x^j)}{f_{i_2}(x^k)} = \max_{i \neq g} \frac{f_i(x^j)}{f_i(x^k)}$.

Proof. In Propositions 3.5.1 and 3.5.2, necessary conditions for the system of linear inequalities were given by showing that the dual of a corresponding linear program is unbounded. Consider the dual linear program that is encountered during the generation of an edge (u_{hj}, u_{gk}) of the conflict graph

$$\text{minimize } (\pi_h f_h(x^j) - \pi_g f_g(x^j))\eta_{gj} + (\pi_g f_g(x^k) - \pi_h f_h(x^k))\eta_{hk} + \pi_h f_h(x^j)\gamma_j + \pi_g f_g(x^k)\gamma_k$$

subject to

$$\begin{aligned} f_i(x^j)\eta_{gj} - f_i(x^k)\eta_{hk} + f_i(x^j)\gamma_j &\geq 0 \quad i \neq h \\ -f_i(x^j)\eta_{gj} + f_i(x^k)\eta_{hk} + f_i(x^k)\gamma_k &\geq 0 \quad i \neq g \\ \eta_{gj},\ \eta_{hk},\ \gamma_j,\ \gamma_k &\geq 0 \end{aligned}$$

From the two constraints,

$$\begin{aligned} \gamma_j &\geq -\eta_{gj} + \tfrac{f_i(x^k)}{f_i(x^j)}\eta_{hk} \quad i \neq h \\ \gamma_k &\geq \tfrac{f_i(x^j)}{f_i(x^k)}\eta_{gj} - \eta_{hk} \quad i \neq g \end{aligned}$$

Note that for an optimal basic feasible solution, at least one of the γ_j and one of the γ_k constraints hold at equality because otherwise the objective could be decreased by decreasing the values of γ_j or γ_k. Therefore, at an optimal solution, the γ's are either basic or fixed at their lower bound of zero. Also, when the linear program is unbounded, a γ_j constraint and a γ_k constraint can be made to hold at equality (or they can be zero) along an extreme ray.

($\Rightarrow$) Suppose condition 1 is true for a potential edge (u_{hj}, u_{gk}). The let $\eta_{gj} = \frac{f_{i_1}(x^k)}{f_{i_1}(x^j)}\eta_{hk}$, $\gamma_k = 0$, $\gamma_j = \frac{f_{i_2}(x^k)}{f_{i_2}(x^j)}\eta_{hk} - \eta_{gj} = (\frac{f_{i_2}(x^k)}{f_{i_2}(x^j)} - \frac{f_{i_1}(x^k)}{f_{i_1}(x^j)})\eta_{hk}$. Note that as γ_{gj}, η_{hk}, and η_{gj} are increased from zero, the constraints of the LP remain feasible.

The objective function along this ray is

$$
\begin{aligned}
&(\pi_h f_h(x^j) - \pi_g f_g(x^j))\eta_{gj} + (\pi_g f_g(x^k) - \pi_h f_h(x^k))\eta_{hk} + \pi_h f_h(x^j)\gamma_j + \pi_g f_g(x^k)\gamma_k \\
&= (\pi_h f_h(x^j) - \pi_g f_g(x^j))\tfrac{f_{i_1}(x^k)}{f_{i_1}(x^j)}\eta_{hk} + (\pi_g f_g(x^k) - \pi_h f_h(x^k))\eta_{hk} + \pi_h f_h(x^j)(\tfrac{f_{i_2}(x^k)}{f_{i_2}(x^j)} - \tfrac{f_{i_1}(x^k)}{f_{i_1}(x^j)})\eta_{hk} \\
&= (\pi_g f_g(x^k) - \pi_h f_h(x^k) - \pi_g f_g(x^j)\tfrac{f_{i_1}(x^k)}{f_{i_1}(x^j)} + \pi_h f_h(x^j)\tfrac{f_{i_2}(x^k)}{f_{i_2}(x^j)})\eta_{hk}
\end{aligned}
$$

Condition 1 dictates that the coefficient of η_{hk} is negative, so that as η_{hk} is increased along the ray, the objective function decreases. An analogous proof shows that if condition 2 holds, then the LP is unbounded.

($\Leftarrow$) Suppose that the LP is unbounded. Then by Corollary 3.5.3, $\frac{f_{i_1}(x^k)}{f_{i_1}(x^j)} < \frac{f_h(x^k)}{f_h(x^j)}$ or $\frac{f_{i_2}(x^j)}{f_{i_2}(x^k)} < \frac{f_g(x^j)}{f_g(x^k)}$, or both inequalities hold.

- Suppose that both $\frac{f_{i_1}(x^k)}{f_{i_1}(x^j)} < \frac{f_h(x^k)}{f_h(x^j)}$ and $\frac{f_{i_2}(x^j)}{f_{i_2}(x^k)} < \frac{f_g(x^j)}{f_g(x^k)}$. Then

$$
\begin{aligned}
&\pi_g f_g(x^k) - \pi_h f_h(x^k) - \pi_g f_g(x^j)\tfrac{f_{i_2}(x^k)}{f_{i_2}(x^j)} + \pi_h f_h(x^j)\tfrac{f_{i_1}(x^k)}{f_{i_1}(x^j)} \\
&= \pi_h(f_h(x^j) - f_h(x^k)\tfrac{f_{i_1}(x^j)}{f_{i_1}(x^k)}) + \pi_g(f_g(x^k)\tfrac{f_{i_2}(x^j)}{f_{i_2}(x^k)} - f_g(x^j) \\
&< \pi_h(f_h(x^k)\tfrac{f_{i_1}(x^j)}{f_{i_1}(x^k)} - f_h(x^k)\tfrac{f_{i_1}(x^j)}{f_{i_1}(x^k)}) + \pi_g(f_g(x^k)\tfrac{f_{i_2}(x^j)}{f_{i_2}(x^k)} - f_g(x^k)\tfrac{f_{i_2}(x^j)}{f_{i_2}(x^k)}) \\
&= 0
\end{aligned}
$$

  and condition 1 holds.

- Suppose that $\frac{f_{i_1}(x^k)}{f_{i_1}(x^j)} < \frac{f_h(x^k)}{f_h(x^j)}$, but $\frac{f_{i_2}(x^j)}{f_{i_2}(x^k)} > \frac{f_g(x^j)}{f_g(x^k)}$. Suppose also that the LP is unbounded along an extreme ray such that $\gamma_j = -\eta_{gj} + \frac{f_{i_1}(x^k)}{f_{i_1}(x^j)}\eta_{hk} \geq 0$ and $\gamma_k = \frac{f_{i_2}(x^j)}{f_{i_2}(x^k)}\eta_{gj} - \eta_{hk} \geq 0$ so that the γ_j and γ_k inequalities hold at equality for i_1 and i_2, respectively. Then the objective function along the extreme ray is

$$
\begin{aligned}
&(\pi_h f_h(x^j) - \pi_g f_g(x^j))\eta_{gj} + (\pi_g f_g(x^k) - \pi_h f_h(x^k))\eta_{hk} + \pi_h f_h(x^j)\gamma_j + \pi_g f_g(x^k)\gamma_k \\
&= (\pi_h f_h(x^j) - \pi_g f_g(x^j))\eta_{gj} + (\pi_g f_g(x^k) - \pi_h f_h(x^k))\eta_{hk} + \pi_h f_h(x^j)(-\eta_{gj} + \tfrac{f_{i_1}(x^k)}{f_{i_1}(x^j)}\eta_{hk}) \\
&\qquad + \pi_g f_g(x^k)(\tfrac{f_{i_2}(x^j)}{f_{i_2}(x^k)}\eta_{gj} - \eta_{hk}) \\
&= (-\pi_g f_g(x^j) + \pi_g f_g(x^k)\tfrac{f_{i_2}(x^j)}{f_{i_2}(x^k)})\eta_{gj} + (-\pi_h f_h(x^k) + \pi_h f_h(x^j)\tfrac{f_{i_1}(x^k)}{f_{i_1}(x^j)})\eta_{hk} \\
&\geq (-\pi_g f_g(x^j) + \pi_g f_g(x^k)\tfrac{f_{i_2}(x^j)}{f_{i_2}(x^k)})\tfrac{f_{i_2}(x^k)}{f_{i_2}(x^j)}\eta_k + (-\pi_h f_h(x^k) + \pi_h f_h(x^j)\tfrac{f_{i_1}(x^k)}{f_{i_1}(x^j)})\eta_{hk} \\
&= (\pi_g f_g(x^k) - \pi_h f_h(x^k) - \pi_g f_g(x^j)\tfrac{f_{i_2}(x^k)}{f_{i_2}(x^j)} + \pi_h f_h(x^j)\tfrac{f_{i_1}(x^k)}{f_{i_1}(x^j)})\eta_{hk}
\end{aligned}
$$

Because the objective is unbounded along the extreme ray, the right-hand side on the last line above must be unbounded, which is true only if condition 1 holds. A similar proof shows that condition 2 must hold if $\frac{f_{i_2}(x^j)}{f_{i_2}(x^k)} < \frac{f_g(x^j)}{f_g(x^k)}$, but $\frac{f_{i_1}(x^k)}{f_{i_1}(x^j)} > \frac{f_h(x^k)}{f_h x^j)}$.

- Suppose again that $\frac{f_{i_1}(x^k)}{f_{i_1}(x^j)} < \frac{f_h(x^k)}{f_h(x^j)}$, but $\frac{f_{i_2}(x^j)}{f_{i_2}(x^k)} > \frac{f_g(x^j)}{f_g(x^k)}$. Suppose also that $\gamma_k = 0 > \frac{f_{i_2}(x^j)}{f_{i_2}(x^k)}\eta_{gj} - \eta_{hk}$ and $\gamma_j = -\eta_{gj} + \frac{f_{i_1}(x^k)}{f_{i_1}(x^j)}\eta_{hk} \geq 0$. Then the objective along the extreme ray is

$$
\begin{aligned}
&(\pi_h f_h(x^j) - \pi_g f_g(x^j))\eta_{gj} + (\pi_g f_g(x^k) - \pi_h f_h(x^k))\eta_{hk} + \pi_h f_h(x^j)\gamma_j + \pi_g f_g(x^k)\gamma_k \\
&= (\pi_h f_h(x^j) - \pi_g f_g(x^j))\eta_{gj} + (\pi_g f_g(x^k) - \pi_h f_h(x^k))\eta_{hk} + \pi_h f_h(x^j)(-\eta_{gj} + \tfrac{f_{i_1}(x^k)}{f_{i_1}(x^j)}\eta_{hk}) \\
&= -\pi_g f_g(x^j)\eta_{gj} + (\pi_g f_g(x^k) - \pi_h f_h(x^k) + \pi_h f_h(x^j)\tfrac{f_{i_1}(x^k)}{f_{i_1}(x^j)})\eta_{hk} \\
&> -\pi_g f_g(x^j)\tfrac{f_{i_2}(x^k)}{f_{i_2}(x^j)}\eta_{hk} + (\pi_g f_g(x^k) - \pi_h f_h(x^k) + \pi_h f_h(x^j)\tfrac{f_{i_1}(x^k)}{f_{i_1}(x^j)})\eta_{hk} \\
&> (\pi_g f_g(x^k) - \pi_h f_h(x^k) - \pi_g f_g(x^j)\tfrac{f_{i_2}(x^k)}{f_{i_2}(x^j)} + \pi_h f_h(x^j)\tfrac{f_{i_1}(x^k)}{f_{i_1}(x^j)})\eta_{hk}
\end{aligned}
$$

Because the objective is unbounded along the extreme ray, the right-hand side on the last line above must also be unbounded, which is true only if condition 1 holds. A similar proof shows that condition 2 must hold if $\frac{f_{i_2}(x^j)}{f_{i_2}(x^k)} < \frac{f_g(x^j)}{f_g(x^k)}$, but $\frac{f_{i_1}(x^k)}{f_{i_1}(x^j)} > \frac{f_h(x^k)}{f_h(x^j)}$. Note that both $\gamma_k = 0 > \frac{f_{i_2}(x^j)}{f_{i_2}(x^k)}\eta_{gj} - \eta_{hk}$ and $\gamma_j = 0 > -\eta_{gj} + \frac{f_{i_1}(x^k)}{f_{i_1}(x^j)}\eta_{hk}$ cannot occur simultaneously. Therefore, if the LP is unbounded, then condition 1 or condition 2 holds.

□

The following proposition shows that the necessary condition in Proposition 3.5.2 is both necessary and sufficient for edges (u_{0j}, u_{gk}) in the conflict graph.

Proposition 3.5.5. *An edge (u_{0j}, u_{gk}) is in the conflict graph if and only if $f_{i_0}(x^j)f_g(x^k) - f_{i_0}(x^k)f_g(x^j) < 0$ for i_0 such that $\frac{f_{i_0}(x^j)}{f_{i_0}(x^k)} = \max_{i \neq a} \frac{f_i(x^j)}{f_i(x^k)}$.*

Proof. Edge (u_{0j}, u_{gk}) is in the conflict graph if and only if the following linear program is infeasible

$$\text{maximize} \sum_{i \neq g} 0\lambda_{ig}$$

subject to

$$\begin{aligned} \sum_{i \neq g} f_i(x^k)\lambda_{ig} &\leq \pi_g f_g(x^k) \quad (1) \\ \sum_{i \neq g} f_i(x^j)\lambda_{ig} &> \pi_g f_g(x^j) \quad (2) \\ \lambda_{ig} &\geq 0 \end{aligned}$$

Let γ be the dual variable for (1) and η the dual variable for (2). When the ϵ constraints are present for the reserved judgment region, the second inequality can be changed to a greater-than-or-equal-to constraint with no loss in the quality of the solution. The dual of the linear program is

$$\text{minimize } \pi_g f_g(x^k)\gamma - \pi_g f_g(x^j)\eta$$

subject to

$$\begin{aligned} f_i(x^k)\gamma + f_i(x^j)\eta &\geq 0 \quad i \neq a \\ \gamma,\ \eta &\geq 0 \end{aligned}$$

The dual is feasible because all constraints are satisfied at $\gamma = \eta = 0$. The primal is infeasible if and only if the dual is unbounded. For any feasible solution, $\gamma \geq \frac{f_i(x^j)}{f_i(x^k)}\eta$ for each $i \neq a$. Therefore, any feasible solution can be improved by setting $\gamma = \frac{f_{i_0}(x^j)}{f_{i_0}(x^k)}\eta$. The objective function becomes

$$\text{maximize } \pi_g f_g(x^k)\frac{f_{i_0}(x^j)}{f_{i_0}(x^k)}\eta - \pi_g f_g(x^j)\eta = (\pi_g f_g(x^k)\frac{f_g(x^j)}{f_g(x^k)} - \pi_g f_g(x^j))\eta$$

which is unbounded if $f_{i_0}(x^j)f_g(x^k) - f_g(x^j)f_{i_0}(x^k) < 0$. □

3.6 Upper bounds for M

The relative size of the M and ϵ constraints contributes to an ill-conditioned constraint matrix for the DAMIP. Ideally, M is chosen large enough so that the feasibility of potential solutions is not affected. These competing considerations lead us to seek the effects of different values for M.

The following proposition and lemma suggest a method for placing effective upper bounds on the λ_{ih}'s, which in turn can be used to derive upper bounds for M.

Proposition 3.6.1. *An entity x^{gj} is placed in the reserved judgment region if for each $h \in \mathcal{G}$, there exists a group a such that*

$$\lambda_{ah} > \frac{\pi_h f_h(x^{gj})}{f_a(x^{gj})}$$

The converse is true for the 2-group model.

Proof. An entity is placed in the reserved judgment region (i.e., $u_{0gj} = 1$) if and only if $L_{hgj} < 0$ for all h. Suppose entity x^{gj} has the property that for each $h \in \mathcal{G}$, there exists a a such that $\lambda_{ah} > \frac{\pi_h f_h(x^{gj})}{f_a(x^{gj})}$. Then, for any L_{hgj},

$$\begin{aligned} L_{hgj} &= \pi_h f_h(x^{gj}) - \sum_{i \in \mathcal{G} \setminus \{h\}} f_i(x^{gj}) \lambda_{ih} \\ &= \pi_h f_h(x^{gj}) - f_a(x^{gj}) \lambda_{ah} - \sum_{i \in \mathcal{G} \setminus \{h,a\}} f_i(x^{gj}) \lambda_{ih} \\ &< \pi_h f_h(x^{gj}) - f_a(x^{gj}) \frac{\pi_h f_h(x^{gj})}{f_a(x^{gj})} \\ &= 0 \end{aligned}$$

Note that for the two group model, the sum $\sum_{i \in \mathcal{G} \setminus \{h,a\}} f_i(x^{gj}) \lambda_{ih}$ is not present, so the condition is necessary and sufficient. □

Lemma 3.6.2. *For every pair of groups i, h where $i \neq h$, consider the following upper bound on the λ_{ih}'s:*

$$\lambda_{ih} \leq \max_{g \in \mathcal{G}, j \in \mathcal{N}_g} \frac{\pi_h f_h(x^{gj})}{f_i(x^{gj})} + \delta$$

The δ term can be made arbitrarily small. If this upper bound is enforced, then the set of integer feasible solutions is not reduced and no meaningful solutions are rendered infeasible.

Proof. Given $a,\ h,\ \lambda_{ah}$ is found only in the equations L_{hgj} for all $g \in \mathcal{G},\ j \in \mathcal{N}_g$. The modified posterior probabilities are of the form

$$L_{hgj} = \pi_h f_h(x^{gj}) - \sum_{i \in \mathcal{G} \setminus h} f_i(x^{gj})\lambda_{ih}\ g, h \in \mathcal{G},\ j \in \mathcal{N}_g$$

Therefore, the λ_{ih}'s can be bounded above by any value large enough to force the appropriate L_{hgj}'s to be negative. These values are valid upper bounds because the only positive term in the equation for the L_{hgj}'s is a constant. After a λ_{ih} forces a L_{hgj} negative, then that entity is either placed in the reserved judgment region or another group. For each entity x^{gj}, if

$$\lambda_{ah} > \max_{g \in \mathcal{G}, j \in \mathcal{N}_g} \frac{\pi_h f_h(x^{gj})}{f_a(x^{gj})}$$

then

$$\begin{aligned}
L_{hgj} &= \pi_h f_h(x^{gj}) - \sum_{i \in \mathcal{G} \setminus h} f_i(x^{gj})\lambda_{ih} \\
&= \pi_h f_h(x^{gj}) - f_a(x^{gj})\lambda_{ah} - \sum_{i \in \mathcal{G} \setminus \{h,a\}} f_i(x^{gj})\lambda_{ih} \\
&\leq \pi_h f_h(x^{gj}) - f_a(x^{gj})\lambda_{ah} \\
&< \pi_h f_h(x^{gj}) - f_a(x^{gj}) \max_{g \in \mathcal{G}, j \in \mathcal{N}_g} \tfrac{\pi_h f_h(x^{gj})}{f_i(x^{gj})} \\
&\leq \pi_h f_h(x^{gj}) - f_a(x^{gj}) \tfrac{\pi_h f_h(x^{gj})}{f_a(x^{gj})} \\
&= 0
\end{aligned}$$

The upper bound given for λ_{ah} allows for negative L_{hgj} values and is therefore valid. □

Proposition 3.6.3. *For an entity x^{gj}, an upper bound on M in the big-M constraint $y_{gj} \leq M(1 - u_{0gj})$ is*

$$\max_{h \in \mathcal{G}} \pi_h f_h(x^{gj})$$

which is equivalent to an upper bound on y_{gj}. For an entity x^{gj} and a group h, an upper bound on M in the big-M constraint $y_{gj} \leq M(1 - u_{agj})$ is

$$\max_{h \in \mathcal{G} \setminus a} \{\pi_h f_h(x^{gj})\} - \pi_a f_a(x^{gj}) + \sum_{i \in \mathcal{G} \setminus a} f_i(x^{gj})\lambda^*_{ia}$$

where

$$\lambda_{ia}^{*} = \max_{g \in \mathcal{G}, j \in \mathcal{N}_g} \frac{\pi_a f_a(x^{gj})}{f_i(x^{gj})} + \delta$$

for each i. The upper bound is equivalent to an upper bound on $y_{gj} - L_{agj}$.

Proof. The y_{gj}'s are defined as the maximum of the L_{hgj}'s which are Bayesian probabilities with nonnegative terms subtracted. Therefore, the maximum value that y_{gj} can attain is the maximum Bayesian probability, or $\max_{h \in \mathcal{G}} \pi_h f_h(x^{gj})$. Setting M in the constrain $y_{gj} \leq M(1 - u_{0gj})$ to this upper bound will enforce the restriction that if $y_{gj} > 0$, then $u_{0gj} = 0$. In other words, if an entity has a positive modified Bayesian probability, then it cannot be placed in the reserved judgment region.

Now consider $y_{gj} - L_{agj}$ for some a, m, and n.

$$\begin{aligned} y_{gj} - L_{agj} &= y_{gj} - \pi_a f_a(x^{gj}) + \sum_{i \neq a} f_i(x^{gj}) \lambda_{ia} \\ &\leq \max_{h \in \mathcal{G} \setminus a} \{\pi_h f_h(x^{gj})\} - \pi_a f_a(x^{gj}) + \sum_{i \neq a} f_i(x^{gj}) \lambda_{ia}^{*} \end{aligned}$$

The inequality is due to the the upper bound on y_{gj} from the first part of the proposition and from the upper bound on the λ_{ih}'s from Lemma 3.6.2. Note that the upper bound is nonnegative:

$$\begin{aligned} &\max_{h \in \mathcal{G} \setminus a} \{\pi_h f_h(x^{gj})\} - \pi_a f_a(x^{gj}) + \sum_{i \in \mathcal{G} \setminus a} f_i(x^{gj}) \lambda_{ia}^{*} \\ &\geq \max_{h \in \mathcal{G} \setminus a} \pi_h f_h(x^{gj}) - \pi_a f_a(x^{gj}) + \sum_{i \in \mathcal{G} \setminus a} f_i(x^{gj}) \max_{m \in \mathcal{G}, n \in \mathcal{N}_g} \frac{\pi_g f_a(x^{mn})}{f_i(x^{mn})} \\ &\geq \max_{h \in \mathcal{G} \setminus a} \pi_h f_h(x^{gj}) - \pi_a f_a(x^{gj}) + \sum_{i \in \mathcal{G} \setminus a} f_i(x^{gj}) \frac{\pi_a f_a(x^{gj})}{f_i(x^{gj})} \\ &= \max_{h \in \mathcal{G} \setminus a} \pi_h f_h(x^{gj}) + (G - 2) \pi_a f_a(x^{gj}) \\ &> 0 \end{aligned}$$

Therefore, the case that $y_{gj} = L_{hgj}$ is also included, and the upper bound is valid. □

In practice, the upper bound on the λ_{ih}'s from Lemma 3.6.2 can be extremely large due to extremely small values of $f_i(x^{gj})$. For example if π_h and $f_h(x^{gj})$ are on the order of 10^{-1} and $f_i(x^{gj})$ is on the order of 10^{-21}, then the upper bound for λ_{ih} is on the order of

10^{20}. These high upper bounds on the λ_{ih}'s can dominate the upper bound for M given in Proposition 3.6.3, especially for an entity with $f_i(x^{gj})$ on the order of 10^{-1}.

The λ_{ih}'s can be interpreted as a measure of the likelihood of placing entities in group i that would otherwise be placed in group h. Intuitively, if $f_i(x^{gj})$ is extremely small for an entity x^{gj}, then that entity would likely not be placed in group i in an optimal solution. Therefore, the upper bound for λ_{ih} derived from the data on an entity such as x^{gj} can be "safely" ignored in most cases. The safety of these assumptions can be quantified.

Suppose the conditional probability $f_i(x^{gj})$ that entity x^{gj} belongs to group i is less than or equal to $\frac{1}{kG}$ where k is a multiplying factor greater than 1. Suppose that for entities with $f_i(x^{gj}) > \frac{1}{kG}$, one wishes that λ_{ih} can assume a value that will render the corresponding L_{hgj} negative. In that case, such entities may be placed in group i, another group, or in the reserved judgment region. The upper bound for λ_{ih} is

$$\begin{aligned}
\lambda_{ih} &\leq \max_{g\in\mathcal{G},\ j\in\mathcal{N}_g} \{\tfrac{\pi_h f_h(x^{gj})}{f_i(x^{gj})} + \delta | f_i(x^{gj}) > \tfrac{1}{kG}\} \\
&\leq \max_{g\in\mathcal{G},\ j\in\mathcal{N}_g} kG\pi_h f_h(x^{gj}) + \delta \\
&\leq kG
\end{aligned}$$

for an appropriately chosen $\delta > 0$. If these upper bounds are placed on all of the λ_{ih}'s, then the upper bound on a big M variable can be

$$\begin{aligned}
y_{gj} - L_{hgj} &= y_{gj} - \pi_h f_h(x^{gj}) + \sum_{i\in\mathcal{G}\setminus h} \lambda_{ih} f_h(x^{gj}) \\
&\leq \max_{k\in\mathcal{G}} \pi_k f_k(x^{gj}) - \pi_h f_h(x^{gj}) + kG \sum_{i\in\mathcal{G}\setminus h} f_h(x^{gj}) \\
&\leq \max_{k\in\mathcal{G}} \pi_k f_k(x^{gj}) - \pi_h f_h(x^{gj}) + kG \\
&< (k+1)G
\end{aligned}$$

For example, if $G = 3$ and $k = 10$, then using a big $M = 33$, all entities with $f_i(x^{gj}) > 1/30$ can be placed in group i. Note that for entities with $f_i(x^{gj}) < 1/30$, there must exist at least one group for which

$$f_h(x^{gj}) > \frac{1}{2}\left(1 - \frac{1}{30}\right) = \frac{29}{60}$$

which is significantly larger than 1/30. In other words, eliminating the opportunity for such an observation x^{gj} to be moved from group h to group i is reasonable.

As k gets large, the lower bound for the maximum $f_h(x^{gj})$, $h \neq i$ approaches $\frac{1}{G-1}$:

$$f_h(x^{gj}) > \frac{1}{G-1}\left(1 - \frac{1}{kG}\right) = \frac{kG-1}{kG}\frac{1}{G-1}$$

Also, as k increases, the difference between $\frac{1}{G-1}$ and $\frac{1}{kG}$ increases. Therefore, entities with reasonably large values for $f_i(x^{gj})$ can potentially be placed in group i, and can be used to derive upper bounds on the big M's.

chapter 3

Chapter IV

Consistency, Robustness, and Stability of the DAMIP

4.1 Consistency of the DAMIP

The DAMIP involves estimation of the posterior probabilities $f(h|x)$, estimation of the conditional group density functions $f(x|h) = f_h(x)$, and subsequent determination of optimal values of λ_{ih} variables to define the modified posterior probabilities. Finding good quality estimates of the posterior probabilities $f(h|x)$ and conditional group density functions $f(x|h)$ is beyond the scope of this work. When investigating the consistency of the DAMIP, we make the strong (and unrealistic) assumption that the values for $f(h|x)$ and $f_h(x)$ (and therefore the prior probabilities π_h) are known. In other words, we assume that the data has a density that is known.

Anderson [1] characterized the optimal solution to the problem

$$\max_{\phi} P\{\phi(X) = Y\}$$

$$\text{subject to } P\{\phi(X) = g, Y = h\} \leq \alpha_{hg},\ g,\ h \in G,\ g \neq h$$

He showed that the optimal allocation is based on modified posterior "probabilities" of the form

$$L_h(x) = f(h|x) - \sum_{i \neq h} f_i(x)\lambda_{ih}$$

so that $\phi^{\dagger}(x) = \arg\max_{h \in \mathcal{G}}(f(h|x) - \sum_{i \neq h} f_i(x)\lambda_{ih}^{\dagger})$ is an *Anderson optimal solution*, characterized by optimally-chosen λ_{ih} values. For a classification problem with misclassification limits, we will consider a classification algorithm *consistent* if, as the sample size increases, $P\{\phi_n(X) = Y\}$ converges to $P\{\phi^{\dagger}(X) = Y\}$ and every sequence of classifiers produced by the method satisfies the misclassification constraints in the limit. This definition represents a generalization of the traditional definition where misclassification constraints were not considered.

Theorem 4.1.1. *Assuming that the conditional group density functions $f_h(x)$ and the prior probabilities π_h are known, the DAMIP is a strongly universally consistent classifier.*

Proof. To show that the DAMIP is a strongly universally consistent classifier, we will use VC Theory to show that the objective function converges uniformly to $P\{\phi^\dagger(X) = Y\}$ with n and then show that any sequence of optimal solutions of the DAMIP ϕ_n will converge uniformly to a function satisfying the constraints

$$P\{\phi(X) = g, Y = h\} \leq \alpha_{hg},\ g,\ h \in G,\ g \neq h$$

The modified posterior probabilities are linear functions in the λ_{ih}'s. One can now think of the posterior probabilities and conditional group density function values as input data. The modified posterior probabilities are a certain class of classifiers, with the objective to find the optimal λ_{ih}'s based on the input data. Given a sample of size n, the DAMIP solves the following problem

$$\max_{\phi} \sum_{j=1}^{n} I_{\{\phi(X_j)=Y_j\}}$$

subject to

$$\begin{aligned}
\sum_{j=1}^{n} \frac{I_{\{\phi(X_j)=g, Y=h\}}}{n} &\leq \alpha_{hg} && g,\ h \in G,\ g \neq h \\
\phi(X_j) &= \arg\max_{h \in \mathcal{G}} (f(h|X_j) - \sum_{i \neq h} f_i(X_j)\lambda_{ih}) && j = 1, \ldots, n \\
\lambda_{ih} &\geq 0 && i,\ h \in G,\ i \neq h
\end{aligned}$$

Consider the DAMIP without the misclassification constraints. Finding the optimal values for the λ_{ih} variables is equivalent to finding a best linear classifier given the input data. The input data are the estimates for the conditional group density functions $f_h(x)$ and the prior probabilities π_h. An upper bound on the shatter coefficient for this class of linear functions is $(2(n-1)^{G(G-1)} + 2)^{G(G-1)}$ where the $G(G-1)$ term derives from the number of λ_{ih} variables, which is the same as the dimension of the "input data" and the number of hyperplanes in the class of possible classifiers. Considering this bound on the shatter coefficient and the theorem of Vapnik and Chervonenkis concerning the convergence

of frequencies to their probabilities,

$$\begin{aligned} P\{\sup_{\phi\in\mathcal{C}} |B_n(\phi) - B(\phi)| > \epsilon\} &= P\{\sup_{\phi\in\mathcal{C}} B(\phi) - B(\phi_n) > \epsilon\} \\ &= P\{B(\phi^\dagger) - B(\phi_n) > \epsilon\} \\ &\leq 8(2(n-1)^{G(G-1)} + 2)^{G(G-1)} e^{-n\epsilon^2/8} \end{aligned}$$

which implies that the objective function of the DAMIP converges uniformly to $P\{\phi^\dagger(X) = Y\}$.

Now consider the misclassification constraints. For every n, the DAMIP requires that

$$M_n^{ab}(\phi) = \sum_{j=1}^{n} \frac{I_{\{\phi(X_j)=b, Y=a\}}}{n} \leq \alpha_{ab}$$

The left-hand side $M_n^{ab}(\phi)$ is expressed as another empirical risk functional associated with the DAMIP. The indicator functions are selected from the same set as for the objective function. Applying again the fundamental result from Vapnik and Chervonenkis [83], $M_n^{ab}(\phi)$ will converge uniformly to $P\{\phi(X) = b, Y = a\}$. Because each term in the sequence will have $M_n^{ab} \leq \alpha_{ab}$, the limit will also satisfy the constraint. This convergence occurs for all pairs of groups a and b. Therefore, any convergent sequence of solutions to the DAMIP will converge to a solution satisfying the misclassification constraints.

Suppose that for groups a and b, the Bayes optimal solution has $P\{\phi^*(X) = b, Y = a\} \geq \alpha_{ab}$. The Anderson optimal solution will have $P\{\phi^\dagger(X) = b, Y = a\} = \alpha_{ab}$. The misclassification rates of the DAMIP must converge to the misclassification limits because if they converged to a value strictly less than α_{ab}, then the optimality of an Anderson optimal solution would be contradicted. If, on the other hand, the Bayes optimal solution has $P\{\phi^*(X) = b, Y = a\} < \alpha_{ab}$, then the sequence of misclassification rates of solutions to the DAMIP will converge to some value between $P\{\phi^*(X) = b, Y = a\}$ and α_{ab}. The Bayes optimal solution is the best possible, so the sequence cannot converge to a value less than the Bayes optimal rate. Additionally, every solution to the DAMIP will have rates less than or equal to α_{ab}, so the sequence will converge to a solution that satisfies the misclassification constraints.

Thus, we have that the objective value of the DAMIP converges uniformly to $P\{\phi^\dagger(X) =$

$Y\}$ and any limit of a sequence of solutions for the DAMIP satisfies the misclassification constraints as the sample size increases. This convergence does not depend on the distribution of the data. The DAMIP is therefore strongly universally consistent. □

4.2 Stability of solutions to the DAMIP and stability of the corresponding classification rules

Classification methods can be described as a way of partitioning data or some transformation of the data. Using the DAMIP involves estimating prior probabilities and likelihood functions, which can be considered a nonlinear transformation of the raw data. The coefficients of the modified posterior probabilities are then derived from the solution of a mixed-integer program, which is a projection of the prior probabilities and likelihood function values onto a linear subspace. Given an observation x^{gj}, likelihood functions $f_h(x^{gj})$ for each $h \in \mathcal{G}$, and prior probability π_g, the modified posterior probability is

$$L_{hgj} = \pi_h f_h(x^{gj}) - \sum_{i \neq h} f_i(x^{gj}) \lambda_{ih}$$

An observation is allocated to a group if the corresponding modified posterior probability is nonnegative and is the largest modified posterior probability among the groups for that observation, or $x^{gj} \in h$ if

$$\begin{aligned} L_{agj} - L_{hgj} &\geq 0 \quad \forall\ h \in \mathcal{G} \\ L_{agj} &\geq 0 \end{aligned}$$

or, equivalently,

$$\begin{aligned} \sum_{i \neq a} f_i(x^{gj}) \lambda_{ia} - \sum_{k \neq h} f_k(x^{gj}) \lambda_{kh} + \pi_a f_a(x^{gj}) - \pi_h f_h(x^{gj}) &\geq 0 \quad \forall\ i \in \mathcal{G} \\ \sum_{i \neq a} f_a(x^{gj}) \lambda_{ia} &\geq 0 \end{aligned}$$

Note that prior to solving for the λ_{ih}'s, the only quantities that depend on the data are the estimates of the likelihood functions evaluated at the observations. If a set of λ_{ih}'s is given by a solution of the DAMIP and the estimates of prior probabilities are fixed, the equations above partition the observations in the space of their likelihood function values

$f_h(x^{gj})$. The partitions are defined by hyperplanes, as the equations are linear in the priors π_h and lambda's λ_{ih}.

In a sense, the DAMIP seeks to partition entities with linear hyperplanes in the space of their likelihood function values. A natural question is how well the hyperplanes separate the observations; in other words, is there the possibility of confusing the allocation of an observation between multiple groups? Can the regions between groups be adjusted? With respect to the DAMIP, the answers to these questions characterize the stability of solutions to the mixed-integer program and therefore the stability of the corresponding classification rules.

The formulation of the DAMIP (Section 3.1) as a mixed-integer program requires each observation to be allocated to exactly one group. Additionally, the constraints $y_{gj} \geq \epsilon u_{hgj}$ are added so that if an entity is allocated to a group (instead of the reserved judgment region), the modified posterior probability is at least ϵ. These new ϵ constraints, together with the original ϵ constraints, provide a natural stability to the classification rules. An observation x^{gj} is placed in group a if and only if

$$\begin{aligned} L_{agj} - L_{hgj} &\geq \epsilon \quad \forall\ h \in \mathcal{G} \\ L_{agj} &\geq \epsilon \end{aligned}$$

or,

$$\begin{aligned} \sum_{i \neq a} f_i(x^{gj})\lambda_{ia} - \sum_{k \neq h} f_k(x^{gj})\lambda_{kh} + \pi_a f_a(x^{gj}) - \pi_h f_h(x^{gj}) &\geq \epsilon \quad \forall\ i \in \mathcal{G} \\ \sum_{i \neq a} f_a(x^{gj})\lambda_{ia} &\geq \epsilon \end{aligned}$$

so that the relative difference in the modified posterior probability for h with the other modified posterior probabilities is at least ϵ. Therefore, there is a buffer between every pair of regions, the thickness of which is determined in part by the ϵ constant. Let β_{ab} be the vector coefficients of the hyperplane that partitions likelihood function values to separate groups a and b, and let $f(x^{gj}) = [f_1(x^{gj}), f_2(x^{gj}), \ldots, f_G(x^{gj})]$ be the vector of likelihood function values. The distance of observation x^{gj} to the ab hyperplane is

$$\frac{\beta_{ab}^T f(x^{gj})}{||\beta_{ab}||} \geq \frac{\epsilon}{||\beta_{ab}||}$$

Therefore, the smaller the coefficients of the hyperplane (the π_h's and λ_{ih}'s), the more stable the solution. The distance to the hyperplane depends partially on the value of ϵ, which can be changed for each entity-group combination. Restrictions can also be placed on the size of the λ_{ih}'s. One begins to think of the objective involved in support vector machines which has a term to bound the size of the coefficients of the linear hyperplane in a high dimensional space and another term to reduce training error. Adding a term such as $-||\beta||^2$ would convert the DAMIP into a quadratic mixed-integer program which is beyond the scope of this work. chapter 4

Chapter V

Computational Methods

The solution of the DAMIP is tested in a branch and cut framework. The ideas developed in the previous chapter are implemented with industry standard software. The conflict graph and conflict 3-hypergraph are used to derive cuts at branch and bound nodes. Tailored strategies for finding initial feasible solutions and setting values for M are employed. A heuristic for finding integer feasible solutions is added and a specialized branching scheme is implemented.

5.1 Formulation

The formulation used in testing solution strategies and classification accuracy of the DAMIP is the same as that given in Section 3.1 with the exception that the L_{hgj} variables are substituted out. The formulation is

$$\text{maximize} \sum_{g\in\mathcal{G}} \sum_{j\in\mathcal{N}_g} u_{ggj}$$

subject to

$$\begin{aligned}
y_{gj} - \pi_h f_h(x^{gj}) + \sum_{\substack{i\in\mathcal{G}\\ i\neq h}} f_i(x^{gj})\lambda_{ih} &\leq M(1-u_{hgj}) && g,\ h\in\mathcal{G},\ j\in\mathcal{N}_g\\
y_{gj} &\leq M(1-u_{0gj}) && g\in\mathcal{G},\ j\in\mathcal{N}_g\\
y_{gj} &\geq \epsilon u_{hgj} && g,\ h\in\mathcal{G},\ j\in\mathcal{N}_g\\
y_{gj} - \pi_h f_h(x^{gj}) + \sum_{\substack{i\in\mathcal{G}\\ i\neq h}} f_i(x^{gj})\lambda_{ih} &\geq \epsilon(1-u_{hgj}) && g,\ h\in\mathcal{G},\ j\in\mathcal{N}_g\\
\sum_{0,h\in\mathcal{G}} u_{hgj} &= 1 && g\in\mathcal{G},\ j\in\mathcal{N}_g\\
\sum_{j\in\mathcal{N}_g} u_{hgj} &\leq \lfloor \alpha_{hg} n_g \rfloor && h,\ g\in\mathcal{G},\ h\neq g
\end{aligned}$$

$$y_{gj}\geq 0,\ \lambda_{ih}\geq 0,\ u_{hgj}\in\{0,1\}$$

5.2 Finding an initial integer feasible solution

The difficulty in finding an initial integer feasible solution depends in large part on the misclassification limits. If the number of misclassified training entities is restricted to zero, then most of the integer variables are zero in every feasible solution. The DAMIP either correctly classifies an observation or places it in the reserved judgment region, and other integer variables for that observation can be set to zero before branch-and-bound begins. This case is intuitively easier to solve than other misclassification limits, and returns a solution that is integer feasible for any other misclassification limit desired. Also, if the data are completely separable, the solution to this problem quickly returns a solution in which every entity is correctly classified.

If the number of misclassified observations is unrestricted, the "Bayes rule" always provides an initial feasible solution. The Bayes rule allocates entities to groups for which the estimate for the posterior probability is largest. This rule corresponds to a solution of the DAMIP where $\lambda_{ih} = 0$ for all i and h.

For misclassification limits in between the two extremes, finding an initial integer feasible solution for the DAMIP may be difficult. One way to guarantee an initial integer feasible solution is to use the solution derived when misclassification limits are set at zero. For higher misclassification limits, for example $15 - 20\%$ and higher, this solution is unlikely to be a "good" integer feasible solution for the DAMIP. Note that a solution derived with misclassification rates of 5% is feasible for the problem with rates of 5% and higher. Therefore, slowly raising the misclassification limits and using the preceding solutions guarantees good initial integer feasible solutions. In the computational tests of Section 6.2.2, the problems are solved in order of increasing misclassification limits and optimal solutions are used as initial feasible solutions for subsequent problems.

5.3 Defining values for big M

The values for M in the formulation are derived using the upper bound results in Section 3.6, with a maximum value of 100. A different value for M is calculated for each integer variable. If the calculated upper bound for a particular M exceeds 100, then that M

is set to 100. According to the analysis in Section 3.6, this upper bound on M allows observations to be allocated to groups h (subject to misclassification constraints) for which $f_h(x^{gj}) > 1/(32 \cdot 3) = 1/96$ for a *3-group* problem and for which $f_h(x^{gj}) > 1/(32 \cdot 5) = 1/160$ for a *5-group* problem.

5.4 Generating and storing the conflict graph and conflict 3-hypergraph

As shown in Section 3.4, the solution of small linear programs can be used to find edges in the conflict graph and hypergraphs. Necessary conditions and sufficient conditions are given for edges to be present in the graph and hypergraphs. These concepts are implemented for computational testing to generate and store the conflict graph and conflict 3-hypergraph. The conflict graph and conflict 3-hypergraph are used to derive cutting planes during solution of the DAMIP.

5.4.1 Necessary and sufficient conditions for edges in the conflict graph

In Section 3.5.2, necessary and sufficient conditions are given for the existence of edges of the conflict graph of the form (u_{0j}, u_{gk}) or (u_{hj}, u_{gk}) in the absence of misclassification constraints. If misclassification constraints are present, the conditions are sufficient but not necessary.

An alternative to using the conditions for finding edges in the conflict graph is to determine if a corresponding linear program is infeasible. Using the conditions to determine the edges of the conflict graph is faster and more accurate than determining if the linear program is infeasible. As an illustration of the increased accuracy, consider the input data from the *va* data set

$$\begin{aligned}
f_1(x^{5j}) &= 0.806546994424451 \quad & f_1(x^{3k}) &= 0.017962285617577 \\
f_2(x^{5j}) &= 0.146253643655761 \quad & f_2(x^{3k}) &= 0.037324280602612 \\
f_3(x^{5j}) &= 0.349467241010385 \quad & f_3(x^{3k}) &= 0.115602826096817 \\
f_4(x^{5j}) &= 0.010550943880602 \quad & f_4(x^{3k}) &= 0.293548055935253 \\
f_5(x^{5j}) &= 0.001701693938146 \quad & f_5(x^{3k}) &= 0.535562551747739
\end{aligned}$$

and the associated linear program

$$\min 0 \sum_{i,h} \lambda_{ih}$$

subject to

$$\begin{aligned}
0.806546994424451\lambda_{13} + 0.146253643655761\lambda_{23} & \\
+0.010550943880602\lambda_{43} + 0.001701693938146\lambda_{53} & \\
-0.806546994424451\lambda_{15} - 0.146253643655761\lambda_{25} & \\
-0.0349467241010385\lambda_{35} - 0.010550943880602\lambda_{45} &\geq 0.00686886004637505 \\
0.806546994424451\lambda_{15} + 0.146253643655761\lambda_{25} & \\
+0.0349467241010385\lambda_{35} + 0.010550943880602\lambda_{45} &\leq 0.00007961141231096 \\
-0.017962285617577\lambda_{13} - 0.037324280602612\lambda_{23} & \\
-0.293548055935253\lambda_{43} - 0.535562551747739\lambda_{53} & \\
+0.017962285617577\lambda_{15} + 0.0373242806026121\lambda_{25} & \\
+0.115602826096817\lambda_{35} + 0.293548055935253\lambda_{45} &\geq 0.00207020074087797 \\
0.0179622856175779\lambda_{13} + 0.0373242806026121\lambda_{23} & \\
+0.293548055935253\lambda_{43} + 0.535562551747739\lambda_{53} &\leq 0.0229853572356244 \\
\lambda_{ih} &\geq 0
\end{aligned}$$

Using the default tolerances, CPLEX determines that this linear program is feasible, though the conditions for infeasibility are satisfied. If the feasibility tolerance setting in CPLEX *CPX_PARAM_EPRHS* [45] is decreased to 1×10^{-8}, CPLEX determines that the linear program is infeasible. Therefore, the conditions for infeasibility are used to derive the conflict graph rather than solve the series of linear programs.

5.4.2 Finding edges of the conflict 3-hypergraph

Necessary and sufficient conditions are not known for the existence of edges in the conflict 3-hypergraph. The known necessary conditions for infeasibility of associated linear programs given in Section 3.5.1 can be used to reduce the number of linear programs solved. As with the conditions for conflict graph edges, the evaluation of the necessary conditions is much faster than determining the feasibility of the linear program.

As noted in Section 3.5, any edge of the conflict graph can be extended to an edge of

the conflict 3-hypergraph. The edges of the conflict 3-hypergraph that are implied by the conflict graph are not checked twice. This practice further reduces the computation needed to check for edges of the conflict 3-hypergraph.

5.4.3 Storing the conflict graph and conflict 3-hypergraph

Decisions about the method for generating and storing the conflict graph and conflict 3-hypergraph were made based on the size of the example problems. Table 1 contains information about the size of the conflict graphs for various test problems with a misclassification limit 5% for all pairs of groups. The data that were used in generating the test problems are described in Section 6.1. The relatively small number of nodes allowed for storage of the conflict graph in both an adjacency matrix and in adjacency lists. The different representations are used in different graph-searching algorithms as described in Section 5.6.

Table 1: The size of the conflict graph and the number of variables(nodes) fixed to zero for various data sets when the misclassification limit is 5% for all pairs of groups. The size of the conflict graph is given before and after the fixed variables are set to zero. Conflict graph density is calculated as $100 \times \frac{(Edges\ after)}{\binom{Node\ after}{2}}$. The data sets are described in Section 6.1.

Problem	Groups	Entities	Nodes	Edges	Fixed Variables	Nodes after	Edges after	Density (%)
iris	3	135	540	47280	85	455	28628	27.7
wine	3	153	612	59853	174	438	19558	20.4
new-thyroid	3	189	756	67522	49	707	51120	20.5
sepal	3	135	540	38458	74	466	26798	24.7
FNlnVN.alltree1	3	63	252	8283	116	136	963	10.5
va	5	76	456	15924	237	219	2507	10.5
switzerland	5	90	540	23168	186	354	6190	9.9
hungarian	5	233	1398	178558	254	1144	82427	12.6
cleveland	5	264	1584	227953	309	1275	101443	12.5

The conflict 3-hypergraph is either generated before optimization and stored or edges are checked as needed inside the structure-finding algorithms. When the conflict hypergraph is generated, it is stored using an extension of the data structure introduced by Atamtürk et. al [5] for conflict graphs. The structure for conflict graphs employs an array *last* of length equal to the number of nodes and arrays *adj* and *next* of length equal to twice the number of edges in the graph. For node v, the entry $adj[last[v]]$ contains the last node added to *adj* to which v is adjacent. The entry $adj[next[last[v]]]$ contains the index of the second-to-last node added to *adj* to which v is adjacent. If no other edges are adjacent to v, then $next[last[v]] = 0$.

The conflict 3-hypergraph is stored using an array *last* of length equal to the number of nodes and arrays $adj1$, $adj2$, and *next* of length equal to 3 times the number of edges in the hypergraph. For node v, the last edge added to the data structure containing v is $(v, adj1[last[v]], adj2[last[v]])$. The second-to-last edge added to the data structure containing v is $(v, adj1[next[last[v]]], adj2[next[last[v]]])$ and so on. When an edge is added to the data structure, 3 entries are added to $adj1$, $adj2$, and *next* for each node in the edge. As an example, suppose that nodes $\{1, 2, 3, 4, 5, 6\}$ and edges $\{(1, 2, 3), (1, 4, 5), (1, 3, 6), (2, 3, 4)\}$ are contained in the hypergraph. Then the array entries are

$$\begin{aligned} last &= [7, 10, 11, 12, 6, 9] \\ adj1 &= [2, 1, 1, 4, 1, 1, 3, 1, 1, 3, 2, 2] \\ adj2 &= [3, 3, 2, 5, 5, 4, 6, 6, 3, 4, 4, 3] \\ next &= [0, 0, 0, 1, 0, 0, 4, 3, 0, 2, 8, 5] \end{aligned}$$

Note that this structure can easily be extended further to conflict n-hypergraphs using an array *last* of length equal to the number of nodes, $n - 1$ arrays adj_i of length n times the number of edges in the hypergraph, and an array *next* of length n times the number of edges in the hypergraph.

5.4.4 Floating point accuracy and validity of conflict graph and hypergraph edges

The integer programming formulation of the DAMIP is intrinsically stable due to the presence of the ϵ constraints (see Chapter 3). The ϵ constants ensure a buffer between the regions corresponding to the different groups. The linear programs considered in generating edges of the conflict graph and conflict 3-hypergraph do not take into account the ϵ constants in the sense that they are formulated with the assumption that there is no buffer between the group regions. Therefore, if an edge is present in the conflict graph (or hypergraph), then the associated integer variables cannot both have value 1 for any value of ϵ, including in the limit as ϵ approaches 0. In other words, the edges of the conflict graph and conflict hypergraph are generated in a conservative manner with respect to their validity for the integer program.

5.5 Fixing variables

As described in Section 3.4.3, the conflict graph and misclassification constraints can immediately imply that certain integer variables will be 0 for any feasible solution. In the computational tests, these variables are fixed at 0 before optimization begins. Table 1 contains information about how many variables are fixed for each data set when the misclassification limit is set to 5%.

5.6 Cutting planes

The conflict graph is generated and stored before optimization for the rapid generation of cutting planes at nodes in the branch and bound tree. Structures such as maximal cliques, odd holes, and maximal hypercliques correspond to valid inequalities for the integer programming formulation of the DAMIP.

5.6.1 Maximal clique constraints from the conflict graph

At nodes in the branch and bound tree, violated maximal clique inequalities are found in the subgraph induced by nodes corresponding to fractional-valued integer variables and integer variables with value 1. The conflict graph is searched using an implementation of

the enumeration algorithm proposed by Bron and Kerbosch [21]. The algorithm maintains sets of nodes *compsub*, *candidates*, and *not* in its search. The set *compsub* is the current clique that is to be extended to a larger clique, or reduced to build another maximal clique. The set *candidates* is the set of nodes not in *compsub* that are adjacent to all nodes in *compsub*; these nodes are eligible to extend *candidates*. The set *not* is the set of nodes that have already served as an extension of the current configuration of *compsub* and are now excluded. The algorithm includes an extension operator on the set *compsub* in which a member of *candidates* is added.

The basic algorithm is as follows

1. Select a candidate from *candidates* and add it to *compsub*.
2. Create new sets *candidates* and *not* from the old sets by removing nodes not connected to the selected candidate.
3. Extend *compsub* based on the sets just formed.
4. Remove the selected candidate from *compsub*, and place it in *not*.

A maximal clique is contained in *compsub* when the sets *candidates* and *not* are empty. If at some stage *not* contains a point connected to all points in *candidates*, then that point will never be removed from *not* and a maximal clique will not be found. Extensions on the current *compsub* need not be explored, and those configurations of *compsub* have been fathomed in a branch-and-bound sense. Bron and Kerbosch [21] further enhance the algorithm by intelligently selecting a candidate from *candidates* so that the fathoming condition is met as rapidly as possible.

The maximal clique algorithm implemented for the DAMIP uses the adjacency matrix representation of the conflict graph. As each maximal clique is generated, the corresponding maximal clique inequality is checked to determine if the current solution violates it. If so, then integer variables with value 0 at the current branch-and-bound node are lifted in by checking for variables whose nodes in the conflict graph are adjacent to nodes in the maximal

clique and nodes that have already been lifted in. After a maximal set of integer variables is lifted, then the cut is added, either locally or globally.

5.6.2 Odd hole constraints from the conflict graph

Odd holes are searched for in the entire conflict graph at a node in the branch-and-bound tree, as opposed to the subgraph induced by fractional variables. A breadth-first search heuristic, described by Bixby and Lee [12], is used to find odd holes. An odd cycle is found when two nodes are on the same level of the breadth-first search tree. The algorithm then backtracks to determine if the odd cycle contains any chords. If the odd cycle is an odd hole and the corresponding inequality is violated by the current solution, then the odd hole inequality is lifted and added.

The breadth-first search is repeated ten times at each node of the branch and bound tree, each time using a randomly selected node in the conflict graph as the root. When a violated odd hole inequality is found, the search is terminated. If the conflict graph is not connected, every tree in the forest is searched.

A violated odd hole inequality is lifted in stages.

1. Variables with nonzero values that are adjacent to every node in the inequality are lifted with a coefficient equal to the size of the odd hole.

2. Remaining variables with nonzero values adjacent to all but one node in the inequality are lifted with a coefficient equal to one less than the size of the odd hole.

3. Remaining variables with nonzero values and nonzero lifting coefficients are lifted.

4. Remaining variables with nonzero lifting coefficients are lifted.

The cut is added after lifting, either locally or globally. The breadth-first search algorithm for the DAMIP uses an adjacency list representation of the conflict graph which is better suited for building the queue than the adjacency matrix representation. The lifting procedure employs the adjacency matrix representation.

5.6.3 Maximal hyperclique constraints from the conflict 3-hypergraph

Maximal hypercliques, as defined by Easton et. al [28], in an n-hypergraph are structures containing all $\binom{m}{n}$ edges for some subset of m nodes.

Violated maximal hyperclique inequalities in the conflict 3-hypergraph are found, lifted, and added at nodes in the branch-and-bound tree. The method for finding maximal hypercliques in the conflict 3-hypergraph is an extension of the algorithm by Bron and Kerbosch [21] for maximal cliques. The new algorithm enumerates all maximal hypercliques and takes advantage of the fathoming concepts in the original algorithm.

In the maximal n-hyperclique algorithm, the sets *compsub*, *candidates*, and *not* are maintained as before. The definitions are slightly altered:

- *compsub* contains a hyperclique that may or may not be extended. Every subset of n nodes in *compsub* is an edge in the conflict n-hypergraph, or the nodes in *compsub* are contained in an edge.

- *candidates* contains nodes v such that every subset of n nodes containing v and nodes from *compsub* is an edge in the conflict n-hypergraph, or v and the nodes from *compsub* are contained in an edge. In other words, v is eligible to extend *compsub*.

- *not* contains nodes v that were contained in a previous extension of the current *compsub*. Therefore, every subset of n nodes containing v and nodes from *compsub* is an edge in the conflict n-hypergraph, or sets of nodes containing v and the nodes from *compsub* is contained in an edge.

In short, the notion of adjacency of a node to a set of nodes is naturally extended to imply that every subset of n nodes is an edge, or the node and set of nodes is contained in an edge.

Due to memory and time limitations, only the conflict 3-hypergraph is used for finding maximal hypercliques. Generating and storing the hypergraph before optimization proved to be exceedingly time- and memory-intensive for large problems, so generating the hypergraph on-the-fly is also tested.

The adjacency matrix representation of the conflict graph is used to check edges of the conflict graph, which can be extended to edges in the conflict 3-hypergraph (see Section 3.5). The conflict 3-hypergraph is stored as described in Section 5.4, and adjacency to a node v is determined using an algorithm which is an extension of that suggested by Atamtürk et al. [5]. The following algorithm finds all edges containing node v

Algorithm 5.6.1. *1.* $k = last[v]$

2. while $k \neq 0$ *do*

3. (a) print "edge $(v, adj1[k], adj2[k])$ *exists in the conflict 3-hypergraph"*

(b) $k = next[k]$

This algorithm can easily be extended for the conflict n-hypergraph.

When a violated maximal hyperclique inequality is found, the inequality is lifted and then added, either locally or globally. Variables are lifted in a manner similar to the routine for odd hole inequalities by using information from the conflict graph:

1. Variables with nonzero values that are adjacent to every node in the inequality are lifted with a coefficient equal to 2.

2. Remaining variables with nonzero values adjacent to all but one node in the inequality are lifted with a coefficient equal to 1.

3. Remaining variables with nonzero lifting coefficients are lifted.

5.6.4 Implications of other inequalities

The inequalities of the form $u_{hgj} \leq u_{hgk}$ (see Section 3.4.2) that are implied by the conflict graph and misclassification constraints are stored before optimization and then added as they are violated. The lifting coefficient of any other integer variable is zero, so these constraints are not lifted.

5.7 Heuristic

As noted in chapter 3, the values of the λ_{ih}s determine the values the rest of the variables in integer programming formulation of the DAMIP. At each node in the branch and bound tree, there exists a integer solution associated with the current values for the λ_{ih}'s. This solution may or may not satisfy the misclassification constraints and may or may not reflect the robustness provided by the ϵ constraints in the DAMIP formulation.

A heuristic is implemented to check the integer solution derived from the λ_{ih} values at every node in the branch and bound tree. The solution is preserved as the incumbent solution if the new objective is better than the current incumbent and the solution satisfies the misclassification constraints and preserved the robustness enforced by the ϵ constraints.

5.8 Branching strategies

The DAMIP is tested with a branching strategy that attempts further exploit information in the conflict graph. The branching strategy involves branching on certain hyperplanes, branching on the most fractional correct classification variable, and branching on the variable that CPLEX provides based on strong branching information. The branching scheme creates 2 branches at each node, trying each of the following components in order.

1. **The "correct classification hyperplane".** Let F_c be the set of correct classification variables with fractional values. If $\sum_{(g,j):u_{ggj}\in F_c} u_{ggj} > |F_c| - 1$, then one branch is created with the constraint $\sum_{(g,j):u_{ggj}\in F_c} u_{ggj} \leq |F_c| - 1$, and another branch is created with all variables in F_c set to value 1. In the second branch, variables adjacent to members of F_c in the conflict graph are set to value 0.

2. **The "misclassification hyperplane".** Let F_{hg} be the set of variables with fractional values that correspond to allocating entities from group g to group h. If $\sum_{j:u_{hgj}\in F_{hg}} u_{hgj} > |F_{hg}| - 1$, then one branch is created with the constraint $\sum_{j:u_{hgj}\in F_{hg}} u_{hgj} \leq |F_{hg}| - 1$, and another branch is created with all variables in F_{hg} set to value 1. In the second branch, variables adjacent to members of F_{hg} in the conflict graph are set to value 0.

3. **Most fractional correct classification variable.** Branches are created for the most fractional correct classification variable. For the branch with the variable set to value 1, all adjacent variables in the conflict graph are set to value 0.

4. **CPLEX-provided variable.** Branches are created for the CPLEX-provided variable based on strong branching information. For the branch with the variable set to value 1, all adjacent variables in the conflict graph are set to value 0.

Note that when a single variable is selected for branching, in effect hyperplanes with nonzero coefficients on the λ_{ih} variables are added to the formulation. If u_{hgj} is selected for branching, the branch for which $u_{hgj} = 1$ requires that $L_{hgj} - L_{agj} \geq \epsilon$ for all $a \neq h$. The branch for which $u_{hgj} = 0$ requires that $y_{gj} - L_{hgj} \geq \epsilon$.

5.9 Preparation of data

For a set of training observations, estimates of the prior probabilities and estimates for the conditional group density functions for each observation are generated. The prior probability for a group is estimated as the proportion of training entities from that group. The conditional group density functions are generated under the assumption that the data is multivariate normal with equal covariance matrices between the groups. This method treats discrete-valued attributes as continuous attributes.

Any method for estimating prior probabilities likelihood functions can be used as input to the DAMIP. For all of the tests that follow, the conditional group density functions are approximated using the values returned by linear discriminant functions where the attributes are assumed to be multivariate normal with a common covariance matrix. This method for generating the conditional group density function values can lead one to view the DAMIP as the second in a two-step procedure, or a modification of the rules derived by another method.

Splus 6.0 for Unix/Linux is used to generate the likelihood function estimates using the *discrim()* function.

Missing values are treated by removing attributes and observations so that no missing values remained. The treatment of missing values is further discussed in the next chapter.

After the estimates of the likelihood functions are generated, the conditional group densities evaluated at some of the points can be extremely small. Some of the conditional group density values are on the order of 1×10^{-20} and smaller. Such small values as 1×10^{-20} can result in floating point inaccuracies in CPLEX, as the default integrality and feasibility tolerances are set to 1×10^{-5} and 1×10^{-6}, respectively. In order to avoid such difficulties, conditional probability densities are perturbed. If a conditional probability density evaluates to less than 1×10^{-6}, the value is set to 1×10^{-6} for the remainder of the calculations.

chapter 5

Chapter VI

Computational Tests

The computational methods for solving the DAMIP are implemented and tested for improvements in computational efficiency. Additionally, the properties of the classification rules returned by the DAMIP are compared to rules that are derived using standard methods.

The tests for computational efficiency are performed using real-world data sets. The tests for classification accuracy are performed using the same data, along with simulated data that are generated from specified distributions.

Descriptions of the real-world data sets that are used for computational tests are in the next section, followed by performance comparisons of industry standard software to code enhanced with the techniques of Chapter 5. Finally, comparisons of the classification accuracy of the DAMIP with standard methods under various conditions are included.

6.1 Real-world data sets

Let an *n-group* problem be a problem where discrimination between n groups is of interest. Five of the data sets are from 3-group problems, and the remaining four data sets are from 5-group problems. The data sets *wine*, *new-thyroid*, *iris*, and *sepal* are 3-group problems from the UCI machine learning database repository. The data set *sepal* is derived from *iris* as described in Gallagher et al. [35] by considering a subset of attributes. The data sets *va*, *switzerland*, *hungarian*, and *cleveland* are 5-groups problems from the UCI machine learning database repository. The remaining data set, *FNlnVN*, is a 3-group problem generated from data as described in [88] from cell motility data gathered at Georgia Tech by Adele Wright.

6.1.1 3-group data sets

wine The *wine* recognition data is concerned with discriminating between wines produced in the same region of Italy, but from three different cultivars (plants). The 13 continuous

attributes measure the amount of various substances found in the wines. There are 59 instances of class 1 wine, 71 instances of class 2 wine, and 48 instances of class 3 wine.

new-thyroid The thyroid gland data is used to discriminate between euthyroidism (normal function), hypothyroidism, and hyperthyroidism based on the results of five lab tests. The results of the lab tests produce continuous measurable attributes. There are 150 normal cases (euthyroidism), 35 hyper, and 30 hypo cases in the original data set.

iris The Iris Plants Database was created by Fisher and is concerned with discriminating between Iris Setosa, Iris Versicolour, and Iris Virginica based on sepal length, sepal width, petal length, and petal width. There are 50 entities in each group, for a total of 150 entities.

sepal The *sepal* data set is created from the *iris* data set as described in [35]. The data set is equivalent to the *iris* data set, except that only sepal length and sepal width are included as attributes.

FNlnVN The *FNlnVN* data set is concerned with discriminating between the behavior of cancerous cells placed in culture with extracellular matrix (ECM) proteins. The groups are determined by the proteins on which cells are placed. For this data set, cells are placed on fibronectin (Fn), laminin (Ln), and vitronectin (Vn). Continuous measurable attributes are gathered for each of the cells. Eight of the attributes are used in this data set. The objective of the predictive model is to determine how well the selected attributes discriminate between the effects of the various proteins on cell behavior. The data consists of 30 cells placed on fibronectin, 30 cells placed on laminin, and 25 cells placed on vitronectin.

6.1.2 5-group data sets

All four of the 5-group problems are from databases that were established in the interest of improving heart disease diagnosis. The five groups are defined by four levels of heart disease and a group for which heart disease is absent. The entities are the patients for

which a variety of measurable attributes have been recorded. The attributes include age, sex, resting blood pressure, cholesterol, and resting electrocardiographic results.

The distribution of entities among the classes of heart disease for computational testing in the next two sections are given in Table 2.

Table 2: The number of entities belonging to each class for the heart disease diagnosis data sets. The quantities represent 1 of 10 folds of training data generated, or a sample of 90% of the data.

	1	2	3	4	5
va	14	25	18	15	4
switzerland	7	35	24	20	4
hungarian	146	32	20	23	12
cleveland	141	50	31	32	10

va The data set *va* was collected at the V.A. Medical Center by Dr. Robert Detrano. For computational purposes, attributes 12 and 13 are removed from consideration because most of the values are missing. Also, 94 entities do not contain sufficient information for assigning group density values and are removed.

switzerland The *switzerland* data set was collected at University Hospital by Dr. William Steinbrunn and Dr. Matthias Pfisterer. Attributes 7, 12, and 13 are removed because of missing values. There are 18 entities in the original data set removed due to missing values.

hungarian The *hungarian* data set was collected by Dr. Andras Janosi at the Hungarian Institute of Cardiology in Budapest. Attributes 11 through 13 are removed from consideration due to missing values. There are 27 entities removed due to missing values for other attributes.

cleveland The *cleveland* data set was collected by Dr. Robert Detrano at the V.A. Medical Center Cleveland Clinic Foundation. None of the attributes or entities are removed from the original data set.

6.2 Comparison of performance: enhanced code vs. CPLEX

Instances of the DAMIP are solved using CPLEX Callable Library V8.1 (CPLEX). The code is also enhanced with the cuts, branching strategy, heuristic described in Chapter 5 (enhanced code) and then used to solve the same instances to determine the improvement in solution times due to the new methods.

6.2.1 Methods and data

The two codes are tested using real-world data from the 9 data sets described in the previous section. For computational testing, a single fold of data is selected out of a set of ten folds. In other words, data from approximately 90% of the observations is selected at random from each data set. For a particular data set, the same fold of data is used for all tests of the speed and efficiency of the codes.

Preliminary results showed that the 3-group instances are solved in less than 5 seconds for both codes, so the tests for computational speed are focused on 5-group problems with varying misclassification limits imposed.

Computational tests are performed on a fold from each of the 5-group data sets using four sets of misclassification limits. Specifically, tests are run with $\alpha = \alpha_{hg} = 0.00$ for all h and g, $\alpha = 0.05$, $\alpha = 0.15$, and $\alpha = 1.00$. These runs correspond to misclassification limits of 0, 5, 15, and 100%, respectively, such that for a given h and g, at most $\lfloor \alpha \mathcal{N}_h \rfloor$ entities from group h are misclassified as belonging to group g. Problems will be referenced by the abbreviation and α values; i.e., *swi-0.05*, *cle-1.00*.

For all tests, the ϵ in the formulation of the DAMIP is set to 0.0001.

The 12 5-group problems are tested on machines with 2x2.4GHz Intel Xeon processors and 2GB RAM. CPLEX is set to terminate if either branch and bound tree memory reached 1.9 gigabytes or 200,000 CPU seconds passed.

The reader is referred to the ILOG CPLEX 8.1 Reference Manual [45] for information regarding parameters controlling cut generation, presolve, probing, optimization strategy, branching procedures, and node selection procedures. The user-defined cuts, branching schemes, and heuristics of the enhanced code are implemented using callback functions which are described in the ILOG CPLEX 8.1 Advanced Reference Manual [46].

Preliminary tests demonstrated that issues concerning floating-point accuracy cause CPLEX to consider infeasible solutions feasible, and suboptimal problems optimal. To facilitate proper comparisons and to ensure accurate calculation of the integrality gap remaining, the tolerance levels are set to their strictest settings in CPLEX and the enhanced code for all tests [45].

6.2.2 Results

Various settings for CPLEX are tested including the generation of cuts, branching strategies, solution strategies, and node selection strategies. Several settings are used in every test (with the exception of the test of the default settings for CPLEX). Strong branching is used as the branching strategy. The only CPLEX-generated cuts used are cliques and generalized

upper bound cuts (GUBs). The alternative best estimate node selection procedure is used.

Preliminary tests showed that generating and storing the conflict 3-hypergraph is prohibitive due to time and memory limitations for the larger instances of the DAMIP (data not shown). In every test reported, the conflict 3-hypergraph is generated and searched on-the-fly.

6.2.2.1 Enhanced code versus CPLEX

The best-performing configurations for CPLEX and the enhanced code are compared in Table 6.2.2.1. The best-performing settings that are observed for CPLEX include using strong branching, generating cliques and GUBs aggressively, probing at the most aggressive level, not using presolve, and using a strategy that emphasizes optimality over feasibility. The enhanced code compared in Table 6.2.2.1 employs these same settings with the exception that the strategy of moving the best bound is used. In separate tests, this strategy proves detrimental to the performance of CPLEX (data not shown). The enhanced code additionally employs maximal clique inequalities and odd hole inequalities derived from the conflict graph, variable fixing, a user-defined heuristic, a user-defined branching scheme, and upper bounds for parameters. These enhancements are described in Chapter 5. The conflict hypergraph is not used in the enhanced code because the time spent generating and finding violated cuts is unmerited (see Section 7).

The first two columns in Table 6.2.2.1 define the data set and misclassification limit. The misclassification limits is the maximum percentage of observations from each group that can be misclassified as belonging to another group. The limit is applied to each pair of groups.

The third through sixth columns summarize the performance of CPLEX on the test problems. Column three contains the highest objective value associated with an integer feasible solution found by CPLEX, column four contains the percentage of integrality gap remaining, column five contains the processor time for CPLEX to find a feasible solution with the objective value of column three, and column six contains the nodes solved by CPLEX before finding a feasible solution with the objective value of column three. By

definition, the percentage of integrality gap remaining is given by $(z_{UB}^{LP} - z^{IP})/(z_{root}^{LP} - z^{IP}) \times 100$, where the value z^{IP} is the optimal objective value of the mixed-integer program. If z^{IP} is unknown, the objective value associated with the best known integer feasible solution is used. The value z_{UB}^{LP} is the best objective value among active LP subproblems at termination. The value z_{root}^{LP} is the optimal objective value of the linear program relaxation at the root of the branch and bound tree.

Columns seven through ten summarize the performance of the enhanced code. Columns seven and eight are defined as columns three and four applied to solutions obtained by the enhanced code. Columns nine and ten contain the processor time and nodes, respectively, needed for the enhanced code to find an integer feasible solution to the test problem with an objective value at least as good as the best solution found by CPLEX alone.

Columns eleven and twelve summarize the improvement in time and nodes, respectively, that the enhanced code provides. Column eleven is calculated by dividing the quantity in column five by the quantity in column nine. Column twelve is calculated by dividing the quantity in column six by the quantity in column ten.

The enhanced code solves 8 of the 12 test problems to optimality (Table 6.2.2.1). CPLEX solves only 4 problems to optimality and obtains a solution with optimal objective in 7 of the 12 problems. The enhanced code finds solutions with objective values at least as good as those found by CPLEX in less time and with less nodes explored for all problems with the exception of *va-0.15*, *hun-0.15*, and *cle-1.00*. CPLEX did not find a feasible solution to *cle-0.15*. The percentage of integrality gap remaining for the enhanced code is smaller than that of CPLEX for the 4 problems that are not solved to optimality. The percentage of integrality gap remaining evaluates the quality of the lowest upper bound achieved, and does not consider the quality of the integer feasible solutions that are found. Thus, the enhanced code outperforms CPLEX in terms of finding upper and lower bounds on the optimal objective value for ever test problem except *cle-1.00*.

Table 3: CPLEX with cliques, GUBs, and strong branching vs. Enhanced code. CPLEX with cliques, GUBs, strong branching, compared to enhanced code including maximal clique cuts, odd hole cuts, variable fixing, user-defined branching scheme, and heuristic. The optimal objective values for problems *cle-0.05* and *cle-0.15* have not been verified, so the best-known objective values are used.

Problem		CPLEX				Enhanced code				Improvement ($\frac{CPLEX}{Enhanced\ code}$)	
Data	α_{ij}	Best obj. value	Gap rem. (%)	Time to best (s)	Nodes to best	Best obj. value	Gap rem. (%)	Time to CPLEX best (s)	Nodes to CPLEX best	Time to CPLEX best	Nodes to CPLEX best
va	0.05	27	0.0	5	110	27	0.0	1	4	5.0	27.5
va	0.15	50	0.0	388	4262	50	0.0	1329	10528	0.29	0.4
va	1.00	52	0.0	139	1490	52	0.0	22	133	6.3	11.2
swi	0.05	30	0.0	2363	33790	30	0.0	789	5036	3.0	6.7
swi	0.15	47	22.0	63013	163655	48	0.0	33783	64414	1.9	2.5
swi	1.00	55	17.1	5773	34538	55	0.0	2585	20968	2.2	1.6
hun	0.05	108	80.4	24896	73340	108	0.0	33375	21763	0.75	3.4
hun	0.15	143	61.2	87893	80731	155	25.1	2912	3048	30.2	26.5
hun	1.00	178	32.8	41850	42579	178	0.0	4276	12003	9.8	3.5
cle	0.05	85	54.2	177940	272243	90	37.2	45426	23582	3.9	11.5
cle	0.15	N/A	79.4	N/A	N/A	122	51.5	N/A	N/A	N/A	N/A
cle	1.00	187	48.2	39910	31003	187	25.0	104944	224682	0.4	0.1

6.2.2.2 Benefits of CPLEX-generated cliques and GUBs

Tables 4 through 6 summarize the performance of the enhanced code with various settings for CPLEX. The first column indicates the setting that is tested. The second and third columns identify the problem and the misclassification limits enforced. Column four contains the optimal objective value for the test problem. For *cle-0.15*, the optimal objective value is not known, so the best-known objective value is used. The fifth column contains the best objective value associated with an integer feasible solution found. Column six contains the lowest upper bound achieved by the enhanced code, and column seven contains the percentage of integrality gap remaining. The percentage of integrality gap remaining is defined in Section 6.2.2.1. The eighth column contains the number of nodes solved. Columns nine and ten contain the nodes solved and time, respectively, before the enhanced code finds a solution with the objective value of column five. Column eleven contains the total processor time used in solving the problem.

Extensive testing was performed to determine which classes of cuts available in CPLEX are most beneficial to solving DAMIP instances. In general, all of the cuts are detrimental to solution times with the exception of clique cuts and GUBs (data not shown). The contribution of cliques and GUBs to improved performance is tested by comparing generating these cuts at their most aggressive levels to not generating the cuts at all in Table 4. The two settings are compared for the problem instances with the misclassification limits at 15%. The enhanced code with the cuts aggressively generated outperforms the same without cuts in terms of time to best solution, percentage gap remaining, and nodes solved before the best solution on every test instance except *cle-0.15* (Table 4) and *cle-0.05* (data not shown).

Table 4: Benefit of CPLEX-generated cliques and GUBs for enhanced code. The performance of the enhanced code with and without cliques and GUBs on 5-*group* problems with misclassification limit 15%. The probing level is set to 3 (most aggressive) and the optimization strategy is set to 2 (emphasize optimality over feasibility). The optimal objective value for problem *cle-0.15* has not been verified, so the best known objective value is used.

Cliques/GUBs (Y/N)	Problem Data	α_{ij}	Solutions Opt Obj	Best Int Obj	Best UB (z_{UB}^{LP})	Gap Rem.(%)	Branch and Cut Nodes	Nodes to Best	Time to Best (s)	Cli	GUB	Total Time (s)
Y	*va*	0.15	50	50	50	0.0	8454	5823	663	9	0	908
N/A	*va*	0.15	50	50	50	0.0	12461	11374	2986	N/A	N/A	3145
Y	*swi*	0.15	48	48	48	0.0	107022	98994	58614	798	0	62756
N/A	*swi*	0.15	48	48	48	0.0	107089	102706	66263	N/A	N/A	68958
Y	*hun*	0.15	159	156	177.6656	34.0	132092	56043	78540	913	0	204071
N/A	*hun*	0.15	159	155	180.9966	59.0	111563	96083	176369	N/A	N/A	201639
Y	*cle*	0.15	(133)	87	191.4371	42.9	46651	0	0	552	0	205457
N/A	*cle*	0.15	(133)	132	190.9919	51.8	191887	70003	190361	N/A	N/A	206267

6.2.2.3 Benefits of aggressive probing

CPLEX allows various settings for probing, the process of determining logical implications of constraints before beginning the branch-and-bound algorithm. The settings range from no probing at all to very aggressive probing. There is little difference in performance between turning probing off compared to allowing CPLEX to automatically determine the level of probing (Table ref{probe). This behavior is an indication that by default, CPLEX does not incorporate probing when solving instances of the DAMIP. With probing set to the most aggressive level, the enhanced code is able to solve one more problem to optimality than with probing turned off. Additionally, the enhanced code with probing solves problems to optimality faster and found better optimal solutions faster than the enhanced code without probing or the default level of probing.

Table 5: Performance of enhanced code with various settings for probing. The performance of the enhanced code under various settings for *CPX_PARAM_PROBE*, the CPLEX parameter controlling probing [45]. The levels of probing tested are -1 (no probing), 0 (automatically determined by CPLEX), and 3 (the most probing). The computational data are shown for 5-*group* problems with misclassification limits set at 15%. CPLEX-generated cliques and GUBs are generated aggressively and the optimization strategy is set to 2 (emphasize optimality over feasibility). The optimal objective value for problem *cle-0.15* has not been verified, so the best known objective value is used.

Probing	Problem		Solutions				Branch and Cut			Total
Level	Data	α_{ij}	Opt Obj	Best Int Obj	Best UB (z_{UB}^{LP})	Gap Rem.(%)	Nodes	Nodes to Best	Time to Best (s)	Time (s)
-1	*va*	0.15	50	50	50	0.0	10385	8622	1174	1307
0	*va*	0.15	50	50	50	0.0	10385	8622	1150	1278
3	*va*	0.15	50	50	50	0.0	8454	5823	663	908
-1	*swi*	0.15	48	48	48	0.0	218949	183018	61528	69956
0	*swi*	0.15	48	48	48	0.0	218949	183018	62331	71834
3	*swi*	0.15	48	48	48	0.0	107022	98994	58614	62756
-1	*hun*	0.15	159	143	176.6658	23.9	129927	37246	65309	203885
0	*hun*	0.15	159	143	176.6658	23.9	104611	37246	66932	203180
3	*hun*	0.15	159	156	177.6656	34.0	132092	56043	78540	204071
-1	*cle*	0.15	(133)	83	190.3258	43.8	25615	0	0	202966
0	*cle*	0.15	(133)	83	190.3258	43.8	26015	0	0	203054
3	*cle*	0.15	(133)	87	191.4371	52.2	46651	0	0	205457

6.2.2.4 Optimization strategy

CPLEX allows for the specification of a solution strategy ranging from an emphasis on verifying optimality to an emphasis on generating integer feasible solutions. The optimization strategy is controlled by the CPLEX parameter *CPX_PARAM_MIPEMPHASIS* [45]. The enhanced code is tested with strategy settings of 0 (a balance between optimality and feasibility), 1 (emphasize feasibility over optimality), 2 (emphasize optimality over feasibility), and 3 (emphasize moving best bound). The reader should note that the user-defined heuristic is implemented in these tests in a slightly different manner, producing different results from the other tests.

The performance of the various settings on 5-*group* problems with misclassification limits of 15% is summarized in Table 6. There is no clear advantage to using any one of the settings in terms of number of problems solved to optimality, time to find the optimal objective value, or best objective value obtained. The strategy of emphasizing moving the best bound solves to optimality noticeably faster than other methods on *swi-0.15*. However, the best solution for *hun-0.15* obtained using the same strategy has lower objective value than that acquired using other strategies. Emphasizing optimality over feasibility is the best strategy for *va-0.15*, while a balance between optimality and feasibility proves best for *hun-0.15*.

Emphasizing moving the best bound is chosen as the best-performing method to compare to CPLEX (Section 6.2.2.1) because of its superiority in solving *swi-0.15*, its ability to find integer feasible solutions quickly, and because it is the only method that solves *hun-0.05* to optimality in the time allotted.

Table 6: Performance of enhanced code with various settings for optimization strategy. The performance of the enhanced code under various settings for *CPX_PARAM_MIPEMPHASIS*, the CPLEX parameter controlling optimization strategy. The settings tested are 0 (a balance between optimality and feasibility), 1 (emphasize feasibility over optimality), 2 (emphasize optimality over feasibility), and 3 (emphasize moving best bound). The computational data are shown for 5-*group* problems with misclassification limits set at 15%. CPLEX-generated cliques and GUBs are generated aggressively and the probing level is set to 3 (most aggressive). The optimal objective value for problem *cle-0.15* has not been verified, so the best known objective value is used.

Solution	Problem		Solutions				Branch and Cut			Total
Strategy	Data	α_{ij}	Opt Obj	Best Int Obj	Best UB (z_{UB}^{LP})	Gap Rem.(%)	Nodes	Nodes to Best	Time to Best (s)	Time (s)
0	*va*	0.15	50	50	50	0.0	23103	20481	2133	2276
1	*va*	0.15	50	50	50	0.0	13413	9874	1268	1460
2	*va*	0.15	50	50	50	0.0	10843	10351	1200	1236
3	*va*	0.15	50	50	50	0.0	11750	10528	1329	1455
0	*swi*	0.15	48	48	48	0.0	185651	152920	46558	55452
1	*swi*	0.15	48	48	48	0.0	178465	155471	40543	45900
2	*swi*	0.15	48	48	48	0.0	100784	96157	54760	56933
3	*swi*	0.15	48	48	48	0.0	84055	83831	42855	42908
0	*hun*	0.15	159	159	177.6656	25.6	275824	137630	107644	204569
1	*hun*	0.15	159	158	178.9979	27.4	225676	65142	70972	207462
2	*hun*	0.15	159	157	177.6656	25.6	154328	39291	36207	203978
3	*hun*	0.15	159	155	177.3323	25.1	144403	3588	3293	203450
0	*cle*	0.15	(133)	91	191.4371	52.2	46521	0	0	205991
1	*cle*	0.15	(133)	128	190.9943	51.8	86223	10377	23196	205540
2	*cle*	0.15	(133)	119	191.4371	52.2	70115	63847	185056	205492
3	*cle*	0.15	(133)	122	190.6545	51.5	30372	5677	1919	203077

6.3 The relative contribution of various components of the enhanced code

The enhanced code includes techniques described in Chapter 5. The enhanced code consistently outperforms CPLEX, and the relative contribution of the various components to the improved performance of the enhanced code is of interest. The current experiment evaluates the value of maximal clique cuts, maximal hyperclique cuts, non-dominated versus all maximal hyperclique cuts, odd hole cuts, the value of adding user-defined cuts locally versus globally, the heuristic, fixing variables, and the branching scheme.

6.3.1 Methods and data

The various components of the enhanced code are tested using the same instances that are used in the performance comparisons. The instances are solved 8 times, each time with a different component removed from the enhanced code. If the enhanced code performs significantly worse with a component removed, then that component can be deemed valuable.

The same machines and stopping criteria as in Section 6.2 are used. In these tests, the tolerance levels are set to their default values. The solutions are verified by the enhanced code, independent of CPLEX. If for a particular instance the solution is determined to be infeasible, then the instance is run again with the tolerance levels set at their strictest values.

Except for the test with cuts added globally, all cuts are added locally. Also, all maximal hyperclique inequalities are eligible to be added.

6.3.2 Results

The performance of the enhanced code with various components removed is compared in Figure 7. Additional information about each of the runs is in Tables 12-19 of Appendix A.

The performance of enhanced code with all cuts added locally is almost identical to the performance of the code with non-dominated hyperclique cuts and with no hyperclique cuts at all. This phenomenon is an indication that the hyperclique cuts are not helpful for solving DAMIP instances.

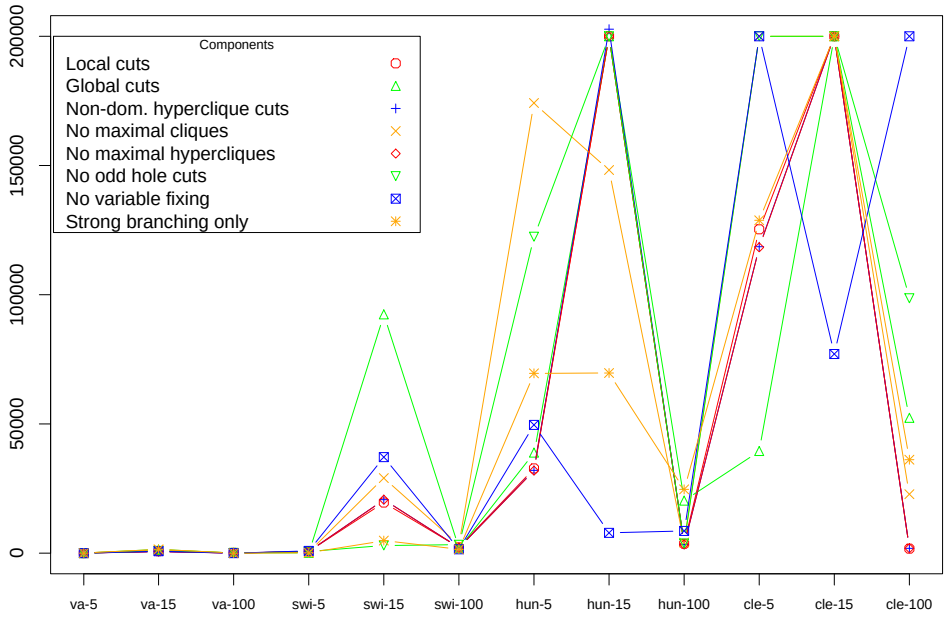


Figure 7: Time to find an integer feasible solution with the best-known objective value for the enhanced code with user-defined cuts added locally (Local cuts), user-defined cuts added globally (Global cuts), with only non-dominated hyperclique cuts (Non-dom. hyperclique cuts), without maximal clique cuts (No maximal cliques), without odd hole cuts (No odd hole cuts), without variable fixing (No variable fixing), and without the user-defined branching scheme (Strong branching only). Some solutions with the best known objective value are found using settings not included in this test.

The enhanced code with cuts added globally explore far less nodes than when cuts are added locally, but the objective values of the best integer feasible solutions found using global cuts are on par with those found using local cuts (Tables 12 and 13).

The enhanced code is among the slowest in finding the best-known integer feasible solution when maximal clique and odd hole cuts are removed (Figure 7), indicating that the cuts derived using information from the conflict graph are the most valuable for finding integer feasible solutions quickly. The best integer feasible solutions found without the cuts are not poor, perhaps due to the fact that without the cuts, the linear programs at nodes in the branch and bound tree are easier to solve which allows for the exploration of more nodes in the time allotted.

For each of the test runs, the enhanced code solves all but 4 problems to optimality with the exception of the test with maximal clique cuts removed and the test with the branching scheme removed. These two tests solve all but 5 problems to optimality.

6.4 Comparison of classification accuracy of DAMIP with standard methods

The DAMIP is tested on real-world and simulated data sets in the interest of determining an indication of its ability to accurately classify test observations under a variety of conditions. The effect of varying misclassification limits for the DAMIP is observed, and the accuracy of the DAMIP is compared to linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), and support vector machines (SVMs).

6.4.1 Methods and data

6.4.1.1 Real-world data

The classification accuracy of the DAMIP is tested using the 3-group and 5-group data sets described in Section 6.1. Observations with missing values are not included in the training or testing sets.

Ten-fold cross-validation is used to estimate the performance of the models on new data. The data sets are partitioned into 10 sets of observations of roughly the same size. The model is trained 10 times, each time withholding one of the sets as a test set. Each set of rules generated from the training sets is executed on the corresponding test set. The performance of the various classification methods is evaluated based on the ability to classify test set observations.

The proportions of training observations from each group are used as estimates for the prior probabilities π_h.

For the DAMIP, test observations are placed in the group for which their modified posterior probability is largest. The number of observations for which two or more groups have the largest modified posterior probabilities (i.e., there are ties for the largest) is recorded.

6.4.1.2 Simulated data

The design of the simulation is based on a simulation previously used to test the DALP, the linear programming approximation of the DAMIP [58]. With simulated data, the degree of data "confusion" is well-designed so as to allow us to correlate model behavior with the data characteristics.

The data are generated from bivariate (2 attributes) normal and contaminated normal distributions with different mean and variance configurations. For each run, 40 training observations from each of 3 groups are generated. The rules are then tested on 1000 test observations from each group. The process is repeated 400 times for each of 8 mean-variance configurations.

The 8 configurations for the means for the various normal distributions are given in Table 7. The configurations represent different ways of arranging three groups of data in the attribute space; i.e., two groups close together and far from the other, all three groups close together, all three groups far apart. The means and covariances are chosen such that the measure of distance, the Mahalanobis distance, is approximately 1 for groups close together and approximately 3 for groups far apart. The Mahalanobis distance between groups i and j is

$$(\mu_i - \mu_j)^T V^{-1} (\mu_i - \mu_j)$$

where μ_i, μ_j are the group means and V is the covariance matrix. For configurations $E1 - E5$, a common covariance matrix is used. For configurations $U1 - U3$, different covariance matrices are used for different groups. In $U1 - U3$, the first group's covariance matrix is diagonal $(1, 0.25)$ and the second and third groups' covariance matrices are diagonal $(0.25, 1)$. When estimating the Mahalanobis distance for configurations, $U1 - U3$, V is approximated as the average of the 2 covariance matrices in question.

For the data from contaminated normal distributions, the same configurations are used, with the exception that 10% of the data is derived from a normal distribution with the covariance matrix multiplied by 100.

The proportions of training observations from each group are used as estimates for the prior probabilities π_h.

For the DAMIP, test observations are placed in the group for which their modified posterior probability is largest. The number of observations for which two or more groups have the largest modified posterior probabilities (i.e., there are ties for the largest) is recorded.

Table 7: The mean-variance configurations for the normal distributions used in the simulation study. Configurations $E1 - E5$ use equal covariance matrices and configurations $U1 - U3$ use unequal covariance matrices.

Config.	Means			Distances		
	Group 1	Group 2	Group 3	d(1,2)	d(1,3)	d(2,3)
$E1$	(0,0)	(-0.500, 0.868)	(0.500, 0.868)	1	1	1
$E2$	(0,0)	(-1.500, 2.598)	(1.500, 2.598)	3	3	3
$E3$	(0,0)	(-1.000, 0.000)	(1.000, 0.000)	1	1	2
$E4$	(0,0)	(-0.500, 2.968)	(0.500, 2.958)	3	3	1
$E5$	(0,0)	(0.000, 2.000)	(2.905, -0.750)	2	3	4
$U1$	(0,0)	(-0.250, 0.750)	(0.250, 0.750)	1	1	1
$U2$	(0,0)	(0.000, 0.791)	(1.990, 0.395)	1	2.6	4
$U3$	(0,0)	(0.000, 0.000)	(2.000, 0.000)	0	2.5	4

6.4.1.3 Settings for the DAMIP

The data is prepared for input to the DAMIP as described in Section 5.9.

The DAMIP is used with ϵ set to 0.0001. The M values in the formulation are calculated as described in Sections 3.6 and 5.3.

All of the components of the enhanced code are used with the DAMIP, including maximal hyperclique cuts. The DAMIP is set to terminate after 7,200 CPU seconds, or after the branch and bound tree memory reached 1.9 GB. If optimality is not obtained for the DAMIP, the best-known integer feasible solution is used.

Splus 6.0 for Linux/Unix and ILOG CPLEX Callable Library 8.1 are used for generating likelihood function values and solving mixed-integer programs, respectively. The data are generated on Sun 280R's, each with 2x900MHz UltraSparc-III-Cu CPU's and 2 GB RAM. The mixed-integer programs are solved on machines with 2X2.4GHz Intel Xeon processors and 2GB RAM.

6.4.1.4 Misclassification limits

Four sets of misclassification limits are tested with the DAMIP. For each pair of groups, the proportion of misclassified training entities allowed is the same, or $\alpha = \alpha_{hg}$ for all h and g. The DAMIP is tested with limits $\alpha = 0.00$, 0.05, 0.15, and 1.00. The test with $\alpha = 0.00$ allows no misclassified training entities, and $\alpha = 1.00$ allows an unlimited number of misclassified training entities.

6.4.1.5 Other methods

Linear Discriminant Functions (LDF) The linear discriminant function coefficients are derived using the Splus 6.0 for Linux/Unix function *discrim()* with the homoscedastic covariance structure. These coefficients maximize the probability of correct classification for data that is normally distributed with equal covariance among the groups. The coefficients are obtained based on training set data. The values of the discriminant functions for each observation are used as estimates for the conditional group density function values $f_h(x^{gj})$. The "Bayes rule", allocation to the group for which the estimated posterior probability

$\pi_h f_h(x^{gj})$ is largest, is applied to the test set data in each fold to obtain a measure of the performance of LDF.

Quadratic Discriminant Functions (QDF) The quadratic discriminant function coefficients are derived using the Splus 6.0 for Linux/Unix function *discrim()* with the heteroscedastic covariance structure. These coefficients maximize the probability of correct classification for data that is normally distributed with different covariance matrices for each group. The Bayes rule is applied to the test set data in each fold to obtain a measure of the performance of QDF. The values of the discriminant functions for each observation are used as estimates for the conditional group density function values $f_h(x^{gj})$. The "Bayes rule", allocation to the group for which the estimated posterior probability $\pi_h f_h(x^{gj})$ is largest, is applied to the test set data in each fold to obtain a measure of the performance of QDF.

Classification Trees (CART) Each of the training data sets is passed through the Splus 6.0 for Linux/Unix function *tree()* with default settings. The default settings terminate tree growth when a node is either homogeneous (meaning that all entities at that node are from the same group) or contains less than 6 observations. Given an observation, the proportions of training observations at the appropriate leaf nodes are used as estimates for the conditional group density function values. The "Bayes rule", allocation to the group for which the estimated posterior probability $\pi_h f_h(x^{gj})$ is largest, is applied to the test set data in each fold to obtain a measure of the performance of CART.

Support Vector Machines (SVM) SVM^{multiclass} [47] is used for generating classification rules using support vector machines. Linear and radial basis function kernels are used. Various values for c, the relative emphasis in the objective on minimizing training error and maximizing the margin between groups, and the width g of the radial basis function kernel were tested. The results are summarized in Appendix B. For the subsequent tests c is set to 0.1 for the linear kernel and 0.01 for the radial basis function kernel. A width of $g = 1.0$ is used for the radial basis function kernel.

Calculations for LDF, QDF, and CART are performed on Sun 280R's, each with 2x900MHz UltraSparc-III-Cu CPU's and 2 GB RAM. The support vector machine code is executed on machines with 2x2.4GHz Intel Xeon processors and 2GB RAM.

For all of the classification methods, if there is no unique maximum discriminant function value for an entity, then the entity is placed in the reserved judgment category.

6.4.2 Results

The various classification methods are evaluated using classification matrices as well as summary measures. The summary measures include correct classification rate, misclassification rate, the rate of non-reservation, and accuracy. They are calculated as follows

$$\begin{aligned} C &= 100 \times \frac{number\ of\ correctly\ classified\ entities}{total\ number\ of\ entities} \\ I &= 100 \times \frac{number\ of\ misclassified\ entities}{total\ number\ of\ entities} \\ N &= 100 \times \frac{number\ of\ non{-}reserved\ entities}{total\ number\ of\ entities} \\ A &= 100 \times \frac{number\ of\ correctly\ entities}{number\ of\ non{-}reserved\ entities} \end{aligned}$$

Note that accuracy is defined as the percentage of correctly classified observations of those not placed in the reserved judgment region.

6.4.2.1 Real-world data

The accuracy, misclassification, correct classification, and non-reserved rates for the various methods on real-world data are given in Figure 8. The classification matrices for these tests are in Figures 22 through 26 of Appendix B. The DAMIP with misclassification limits consistently has lower rates of misclassification, though for some of the less well-separated data sets, large portions of the test observations are placed in the reserved judgment region.

The data sets *wine*, *iris*, and *new-thyroid* are well-separated data sets, as indicated by the high accuracy and low misclassification rates for all methods. Support vector machines tend to place most test observations into one group which reduce its accuracy, particularly on the *wine*, *FNlnVN*, and *swi* data sets. The accuracy of the methods appears to be a function of the data set rather than the method, whereas the DAMIP is able to provide rules with consistently lower misclassification rates by using the reserved judgment region.

All of the methods perform poorest on the data set *FNlnVN*. The misclassification rates are as high or higher than for some of the 5-group data sets that are more computationally intensive for the DAMIP. To further investigate the problem of discriminating between cells placed on fibronectin, laminin, and vitronectin, consider the classification matrices in Figure 9. The results are presented in classification matrices that indicate how entities from each group are allocated. The quantity in row i and column $j+1$ is the proportion of training entities from group i allocated to group $j+1$. The quantity in row i and column 1 is the proportion of training entities from group i allocated to the reserved judgment region.

LDF, QDF, CART, and DAMIP have high rates of correct classification of cells placed in co-culture with Vn. SVM with a linear kernel has significantly more trouble, placing 41.7% of Vn cells in the Ln group. SVM with a radial basis function kernel places most of the observations in the Fn group, misclassifying all cells placed on Vn.

All of the methods encounter difficulty in discriminating between cells placed on Fn and cells placed on Ln. The misclassification rate is minimized by DAMIP as the limit on misclassified training entities is lowered. The high rate of placement of observations in the reserved judgment region is an indication of overlapping attribute values and therefore indistinguishable behavior.

The effects of the misclassification limits of the DAMIP are further explored in Figure 10. For $\alpha = 1.00$, the DAMIP will perform at least as well on the training set as the method used in estimating likelihood function values. This advantage does not translate to significantly higher rates of correct classification for DAMIP over LDF. In fact, the four summary measures are almost identical.

With the exception of the *hun* dataset, the accuracy on the test sets decreases monotonically as the misclassification limits are raised. The correct classification, misclassification, and non-reservation rates increase as the limits are increased. All four measures appear to converge to the values obtained by LDF. When the misclassification limits are set to 0% on the training set, the misclassification rate on the test set is less than 20% for all data sets. With the exception of the *FNlnVN*, *swi*, and *va* datasets, the misclassification rates are less than 10%. The range of misclassification rates when the limits are set to 5% is 1.8-37.0%,

and when the limits are 15%, the range is 1.8-59%.

Perhaps a better measure of the reduction in misclassification of the DAMIP for difficult data sets is the difference between the misclassification rate when the limits are 100% (which is roughly the performance of LDF) and the rates when the limits are lower. For *FNlnVN*, *swi*, and *va*, the improvement in misclassification when limits are set to 0% is a 37-57% reduction in misclassification. The 5% limits afford improvements of 24-51%, and the 15% limits result in improvements of 2-31%.

The number of observations for which two or more groups have the largest modified posterior probabilities (i.e., there are ties for the largest) is 0 for all data sets and misclassification limits for the DAMIP. This result indicates that the DAMIP is stable under these settings. The stability derives in large part due to the method for estimating the conditional group density function values.

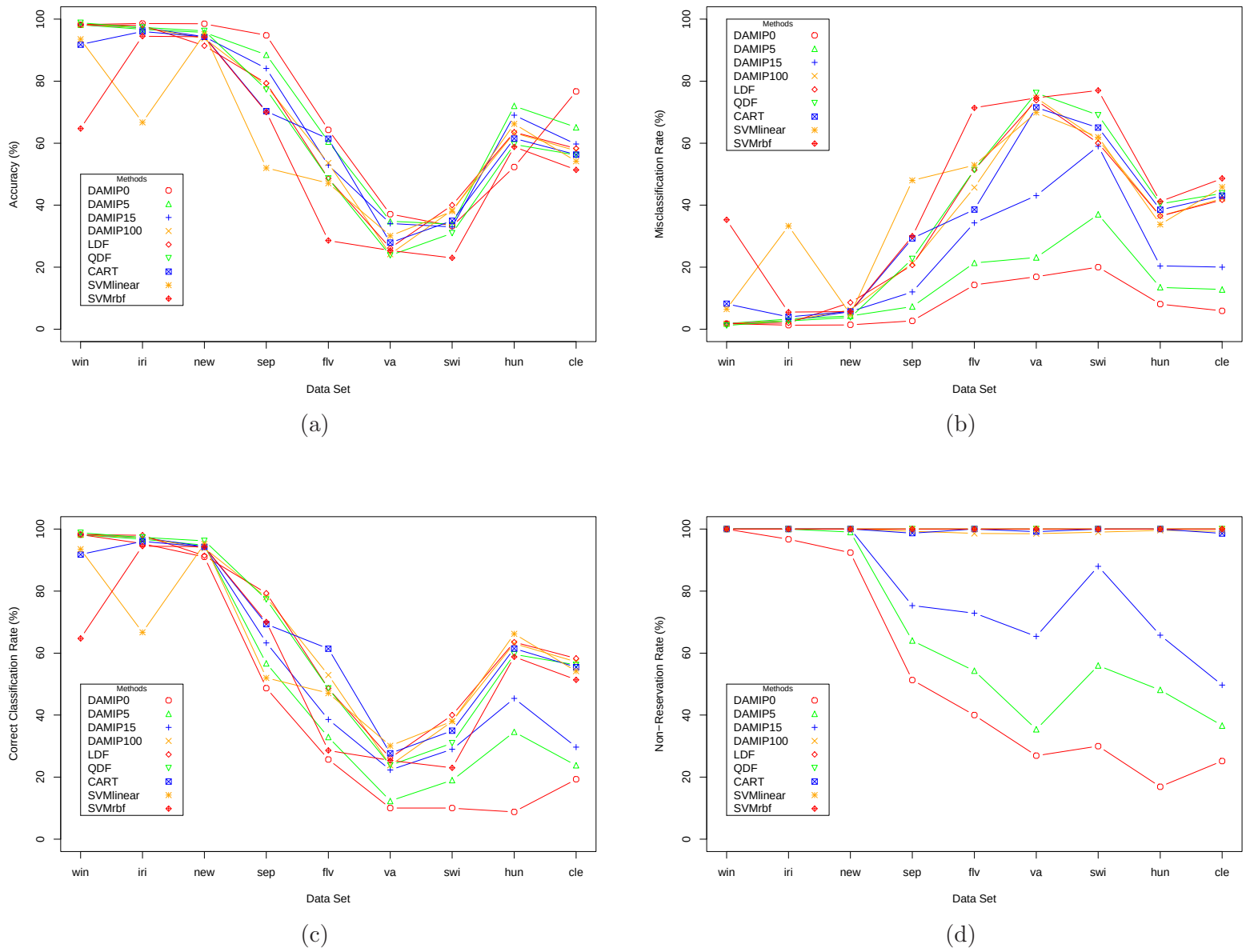
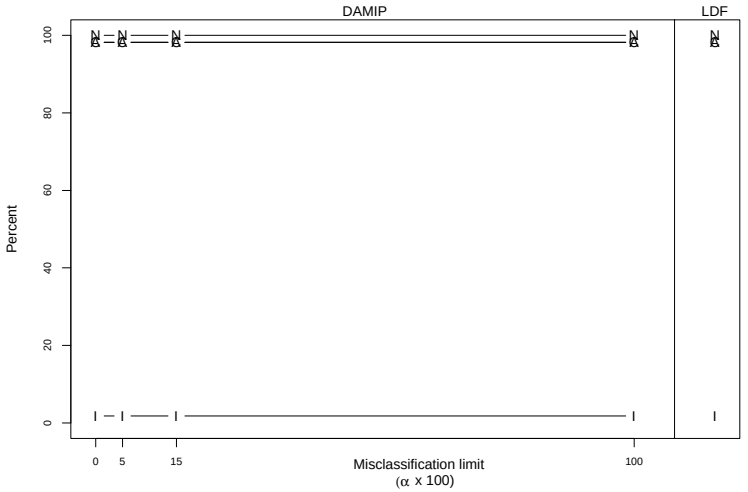


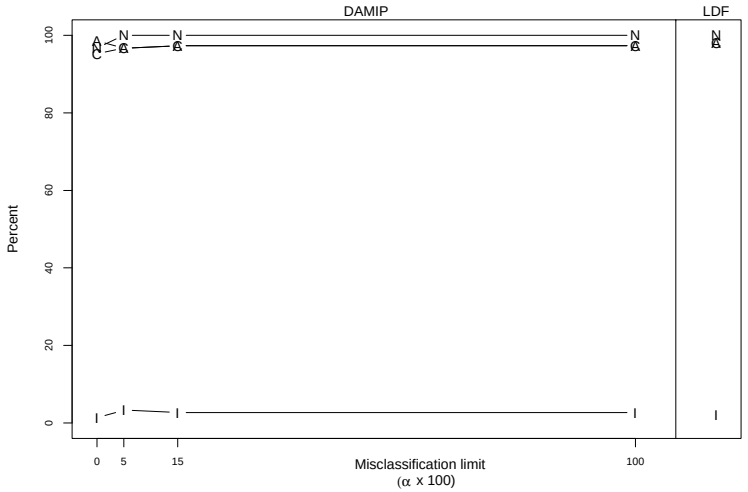
Figure 8: (a) Accuracy, (b) misclassification rates, (c) correct classification rates, and (d) non-reservation rates for various methods on real-world data sets. The DAMIP is tested with misclassification limits of 0 (DAMIP0), 5 (DAMIP5), 15 (DAMIP15), and 100% (DAMIP100). Other methods tested are linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), and support vector machines with a radial basis function kernel (SVMrbf). The data sets are described in Section 6.1.

DAMIP with misclassification limits																				
0%					5%				15%				100%							
	62.5	12.5	12.5	12.5		54.2	16.7	20.8	8.3		20.8	25.0	33.3	20.8		4.2	41.7	33.3	20.8	
	62.5	12.5	20.8	4.2		45.8	25.0	25.0	4.2		29.2	25.0	37.5	8.3		0.0	41.7	37.5	20.8	
	54.5	0.0	0.0	45.5		36.4	4.5	0.0	59.1		31.8	4.5	9.1	54.5		0.0	4.5	13.6	81.8	
LDF					QDF				CART				SVMlinear				SVMrbf			
0.0	37.5	41.7	20.8		0.0	20.8	58.3	20.8		0.0	41.7	45.8	12.5		0.0	33.3	41.7	25.0		0.0	83.3	16.7	0.0
0.0	50.0	41.7	8.3		0.0	25.0	58.3	16.7		0.0	29.2	58.3	12.5		0.0	41.7	54.2	4.2		0.0	87.5	0.0	12.5
0.0	13.6	18.2	68.2		0.0	18.2	13.6	68.2		0.0	0.0	13.6	86.4		0.0	4.5	31.8	63.6		0.0	86.4	13.6	0.0

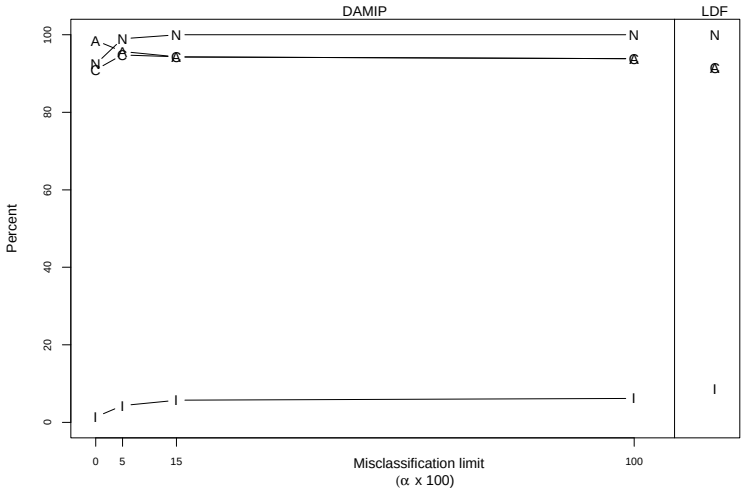
Figure 9: Classification matrices for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j+1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified.



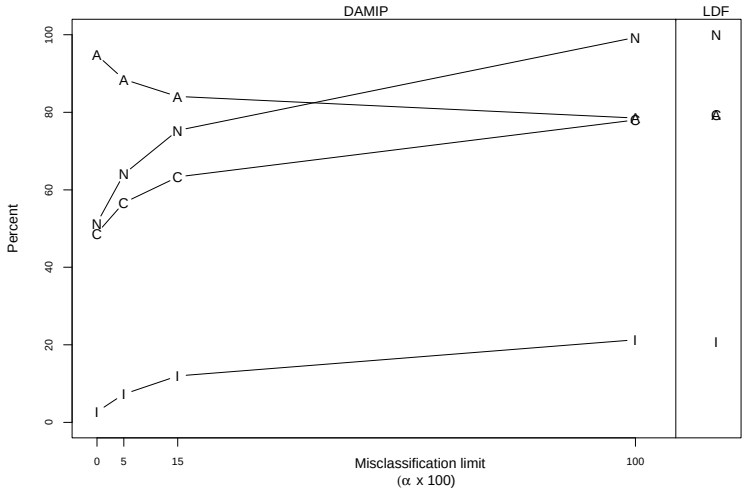
(a) *wine*



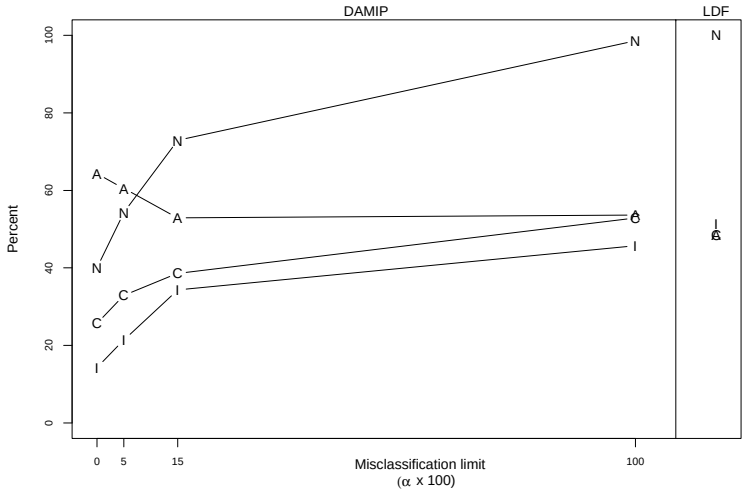
(b) *iris*



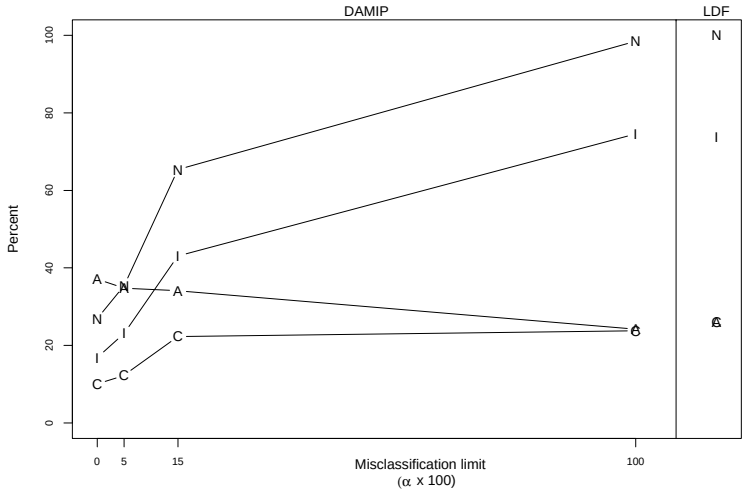
(c) *new-thyroid*



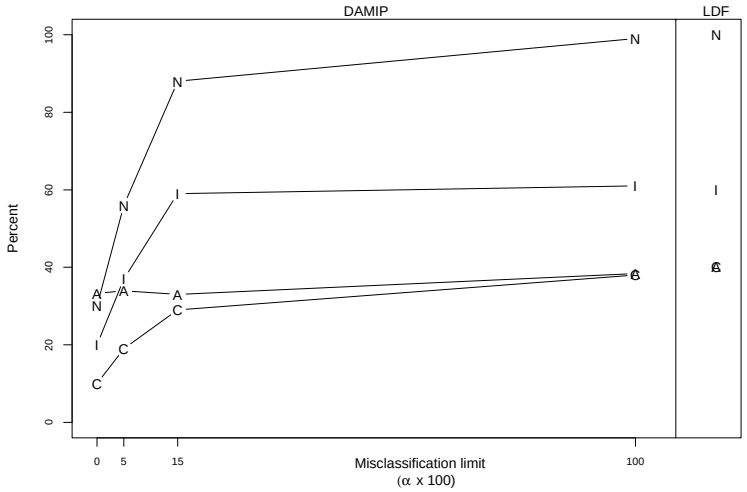
(d) *sepal*



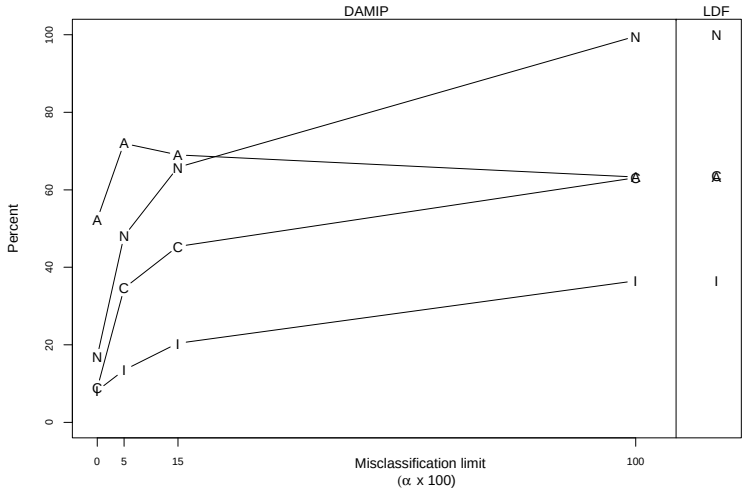
(e) *FNlnVN*



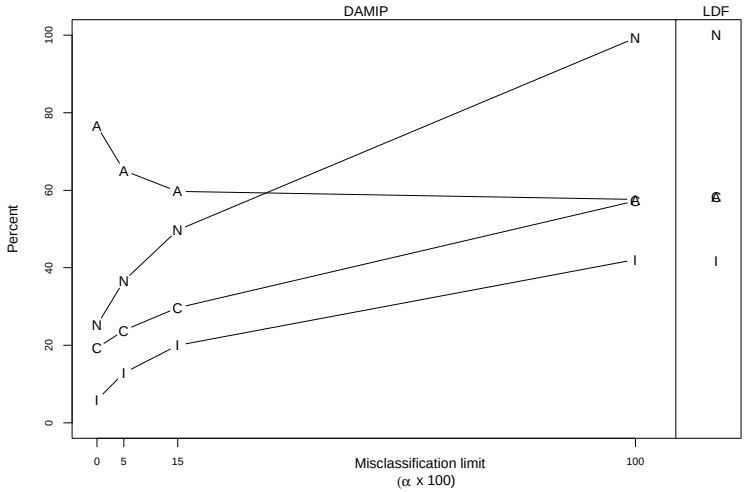
(f) *va*



(g) *swi*



(h) *hun*



(i) cle

C mean percentage of correctly classified entities
I mean percentage of misclassified entities
N mean percentage of non-reserved entities
A mean accuracy

Figure 10: Correct classification percentage, misclassification percentage, percentage of non-reserved entities, and accuracy of DAMIP with misclassification limits of 0, 5, 15, and 100% compared to linear discriminant functions (LDF). The data sets are described in Section 6.1.

6.4.2.2 Simulated data

The accuracy, misclassification, correct classification, and non-reservation rates for the various discrimination methods and simulated data are given in Figures 11 and 12. The classification matrices for configurations $E1$ and $U1$ with non-contaminated data are contained in Appendix B in Figure 30. As with the real-world data, DAMIP with misclassification limits has consistently higher rates of accuracy and lower rates of misclassification. Quadratic discriminant functions produce the highest correct classification rates among rules without a reserved judgment region, while classification trees and support vector machines with a linear kernel generally produce the lowest correct classification rates.

The DAMIP with misclassification limits of 0% of training observations translate into rates of misclassification of test observations consistently under 5%. The DAMIP with misclassification limits of 5% is equally consistent, producing misclassification rates between 9.5% and 14%. These methods have low correct classification rates due to the large numbers of observations placed in the reserved judgment region. The proportion of observations placed in the reserved judgment region varied widely for each of the methods, depending on the configuration for generating the simulated data. Data sets having larger Mahalanobis distances between groups result in lower numbers of test observations placed in the reserved judgment region, and vice versa.

The DAMIP with misclassification limits of 15% produces desirable results in that the correct classification rates are competitive with other methods, the misclassification rates are generally lower, accuracy is generally higher, and the reservation rate is never more than 35%.

The results for the DAMIP with misclassification limits are replicated for the data generated from contaminated normal distributions, except that the reservation rate is much higher for DAMIP with misclassification limits of 0 and 5% (Figure 12). The reservation rates for DAMIP with limits of 15% is largely unaffected, perhaps because only 10% of the data are contaminated. The relationship between the misclassification limits of 15% and the 10% contamination level is not obvious; however, the event that DAMIP with less-restrictive misclassification limits places less test observations in the reserved judgment region with

contamination is both plausible and desirable.

Plots of the summary measures as a function of the misclassification limits for simulated data from normal and contaminated normal distributions are in Figures 13 and 14. As with the real-world data, the summary measures converge to the values obtained by LDF alone as the misclassification limits are raised for the DAMIP. In general, the accuracy decreases and the misclassification, correct classification, and non-reservation rates increase to the values derived using LDF alone.

The simulated data is generated in the same manner as described as in [58], where the classification performance of the DALP (a linear programming approximation of the DAMIP) is evaluated. The plots in Figures 13 and 14 can be easily compared to the plots in [58]. The curves are very similar in that, for the DAMIP with misclassification limits of 5, 15, and 100%, there is a configuration for the DALP for which the four summary measures are within 5% of the values recorded for the DAMIP.

The DAMIP with misclassification limits of 15% performed only marginally better than the best-performing DALP configuration in terms of higher accuracy and non-reservation rates, and lower rates of misclassification [58]. The subtle differences in performance are to be expected in part because the data are generated from normal distributions and the calculation of the conditional group density functions for DALP and DAMIP assumes that the data are normally distributed. In other words, much of the potential accuracy is achieved before the data reaches the DALP or DAMIP, respectively.

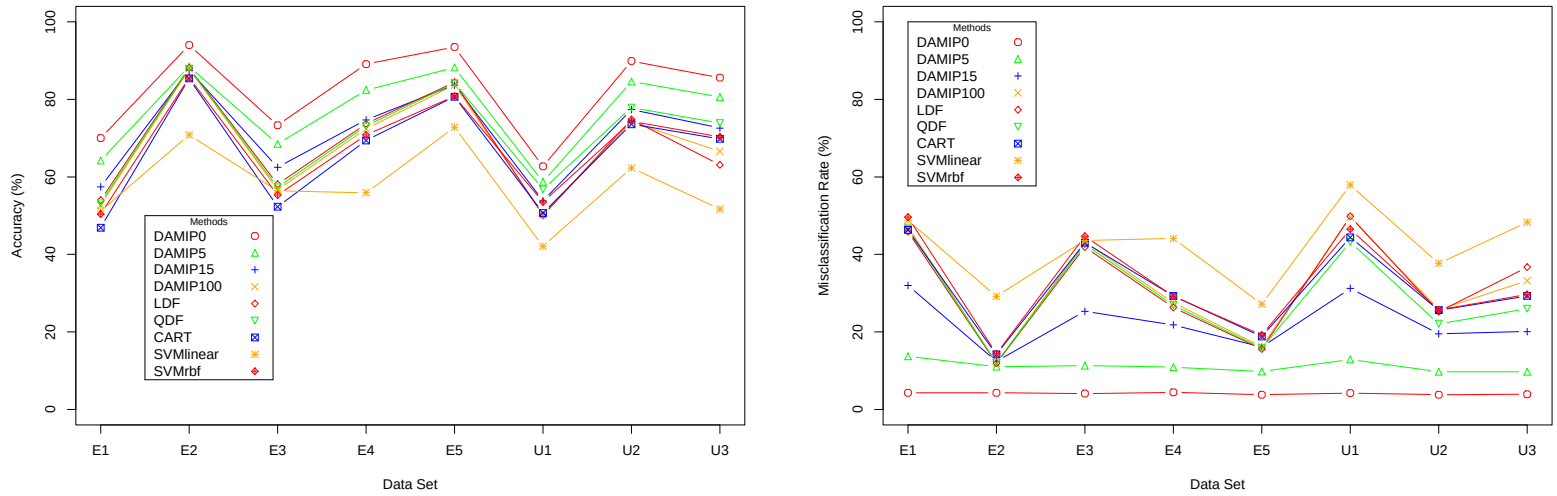
A noticeable difference in the performance of the DAMIP and DALP is that for DAMIP with 0% limits, accuracy is generally higher, misclassification rates are lower, and non-reservation rates are much lower than for any configuration under the DALP. The DAMIP with strict misclassification limits for the training sets provides greater control over the levels of misclassification seen in the test sets.

The differences in performance between DAMIP on the data from normal and contaminated normal distributions are more pronounced for misclassification limits of 0 and 5% than for misclassification limits of 15 and 100%. The accuracy decreases for the data from contaminated normal distributions for the DAMIP with each misclassification limit, but

less so for limits of 15 and 100% than for limits of 0 and 5%.

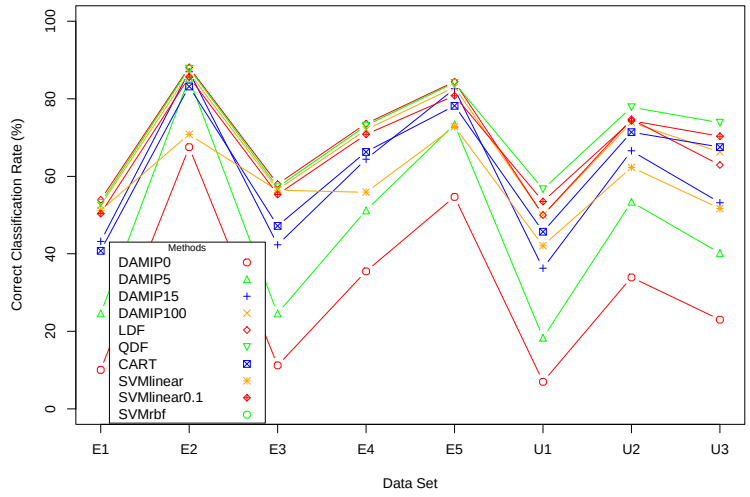
The accuracy and misclassification rates on the contaminated data for DAMIP with limits of 0% are particularly disappointing. The accuracy and non-reservation rates are noticeably lower when the data is contaminated. Contamination levels of 10% are sufficient to undermine the performance of the DAMIP with 0% misclassification limits.

The number of observations for which two or more groups have the largest modified posterior probabilities in these simulations (i.e., there are ties for the largest) is 423 out of $76,800,000$ test observations for all misclassification limits for the DAMIP. There are 43 ties for the non-contaminated data and 380 ties for the contaminated normal data. This result indicates that the DAMIP is stable under these settings. The stability derives in large part due to the method for estimating the conditional group density function values.

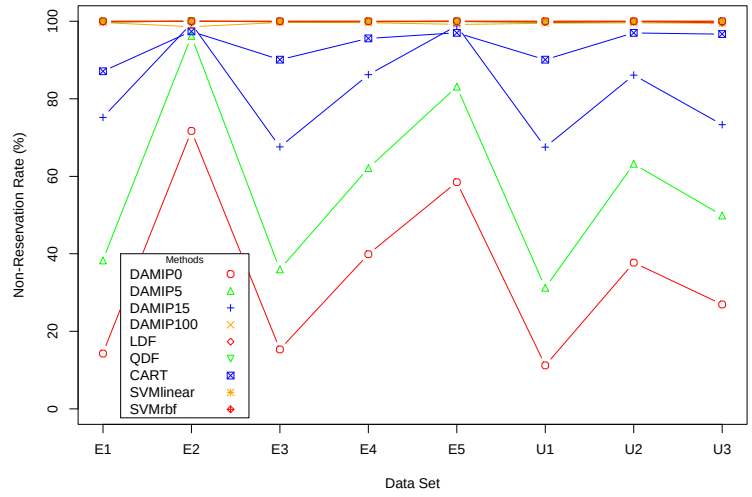


(a)

(b)



(c)



(d)

Figure 11: (a) Accuracy, (b) misclassification, (c) correct classification, and (d) non-reservation rates for various methods on simulated data sets. The data are generated from bivariate normal distributions with different mean and covariance configurations which are described in Table 7 and Section 6.4.1.2. The DAMIP is tested with misclassification limits of 0 (DAMIP0), 5 (DAMIP5), 15 (DAMIP15), and 100% (DAMIP100). Other methods tested are linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), and support vector machines with a radial basis function kernel (SVMrbf).

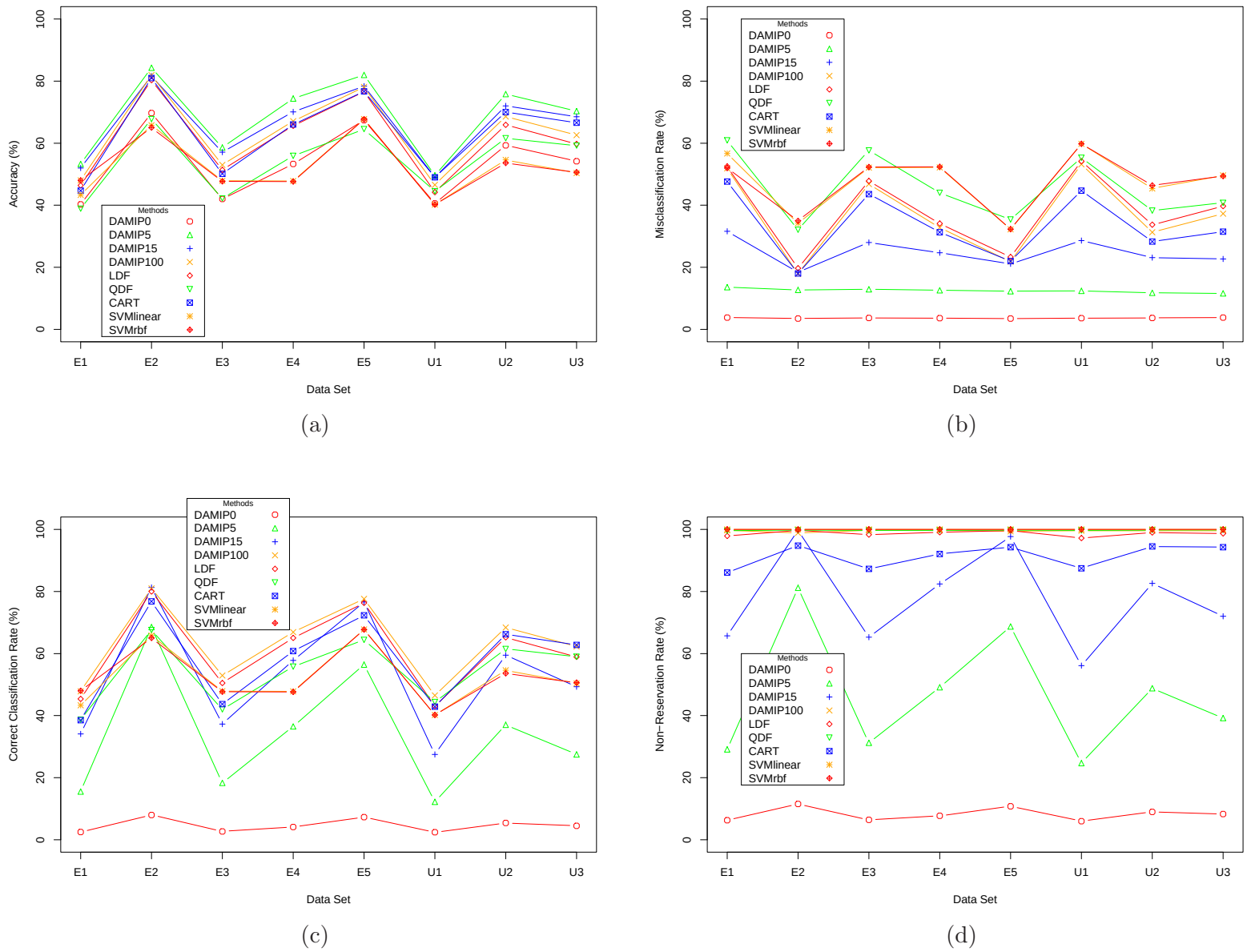
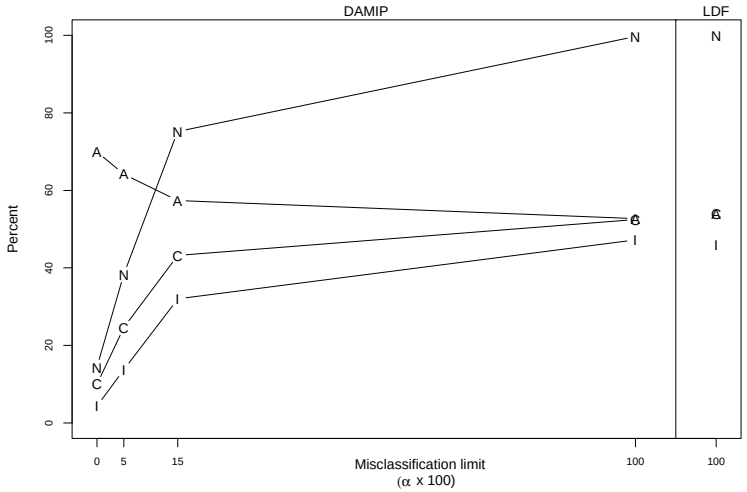
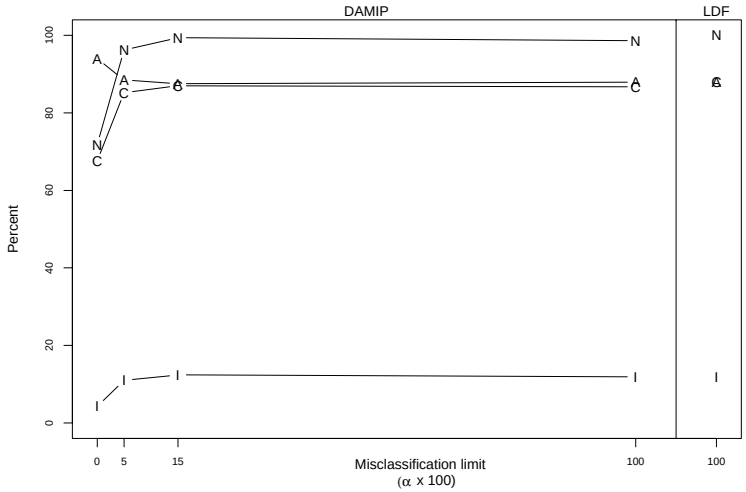


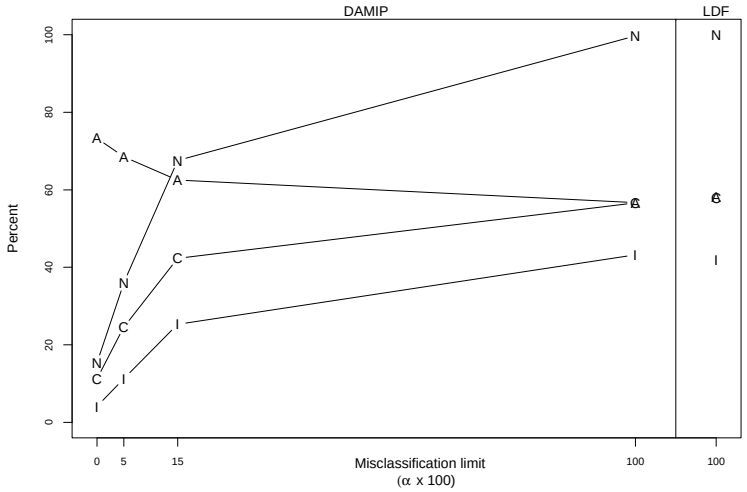
Figure 12: (a) Accuracy, (b) misclassification, (c) correct classification, and (d) non-reservation rates for various methods on simulated data sets. The data are generated from contaminated bivariate normal distributions with different mean and covariance configurations which are described in Table 7 and Section 6.4.1.2, except that 10% of the data in each group are generated using covariance matrices 100 times the matrix in the table. The DAMIP is tested with misclassification limits of 0 (DAMIP0), 5 (DAMIP5), 15 (DAMIP15), and 100% (DAMIP100). Other methods tested are linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), and support vector machines with a radial basis function kernel (SVMrbf).



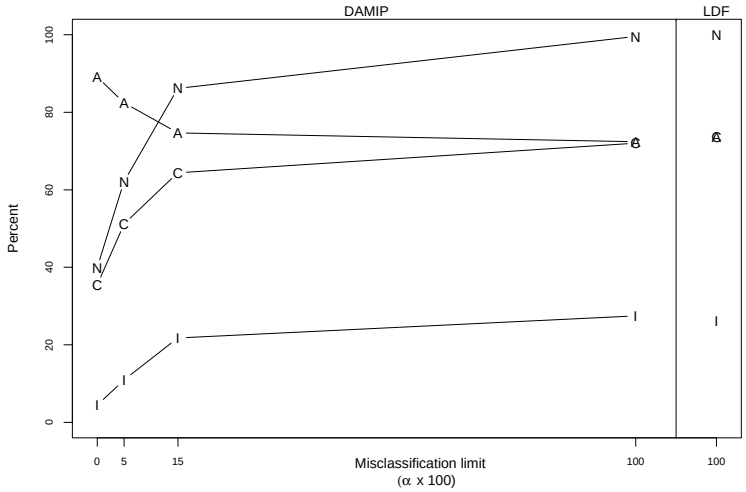
(a) $E1$



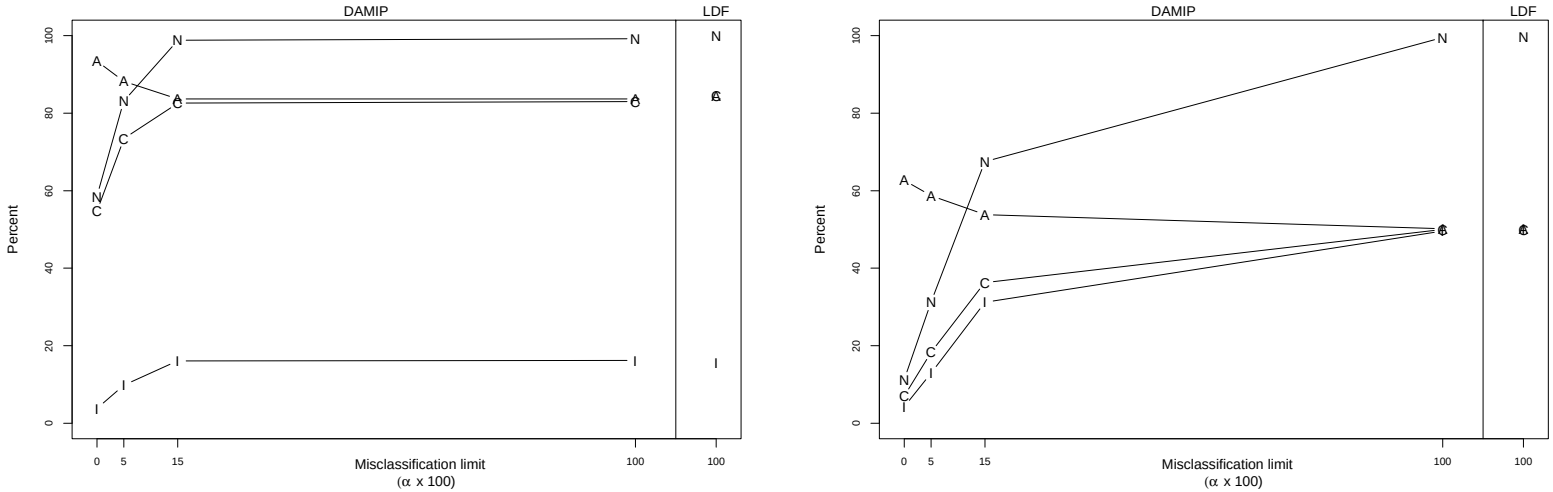
(b) $E2$



(c) $E3$

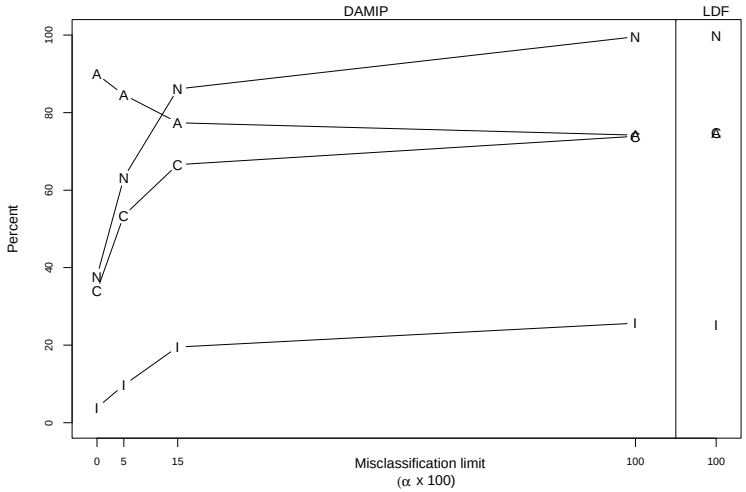


(d) $E4$

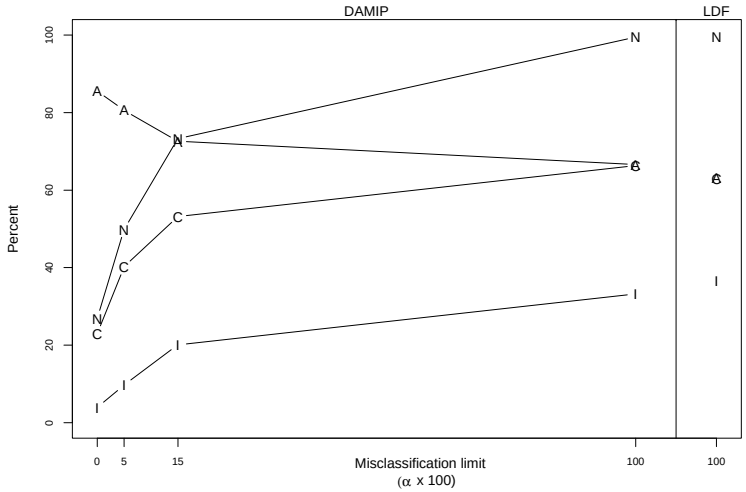


(e) $E5$

(f) $U1$

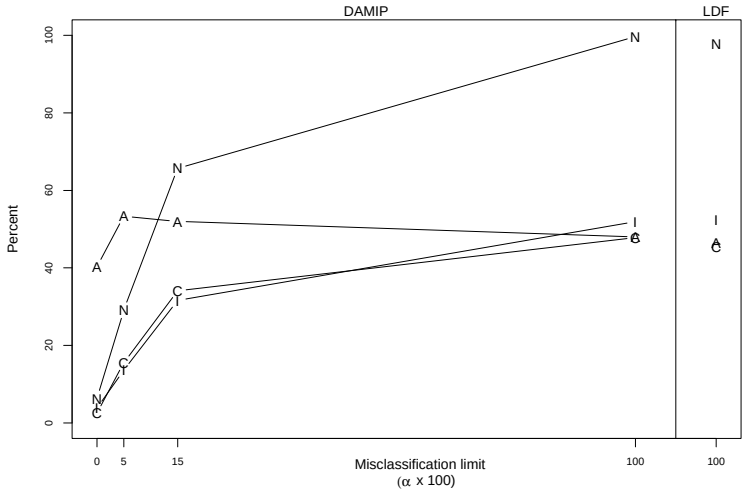


(g) $U2$

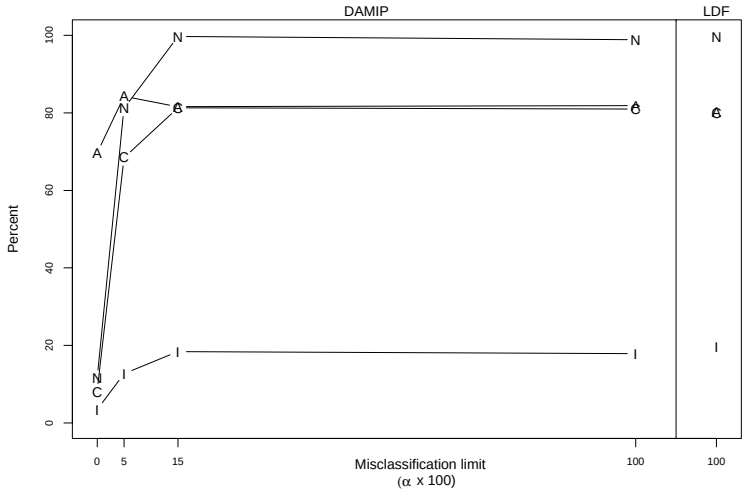


(h) $U3$

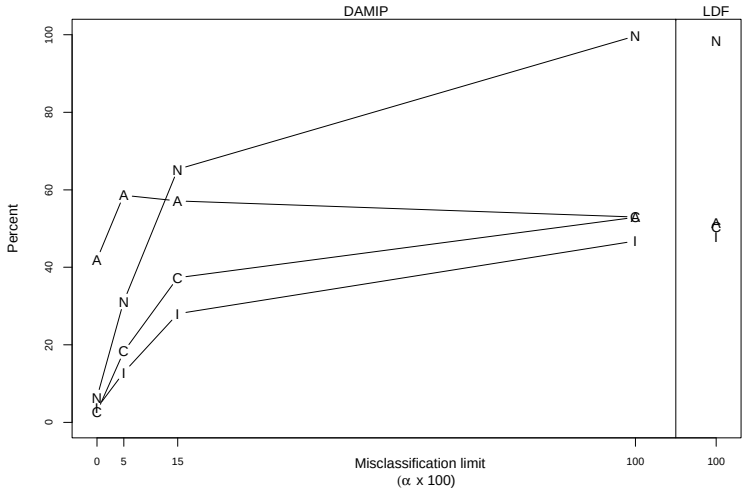
Figure 13: Correct classification percentage, misclassification percentage, percentage of non-reserved entities, and accuracy of DAMIP with misclassification limits of 0, 5, 15, and 100% compared to linear discriminant functions (LDF). The data are generated from bivariate normal distributions with mean-covariance configurations as described in Table 7 and Section 6.4.1.2.



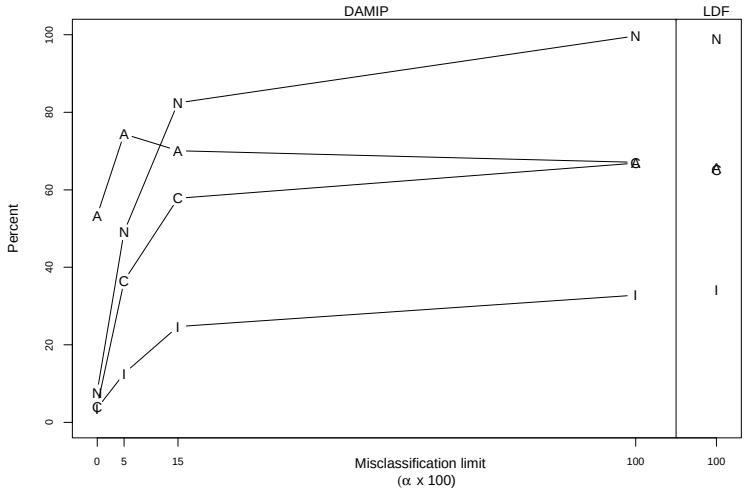
(a) $E1$



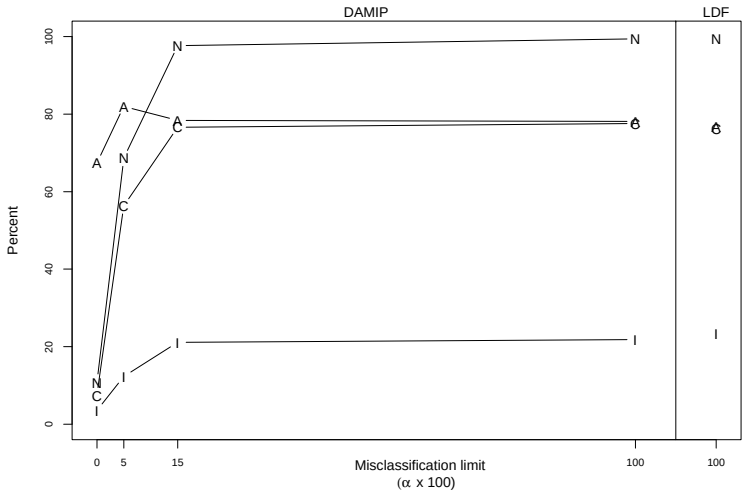
(b) $E2$



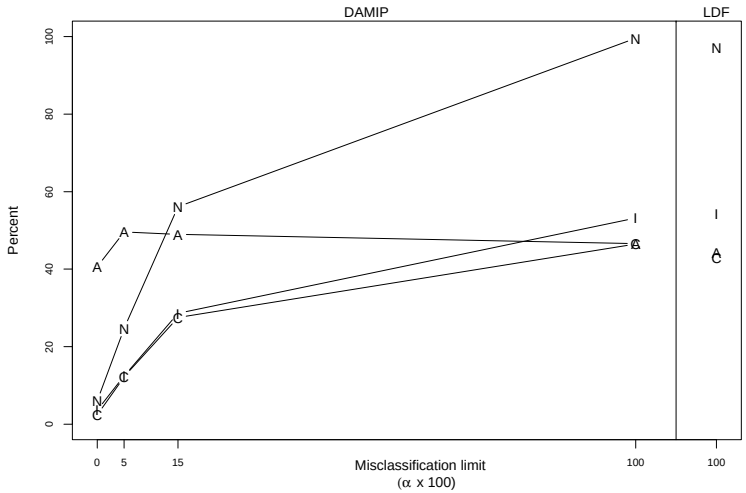
(c) $E3$



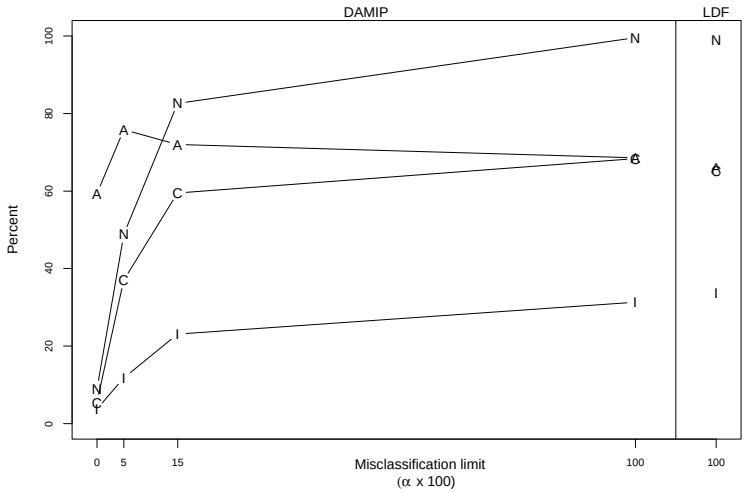
(d) $E4$



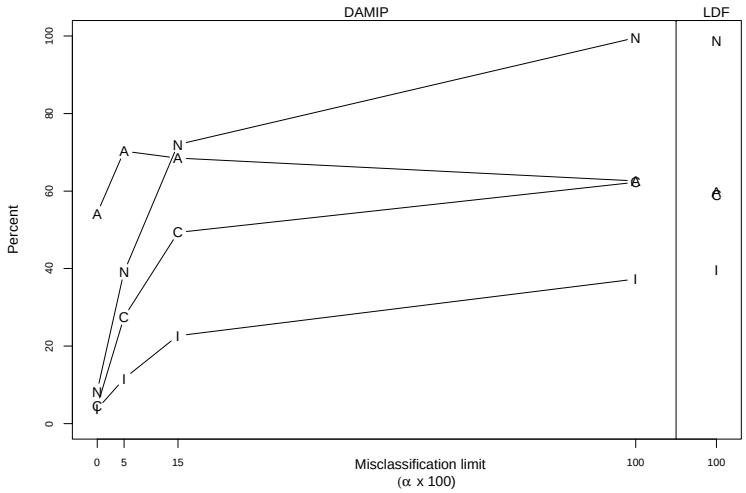
(e) $E5$



(f) $U1$



(g) $U2$



(h) $U3$

Figure 14: Correct classification percentage, misclassification percentage, percentage of non-reserved entities, and accuracy of DAMIP with misclassification limits of 0, 5, 15, and 100% compared to linear discriminant functions (LDF). The data are generated from contaminated bivariate normal distributions with mean-covariance configurations as described in Table 7 and Section 6.4.1.2, except that for 10% of the data in each group, the covariance matrix is multiplied by 100.

6.5 The effect of different sample sizes and group sizes on classification accuracy

In Section 4.1, the consistency of the DAMIP under certain conditions is established, demonstrating that larger sample sizes are desirable for obtaining classification rules that are close to the "Anderson-optimal" strategy. Using simulated data with different training sizes, one can witness this convergence empirically.

In addition to varying sample sizes, the relative proportions of observations from different groups in a training sample can affect the classification rules determined by the DAMIP. The proportions directly affect the rules through the definitions of the prior probabilities, and indirectly through the objective function. The latter effect results from the DAMIP seeking to maximize the number correctly classified observations, which will give preference to groups with higher numbers of training observations. The misclassification limits, which are imposed for every pair of groups, can help to keep this bias in check.

The dependence of the classification rules on sample size and the relative group sizes is investigated empirically using simulated data that are generated from normal distributions.

6.5.1 Methods and data

The data used in the simulations are generated from normal distributions with configurations $E1$ and $U1$ as described in Table 7. These data sets are among the more difficult to achieve high accuracy due to the fact that the Mahalanobis distances between pairs of groups is approximately 1. The settings for the DAMIP, LDF, QDF, CART, and SVMs are as described in Section 6.4.1.2. Additionally, the same machines are used for the calculations.

Simulations are performed using three types of simulated data, with a total of 15 sets of simulated data for each of $E1$ and $U1$. The first set consists of training data with equal numbers of observations from each of 3 groups. The second set consists of training data with more observations from the first group than from the other two groups. The third set consists of training data with more observations from the first and second groups than from the third group. The configurations are given in Table 8.

Table 8: The numbers of observations from each group in the training sets for the simulation study. Data is generated under configurations $E1$ and $U1$ for each set of group sizes.

Equal group sizes	One large group	Two large groups
5/5/5	100/5/5	100/100/5
15/15/15	100/10/10	100/100/10
25/25/25	100/15/15	100/100/15
40/40/40	100/30/30	100/100/30
100/100/100	100/50/50	100/100/50

For the DAMIP, test observations are placed in the group for which their modified posterior probability is largest. The number of observations for which two or more groups have the largest modified posterior probabilities (i.e., there are ties for the largest) is recorded.

The test data sets, as in previously described simulations, consist of 1000 observations from each group. For each group size configuration and each of $E1$ and $U1$, training of the classification model and testing of the rules is repeated 400 times.

6.5.2 Results

The classification matrices for each of the simulations is contained in Appendix B.

The accuracies of various classification methods for the simulations for which the training group sizes are equal are in Figure 15. For each of the classification methods, the accuracy increases slightly with the sample size. All of the methods performed better than expected on sample sizes as low as 15, and the improvement when the sample size is increased to 300 is not remarkable. The improvement seen for each classification method is rather uniform, as indicated by the non-intersecting curves in the graph.

The accuracies for the simulations with varying training group sizes are in Figures 16 and 17. The DAMIP with misclassification limits has significantly higher accuracy than other methods when the number of observations from each group is significantly different. All of the methods demonstrate detectable but not large improvement in accuracy as the proportions are leveled.

To investigate further the effects of different group sizes on the accuracy of the DAMIP, the accuracy of test observations from each group is plotted in Figures 18 and 19. In the graphs, as is to be expected, the accuracy for observations from the group(s) with the largest number(s) of training observations is(are) significantly higher. These effects are minimized when the misclassification limits are set to 15%. The DAMIP with misclassification limits of 5% also shows better ability to handle differences in training group sizes than LDF. For DAMIP with no misclassification limits and for LDF, the difference in accuracies across the groups is considerable.

A more in-depth investigation of the effects of varying group sizes can be achieved by considering the classification matrices in Appendix B. Consider first the simulations with equal numbers of training observations (Figures 27 through 31). For methods without a reserved judgment group, including DAMIP with no misclassification limits (DAMIP with

limits of 100%), the accuracy and correct classification of observations increases gradually or remains constant as the sample size increases. The accuracy, misclassification, correct classification, and non-reservation rates for data from distribution $E1$ are slightly better than those for data from distribution $U1$ (Table 7). Interestingly, the accuracy of methods for observations from group 1 increases for all methods except for SVMs with a linear kernel, for which the group 1 accuracy decreases.

For DAMIP with misclassification limits of 0, 5, and 15%, the accuracy increases with sample size, while misclassification rates decrease. For limits of 0 and 5%, this improved performance includes a tradeoff, as the correct classification rates decrease and the non-reservation rates decrease with increases in sample size. For limits of 15%, the reservation rates and correct classification rates remain relatively constant, while accuracy increases and misclassification decreases (Figures 27 through 31).

Figures 32 through 36 contain the classification matrices for simulations where the number of group 1 training observations is much larger than the numbers of training observations from groups 2 and 3. As expected, methods without misclassification limits tend to place more test observations in group 1 than the other two groups, especially when the numbers of group 2 and 3 training observations are extremely low. As the numbers of group 2 and 3 training observations is increased, most method correctly classify group 2 and 3 test observations at higher rates, and group 1 observations at lower rates. An exception is support vector machines with a linear kernel, which has significantly higher correct classification rates for groups 2 and 3 compared to other methods without misclassification limits.

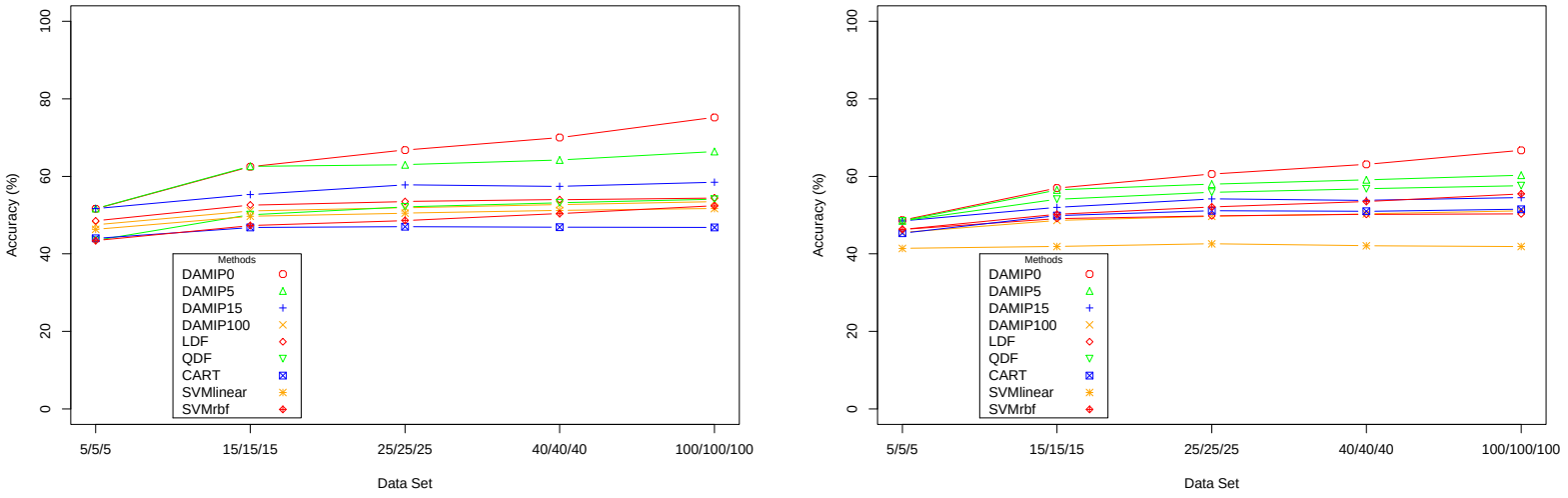
DAMIP with misclassification limits of 0 and 5% have increasing numbers of test observations placed in the reserved judgment region as the proportions of training observations is leveled. The accuracy of these methods increases and misclassification decreases as the numbers of group 2 and 3 training observations are increased. DAMIP with misclassification limits of 15% has decreasing numbers of observations placed in the reserved judgment region as the proportions are evened, while the accuracy increases and misclassification rates decrease. The DAMIP with misclassification limits clearly offers an alternative to traditional methods in terms of accuracy for groups underrepresented in the training set,

while DAMIP with misclassification limits of 15% provides not only increased accuracy, but lower misclassification rates as well.

For simulations with large proportions of training observations from 2 of the 3 groups, the classification matrices are in Figures 37 through 41. The traditional methods tend to correctly classify large numbers of test observations from the groups with larger training observations, especially when the differences in the proportions are more pronounced. Support vector machines with a linear kernel does not demonstrate the same ability to correctly classify observations from groups underrepresented in the training set as it does in other simulations.

DAMIP with misclassification limits of 5 and 15% correctly classify significant numbers of observations in group 3, the group underrepresented in the training set, compared to traditional methods. For DAMIP with limits of 15%, the reservation rates and misclassification rates again decrease as the proportions of training observations are leveled while the accuracy increases. For misclassification rates of 0 and 5%, the reservation and accuracy rates remain approximately constant as the proportions are varied.

The number of observations for which two or more groups have the largest modified posterior probabilities (i.e., there are ties for the largest) for these simulations is 79 out of $163,200,000$ test observations for all misclassification limits for the DAMIP. This result indicates that the DAMIP is stable under these settings. The stability derives in large part due to the method for estimating the conditional group density function values.



(a) (b)

Figure 15: The accuracy of various classification methods on data generated from distributions (a) $E1$ and (b) $U1$ as described in Table 7. The values on the x-axis indicate the numbers of training observations for each of the three groups used in generating classification rules. The accuracy of the methods is the average performance on test observations. The DAMIP is tested with misclassification limits of 0 (DAMIP0), 5 (DAMIP5), 15 (DAMIP15), and 100% (DAMIP100). Other methods tested are linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), and support vector machines with a radial basis function kernel (SVMrbf). Accuracy is defined as the percentage of non-reserved observations that are correctly classified.

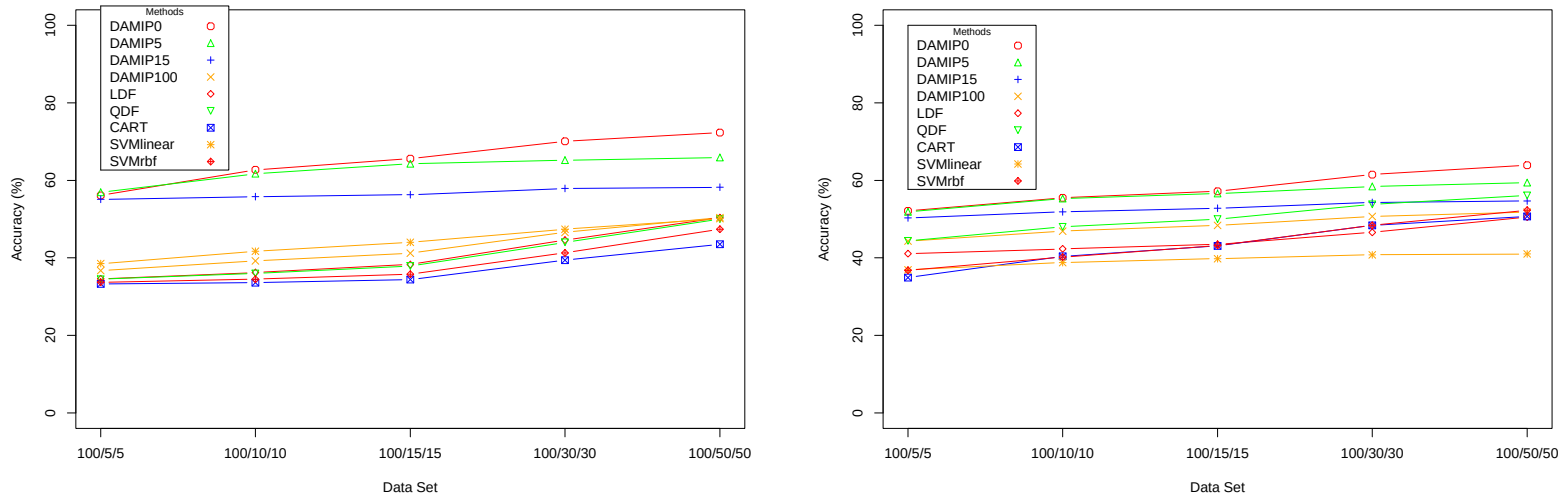
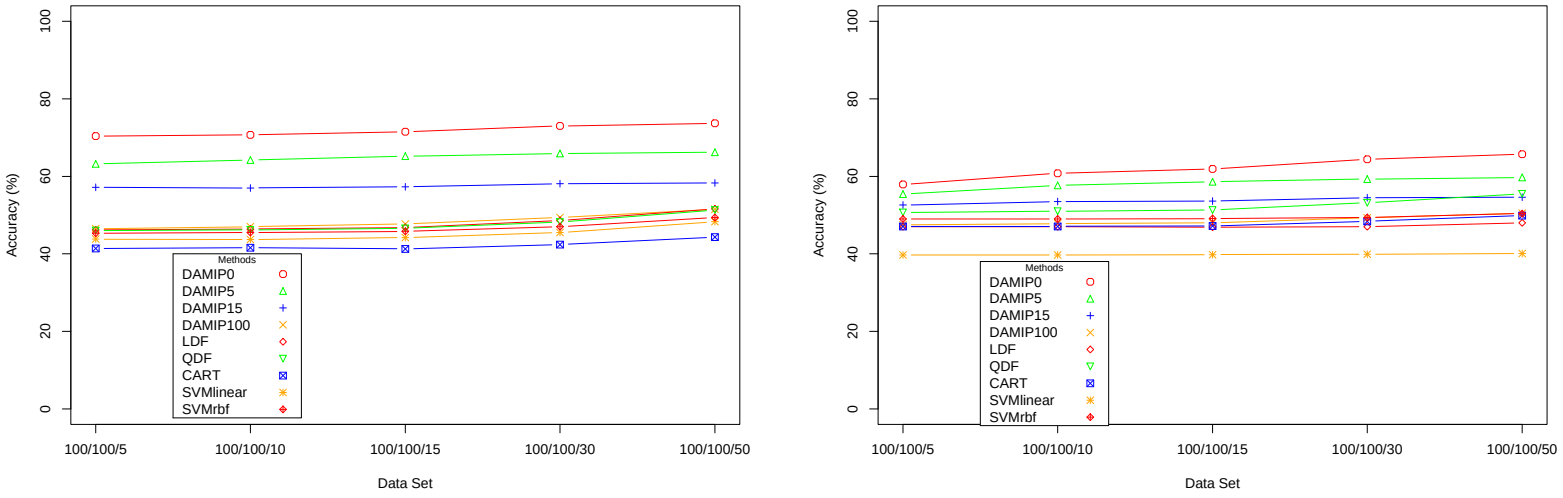
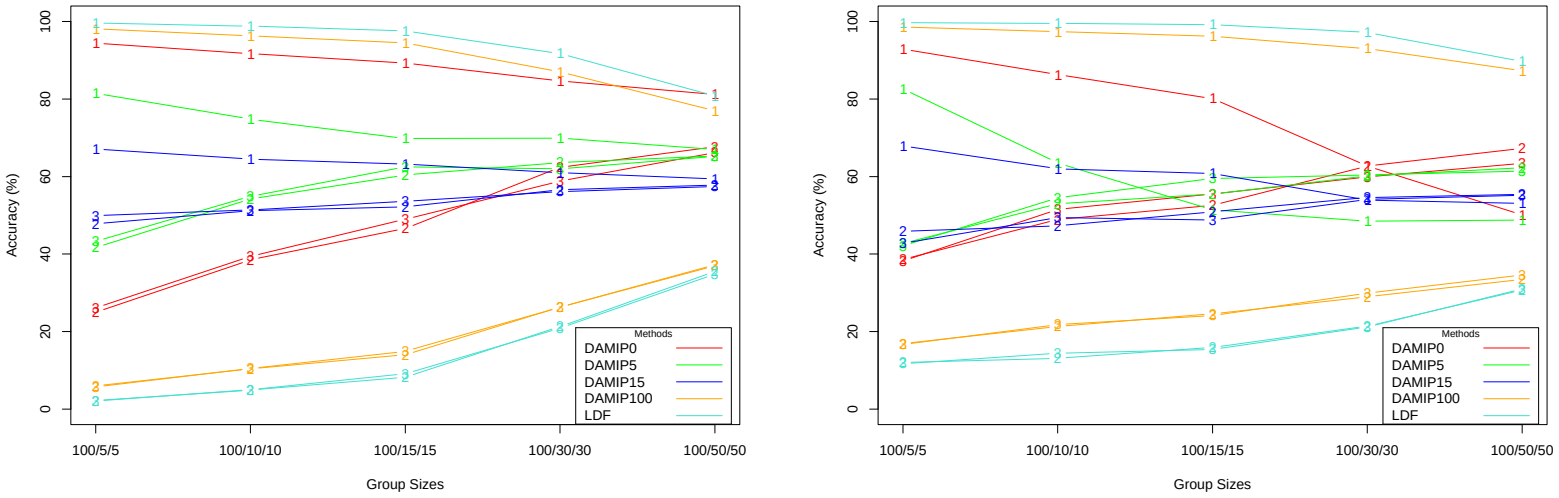


Figure 16: The accuracy of various classification methods on data generated from distributions (a) $E1$ and (b) $U1$ as described in Table 7. The values on the x-axis indicate the numbers of training observations for each of the three groups used in generating classification rules. The accuracy of the methods is the average performance on test observations. The DAMIP is tested with misclassification limits of 0 (DAMIP0), 5 (DAMIP5), 15 (DAMIP15), and 100% (DAMIP100). Other methods tested are linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), and support vector machines with a radial basis function kernel (SVMrbf). Accuracy is defined as the percentage of non-reserved observations that are correctly classified.



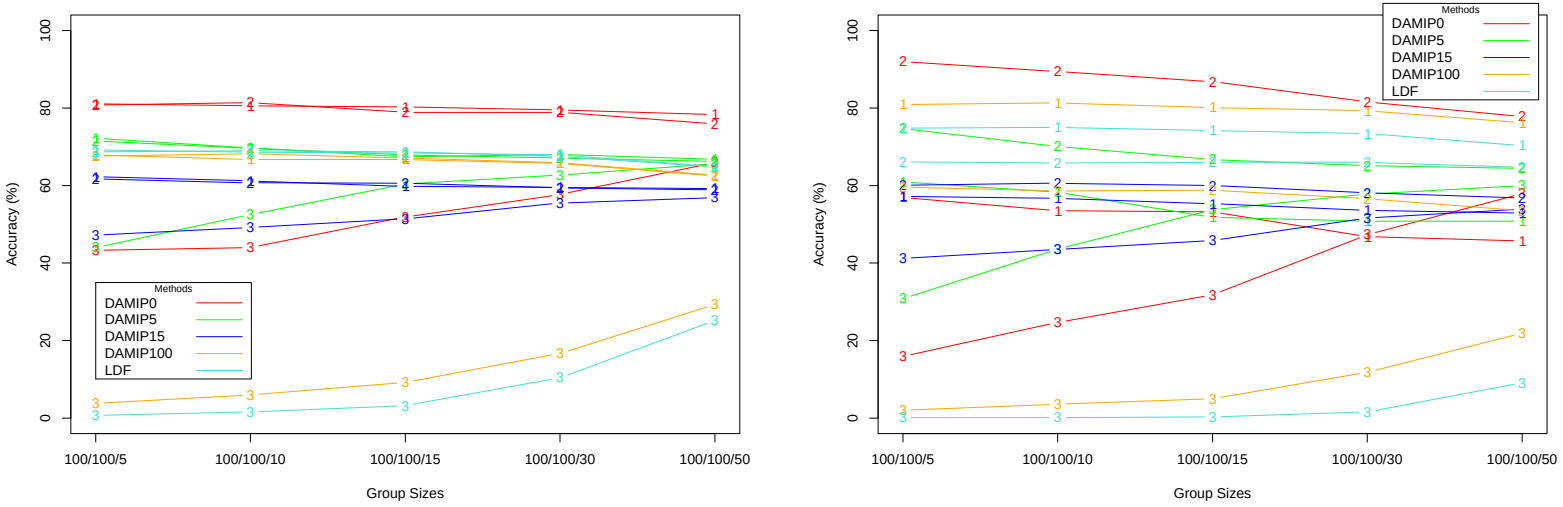
(a) (b)

Figure 17: The accuracy of various classification methods on data generated from distributions (a) $E1$ and (b) $U1$ as described in Table 7. The values on the x-axis indicate the numbers of training observations for each of the three groups used in generating classification rules. The accuracy of the methods is the average performance on test observations. The DAMIP is tested with misclassification limits of 0 (DAMIP0), 5 (DAMIP5), 15 (DAMIP15), and 100% (DAMIP100). Other methods tested are linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), and support vector machines with a radial basis function kernel (SVMrbf). Accuracy is defined as the percentage of non-reserved observations that are correctly classified.



(a) (b)

Figure 18: The classification accuracy of the DAMIP with misclassification limits of 0, 5, 15, and 100% and linear discriminant functions (LDF). The data are generated from bivariate normal distributions (a) $E1$ and (b) $U1$ as described in Table 7. The values on the x-axis indicate the numbers of training observations for each of the three groups. Accuracy is defined as the percentage of non-reserved observations that are correctly classified.



(a) (b)

Figure 19: The classification accuracy of the DAMIP with misclassification limits of 0, 5, 15, and 100% and linear discriminant functions (LDF). The data are generated from bivariate normal distributions (a) $E1$ and (b) $U1$ as described in Table 7. The values on the x-axis indicate the numbers of training observations for each of the three groups. Accuracy is defined as the percentage of non-reserved observations that are correctly classified.

chapter 6

Chapter VII

Conclusions, Contributions, and Future Work

The aim of this work is to take the theoretical results of Anderson [1] on classification to k populations with misclassification limits and develop a practical statistical method for implementing the results in light of previous work by Gallagher et al. [35]. The DAMIP, as introduced in [35], is the first computational framework for implementing Anderson's results. The mixed-integer programming models of the DAMIP can be computationally intensive, and accordingly much of this dissertation is dedicated to developing efficient solution methods. The classification performance of the DAMIP is ascertained, suggesting circumstances under which the model can be helpful.

The DAMIP is shown to be strongly universally consistent (in some sense) with very good rates of convergence from VC Theory, given that the density functions for the data are completely known. In general, it is not possible to guarantee any rate of convergence of a classifier [26]. When the assumption that the density functions are known is dropped, the rates of convergence are no longer valid because they will now depend on the method used to estimate the density functions. The effect of various consistent methods of density function estimation on the consistency of the DAMIP remains an interesting and open question.

The computational tests of the DAMIP on real-world and simulated data validate the theoretical consistency. Moreover, the DAMIP maintains competitive accuracy rates as the sample size increases. The accuracy does not increase significantly with sample size, indicating that only small sets of training observations are sufficient to train the model.

A polynomial-time algorithm for discriminating between two populations with the DAMIP is given. A mixed-integer programming formulation of the DAMIP is shown to be $\mathcal{NP}$-complete for a general number of groups. The proof demonstrating that the DAMIP is $\mathcal{NP}$-complete employs results used in generating edges of the conflict graph. The necessary and sufficient conditions proven for the existence of edges in the conflict graph is the

central contribution to the improvement in solution performance over industry-standard software. The conflict graph is the basis for various valid inequalities, a branching scheme, and for conditions under which integer variables are fixed for all solutions. Additional solution methods include a heuristic for finding solutions at nodes in the branch-and-bound tree, upper bounds for model parameters, and necessary conditions for edges in the conflict hypergraph.

The sizes of the instances tested with the DAMIP are small enough such that the conflict graph is easily generated, stored, and searched. These test instances are based on reasonably-sized samples compared to those commonly encountered statistical classification. While the conflict graph helps to improve solution performance, maximal hypercliques from the conflict 3-hypergraph does not seem to provide additional benefits. Other classes of cuts from the conflict 3-hypergraph can be tested in future work. One particular obstacle in exploiting the conflict 3-hypergraph is developing separation routines that do not require excessive amounts of computational power. Even for these relatively small mixed-integer programs, the conflict 3-hypergraph is too large to store before branch-and-bound and then search at nodes in the branch-and-bound tree.

The DAMIP with various misclassification limits is compared to traditional classification methods on real-world data and simulated data. It should be noted that the classification of real-world data performed in this work is done in a "blind" manner, without considering characteristics pertaining to individual data sets. The computational tests are not meant to provide insight into the data at hand, but rather to provide a testing-ground for various classification methods.

The DAMIP with small misclassification limits (i.e., 0 or 5% of training observations are allowed to be misclassified) provides increased accuracy, but large numbers of training and test observations are forced into the reserved judgment region and are therefore not given any classsification. The DAMIP with 15% misclassification limits shows evidence of being a viable classification method in a variety of situations. The accuracy and correct classification rates are competitive with traditional methods such as linear discriminant functions and support vector machines, while the reservation rates remain low under a

variety of circumstances. The particular circumstances tested here include real-world data, data generated from normal and contaminated normal distributions, and data with varying numbers of training observations from each group. The DAMIP with misclassification limits proves particularly useful when there are disproportionate numbers of training observations from different groups. Additionally, the simulations conducted here indicate that relatively small numbers of training observations are necessary for generating good classification rules. This characteristic can be useful when applying the DAMIP to large data sets for which the associated mixed-integer program is large and difficult to solve. In the future, a variety of simulations could be run to further test the viability of the DAMIP, including varying the methods for generating input data, classifying data from varying numbers of populations, and testing the classification accuracy of data generated from different distributions.

The DAMIP is presented as the second step of a two-step statistical process: (1) estimating prior probabilities and conditional group density function values for training data, and (2) solving a mixed-integer program to determine the optimal parameters defining the classification rules for test observations. The objective of the MIP is to maximize the number of correctly classified training observations, subject to limits on the number of misclassified observations. In light of the math program solved for support vector machines, a quadratic term could be added to the objective of the DAMIP to expand the margins between groups in the space of the conditional group density function values. Future work could consider solution methods for a model, and find coefficients that define the trade-off between maximizing correctly classified training observations and maximizing the margin between groups. These coefficients would enable an investigator to control the robustness of solutions to the DAMIP.

Though quadratic formulations of the DAMIP provide interesting problems, computational tests in this work indicate that, when input data are generated from a distribution with a density, the DAMIP provides stable classification rules. The stability is indicated by extremely low numbers of test observations that have ambiguous group membership when the classification rules of the DAMIP are applied. A final problem that remains open, and is actively studied, is that of estimating the density from which data is generated. Future

work could integrate cutting-edge methods for density estimation with the DAMIP.

In summary, this dissertation provides evidence that the DAMIP is a computationally feasible, consistent, stable, robust, and accurate classifier.

Appendix A

More Computational Test Results

This appendix contains additional data collected from the tests involving the evaluation of the performance of the enhanced code. The first section pertains to comparisons with CPLEX, and the second section is concerned with the relative contribution of the various components of the enhanced code.

The first two columns in each table define the data set and misclassification limit. The misclassification limit is the maximum percentage of observations from each group that can be misclassified as belonging to another group. The limit is applied to each pair of groups.

The third through sixth columns give the best-known objective value for an integer feasible solution, the best objective value for an integer feasible solution achieved under the current settings, the best upper bound achieved, and the percentage of integrality gap remaining. By definition, the percentage of integrality gap remaining is given by $(z_{UB}^{LP} - z^{IP})/(z_{root}^{LP} - z^{IP}) \times 100$, where the value z^{IP} is the optimal objective value of the mixed-integer program (column 4). If the value is unknown, the objective value associated with the best known integer feasible solution is used. The value z_{UB}^{LP} is the best objective value among active LP subproblems at termination (column 3). The value z_{root}^{LP} is the optimal objective value of the linear program relaxation at the root of the branch and bound tree.

For 3 of the problems, none of the methods are able to solve to optimality. For these problems, z^{IP} is defined as the best known objective value. The problems for which this definition is necessary and the z^{IP} used are *hun-0.15* 159, *cle-0.05* 93, and *cle-0.15* 133. Columns 7 through 12 give the number of nodes explored, the number of nodes explored before an integer feasible solution with the best integer objective value is found, the time to the best integer feasible solution found, the number of clique cuts used, and the number of generalized upper bound cuts (GUBs) used. If the optimal solution is found for an instance, then column 8 contains the number of nodes to optimality and column 9 will contain the

time to optimality. The last column gives the total processor time in seconds.

The tables for the enhanced code include an additional column (the second to last column) containing the number of user-generated cuts added.

Note that Table 15 has columns 11 and 12 removed because clique cuts and GUBs are not generated.

A.1 Comparison of performance

A.1.1 CPLEX

Table 9: CPLEX with default settings. Solving 5-group problems with CPLEX using default settings.

Problem		Solutions				Branch and Cut					Total
Data	α_{ij}	Opt Obj	Best Int Obj	Best UB (z_{UB}^{LP})	Gap Rem.(%)	Nodes	Nodes to Best	Time to Best (s)	Cli	GUB	Time (s)
va	0.05	27	27	27	0.0	1329	1278	3	0	0	3
va	0.15	50	49	54.9805	19.2	13158438	3293480	8808	0	0	32064
va	1.00	52	52	52	0.0	1549886	1549886	3857	0	0	3857
swi	0.05	30	N	59.9517	49.9	9493726	N	N	0	0	25634
swi	0.15	48	N	65.9767	42.8	5833840	N	N	0	0	1781
swi	1.00	55	55	55	0.0	8774642	77459	239	0	0	28814
hun	0.05	108	N	201.1208	74.5	2622268	N	N	0	0	14761
hun	0.15	159	N	213.9563	79.6	2275578	N	N	0	0	16025
hun	1.00	178	N	217.8248	65.4	2299520	N	N	0	0	16476
cle	0.05	(93)	N	224.9594	77.2	2071608	N	N	0	0	17672
cle	0.15	(133)	N	250.0411	89.4	2111610	N	N	0	0	16476
cle	1.00	188	178	247.8867	78.9	1960141	318027	3935	0	0	39217

Table 10: CPLEX with strong branching, cliques GUBs, and without presolve. Solving 5-group problems with strong branching, cliques and GUBs aggressively generated and used, and presolve turned off. All other cuts were turned off. The optimization strategy was set to emphasize optimality over feasibility, and the probing setting was at its most aggressive level. Alternative best estimate node selection procedure was used. Feasibility and optimality tolerance parameters were set at their strictest level.

Problem		Solutions				Branch and Cut					Total
Data	α_{ij}	Opt Obj	Best Int Obj	Best UB (z_{UB}^{LP})	Gap Rem.(%)	Nodes	Nodes to Best	Time to Best (s)	Cli	GUB	Time (s)
va	0.05	27	27	27	0.0	144	110	5	21	0	7
va	0.15	50	50	50	0.0	16583	4262	388	495	0	1406
va	1.00	52	52	52	0.0	11028	1490	139	493	0	776
swi	0.05	30	30	30	0.0	120799	33790	2363	192	0	8405
swi	0.15	48	47	56.9999	22.0	420018	163655	63013	2894	0	202217
swi	1.00	55	55	60.9914	17.1	206652	34538	5773	1769	0	29874
hun	0.05	108	108	178.7510	80.4	487560	73340	24896	2624	0	202765
hun	0.15	(159)	143	192.6465	61.2	158295	80731	87893	4404	0	201827
hun	1.00	178	178	195.9942	32.8	146641	42579	41850	7165	0	201827
cle	0.05	(93)	85	156.9474	54.2	306609	272243	177940	2585	0	202521
cle	0.15	(133)	N	221.9288	79.4	99181	N	N	4750	0	201427
cle	1.00	188	187	224.6099	48.2	124222	31003	39910	9625	0	201987

Table 11: CPLEX with strong branching, cliques GUBs, and presolve. Solving 5-group problems with strong branching, cliques and GUBs aggressively generated and used, and presolve turned on. All other cuts were turned off. The optimization strategy was set to emphasize optimality over feasibility, and the probing setting was at its most aggressive level. Alternative best estimate node selection procedure was used. Feasibility and optimality tolerance parameters were set at their strictest level.

Problem		Solutions				Branch and Cut					Total
Data	α_{ij}	Opt Obj	Best Int Obj	Best UB (z_{UB}^{LP})	Gap Rem.(%)	Nodes	Nodes to Best	Time to Best (s)	Cli	GUB	Time (s)
va	0.05	27	27	27	0.0	102	87	3	4	0	3
va	0.15	50	50	50	0.0	29926	10328	1111	35	0	2371
va	1.00	52	52	52	0.0	27591	15007	1337	21	0	2001
swi	0.05	30	30	30	0.0	260711	62560	10403	132	0	51972
swi	0.15	48	48	48	0.0	761218	572910	108776	150	0	130555
swi	1.00	55	55	55	0.0	581925	217696	25940	41	0	54277
hun	0.05	108	N	163.5994	45.7	292844	N	N	64	0	201758
hun	0.15	159	156	214.1800	78.0	381261	54955	28574	121	0	202705
hun	1.00	178	178	217.9425	77.1	1231622	1013165	171106	49	0	204774
cle	0.05	(93)	N	184.4233	55.6	162050	N	N	222	0	201503
cle	0.15	(133)	N	228.8472	76.6	208644	N	N	130	0	202154
cle	1.00	188	181	244.6365	80.7	314391	43175	28726	157	0	202669

A.2 The relative contribution of various components of the enhanced code

Table 12: Enhanced code with cuts added locally. Solving 5-group problems with up to 1 maximal clique, 1 odd hole, 1 maximal hyperclique, and 1 implied constraint added locally per pass. A branch callback function was used for branching. The conflict 3-hypergraph was generated on-the-fly. Variables were fixed to zero before optimization when appropriate. The heuristic was applied. The solution strategy was set to emphasize optimality over feasibility, probing was set to the highest level, strong branching was set as the default branching scheme, alternative best estimate node selection procedure was used, and cliques and GUBS were used.

Problem		Solutions				Branch and Cut						Total
Data	α_{ij}	Opt Obj	Best Int Obj	Best UB (z_{UB}^{LP})	Gap Rem.(%)	Nodes	Nodes to Best	Time to Best (s)	Cli	GUB	User Cuts	Time (s)
va	0.00	24	24	24	0.0	52	10	1	14	0	22	2
va	0.05	27	27	27	0.0	63	3	1	13	0	26	3
*va	0.15	50	50	50	0.0	8454	5823	672	137	0	28	920
va	1.00	52	52	52	0.0	2300	1278	69	25	0	31	107
swi	0.00	14	14	14	0.0	329	20	2	47	0	26	9
swi	0.05	30	30	30	0.0	20268	4483	664	64	0	96	22162
swi	0.15	48	48	48	0.0	107688	92110	19510	740	0	77	95400
swi	1.00	55	55	55	0.0	55329	27332	2201	80	0	76	4117
hun	0.00	33	33	33	0.0	137	72	9	198	0	8	12
hun	0.05	108	108	108	0.0	124266	48870	32719	1017	0	264	85852
hun	0.15	159	158	179.6644	27.9	332069	65445	44979	605	0	152	204579
hun	1.00	178	178	178	0.0	70840	9445	3556	377	0	64	24067
cle	0.00	70	70	70	0.0	1378	549	74	351	0	31	131
cle	0.05	(93)	91	151.5302	34.2	175120	112550	125297	892	0	379	205074
cle	0.15	(133)	91	191.1923	44.4	143997	0	0	1447	0	490	206498
cle	1.00	188	188	206.9962	25.0	435598	2643	1763	764	0	148	205487

Table 13: Enhanced code with cuts added globally. Solving 5-group Model 2 problems with up to 1 maximal clique, 1 odd hole, 1 maximal hyperclique, and 1 implied constraint added globally (rather than locally) per pass. A branch callback function was used for branching. The conflict 3-hypergraph was generated on-the-fly. Variables were fixed to zero before optimization when appropriate. The heuristic was applied. The solution strategy was set to emphasize optimality over feasibility, probing was set to the highest level, strong branching was set as the default branching scheme, alternative best estimate node selection procedure was used, and cliques and GUBS were used.

Problem		Solutions				Branch and Cut						Total
Data	α_{ij}	Opt Obj	Best Int Obj	Best UB (z_{UB}^{LP})	Gap Rem.(%)	Nodes	Nodes to Best	Time to Best (s)	Cli	GUB	User Cuts	Time (s)
va	0.00	24	24	24	0.0	52	10	1	14	0	23	1
va	0.05	27	27	27	0.0	63	3	1	13	0	35	3
*va	0.15	50	50	50	0.0	8385	7212	1373	72	0	1155	1529
va	1.00	52	52	52	0.0	2254	1313	76	25	0	218	115
swi	0.00	14	14	14	0.0	312	20	2	47	0	40	9
swi	0.05	30	30	30	0.0	17008	607	89	43	0	517	2462
swi	0.15	48	48	48	0.0	102438	98041	92347	146	0	6847	95400
swi	1.00	55	55	55	0.0	43919	15990	1812	21	0	1846	4844
hun	0.00	33	33	33	0.0	137	72	8	198	0	21	12
hun	0.05	108	108	108	0.0	80243	32798	38876	355	0	3360	113503
hun	0.15	159	157	179.6644	37.6	217947	15718	148065	245	0	4497	201775
*hun	1.00	178	178	178	0.0	128008	55831	20269	130	0	4375	78409
cle	0.00	70	70	70	0.0	1471	748	114	330	0	50	161
cle	0.05	(93)	91	151.6310	49.7	101914	24533	39489	301	0	3865	201657
cle	0.15	(133)	129	191.1923	52.0	41090	31377	93997	275	0	9709	201143
cle	1.00	188	188	206.9952	25.0	251448	62213	52395	351	0	5042	203140

Table 14: Enhanced code with non-dominated hyperclique cuts. Solving 5-group Model 2 problems with up to 1 maximal clique, 1 odd hole, 1 non-dominated maximal hyperclique, and 1 implied constraint added locally per pass. A branch callback function was used for branching. The conflict 3-hypergraph was generated on-the-fly. Variables were fixed to zero before optimization when appropriate. The heuristic was applied. The solution strategy was set to emphasize optimality over feasibility, probing was set to the highest level, strong branching was set as the default branching scheme, alternative best estimate node selection procedure was used, and cliques and GUBS were used.

Problem		Solutions				Branch and Cut						Total
Data	α_{ij}	Opt Obj	Best Int Obj	Best UB (z_{UB}^{LP})	Gap Rem.(%)	Nodes	Nodes to Best	Time to Best (s)	Cli	GUB	User Cuts	Time (s)
va	0.00	24	24	24	0.0	52	10	1	14	0	22	1
va	0.05	27	27	27	0.0	63	3	1	13	0	26	2
*va	0.15	50	50	50	0.0	8454	5823	559	137	0	28	763
va	1.00	52	52	52	0.0	2300	1278	54	25	0	31	84
swi	0.00	14	14	14	0.0	329	20	2	47	0	26	9
swi	0.05	30	30	30	0.0	20268	4483	596	64	0	96	2370
swi	0.15	48	48	48	0.0	107688	92110	20725	740	0	77	23931
swi	1.00	55	55	55	0.0	55329	27332	2080	80	0	76	3860
hun	0.00	33	33	33	0.0	137	72	8	198	0	8	12
hun	0.05	108	108	108	0.0	124266	48870	32047	1017	0	264	82530
hun	0.15	159	159	179.6644	37.6	337068	334422	202783	609	0	152	205532
hun	1.00	178	178	178	0.0	73712	9445	3331	488	0	64	24238
cle	0.00	70	70	70	0.0	1378	549	68	351	0	31	141
cle	0.05	(93)	91	151.0238	49.2	183787	112550	118540	899	0	379	204988
cle	0.15	(133)	91	191.1923	52.0	144658	0	0	1449	0	490	206444
cle	1.00	188	188	204.9965	22.4	441553	2643	1912	764	0	148	205414

Table 15: Enhanced code without maximal clique cuts. Solving 5-group Model 2 problems with up to 1 odd hole, 1 maximal hyperclique, and 1 implied constraint added locally per pass. A branch callback function was used for branching. The conflict 3-hypergraph was generated on-the-fly. Variables were fixed to zero before optimization when appropriate. The heuristic was applied. The solution strategy was set to emphasize optimality over feasibility, probing was set to the highest level, strong branching was set as the default branching scheme, alternative best estimate node selection procedure was used, and cliques and GUBS generated by CPLEX were used. Maximal cliques from the conflict graph were not used.

Problem		Solutions				Branch and Cut						Total
Data	α_{ij}	Opt Obj	Best Int Obj	Best UB (z_{UB}^{LP})	Gap Rem.(%)	Nodes	Nodes to Best	Time to Best (s)	Cli	GUB	User Cuts	Time (s)
va	0.00	24	24	24	0.0	93	50	2	45	0	6	3
va	0.05	27	27	27	0.0	53	17	1	19	0	9	2
*va	0.15	50	50	50	0.0	10786	9149	1269	364	0	15	1418
va	1.00	52	52	52	0.0	4515	758	39	79	0	26	188
swi	0.00	14	14	14	0.0	248	15	6	94	0	1	5
swi	0.05	30	30	30	0.0	27336	2348	361	117	0	12	3725
swi	0.15	48	48	48	0.0	129986	95729	29100	2140	0	61	41065
swi	1.00	55	55	55	0.0	59679	22451	3163	517	0	54	6826
hun	0.00	33	33	33	0.0	104	28	3	198	0	0	6
hun	0.05	108	108	130.0000	25.0	271263	230186	174149	1511	0	28	203190
hun	0.15	159	124	186.9932	50.9	251562	166907	148255	2437	0	43	202762
hun	1.00	178	178	178	0.0	96936	7973	4099	2400	0	56	38425
cle	0.00	70	70	70	0.0	1486	619	70	494	0	2	150
cle	0.05	(93)	90	135.6096	36.1	199505	117113	109698	1727	0	30	202150
cle	0.15	(133)	132	195.9844	56.3	129588	74695	109641	6225	0	168	203696
cle	1.00	188	188	207.9809	26.3	259931	29359	22750	6601	0	111	204641

Table 16: Enhanced code without hyperclique cuts. Solving 5-group Model 2 problems with up to 1 maximal clique, 1 odd hole, and 1 implied constraint added locally per pass. A branch callback function was used for branching. The conflict 3-hypergraph was generated on-the-fly. Variables were fixed to zero before optimization when appropriate. The heuristic was applied. The solution strategy was set to emphasize optimality over feasibility, probing was set to the highest level, strong branching was set as the default branching scheme, alternative best estimate node selection procedure was used, and cliques and GUBS were used.

Problem		Solutions				Branch and Cut						Total
Data	α_{ij}	Opt Obj	Best Int Obj	Best UB (z_{UB}^{LP})	Gap Rem.(%)	Nodes	Nodes to Best	Time to Best (s)	Cli	GUB	User Cuts	Time (s)
va	0.00	24	24	24	0.0	52	10	1	14	0	22	1
va	0.05	27	27	27	0.0	63	3	1	13	0	26	2
*va	0.15	50	50	50	0.0	8454		559	137	0	28	765
va	1.00	52	52	52	0.0	2300	1278	54	25	0	31	84
swi	0.00	14	14	14	0.0	329	20	2	47	0	26	9
swi	0.05	30	30	30	0.0	20268	4483	600	64	0	96	2370
swi	0.15	48	48	48	0.0	107688	92110	20693	740	0	77	23893
swi	1.00	55	55	55	0.0	55329	27332	2085	80	0	76	3848
hun	0.00	33	33	33	0.0	137	72	8	198	0	8	12
hun	0.05	108	108	108	0.0	124266	48870	31945	1017	0	264	82438
hun	0.15	159	159	179.6644	37.6	337068	334422	202930	609	0	152	204965
hun	1.00	178	178	178	0.0	73712	9445	3338	488	0	64	24255
cle	0.00	70	70	70	0.0	1378	549	87	351	0	31	141
cle	0.05	(93)	91	151.0238	49.2	184017	112550	118389	899	0	379	204849
cle	0.15	(133)	91	191.1923	52.0	144690	0	0	1449	0	490	206380
cle	1.00	188	188	204.9965	22.4	441815	2643	1904	764	0	148	205451

Table 17: Enhanced code without maximal clique cuts. Solving 5-group Model 2 problems with up to 1 maximal clique, 1 maximal hyperclique, and 1 implied constraint added locally per pass. A branch callback function was used for branching. The conflict 3-hypergraph was generated on-the-fly. Variables were fixed to zero before optimization when appropriate. The heuristic was applied. The solution strategy was set to emphasize optimality over feasibility, probing was set to the highest level, strong branching was set as the default branching scheme, alternative best estimate node selection procedure was used, and cliques and GUBS generated by CPLEX were used. Odd hole cuts from the conflict graph were not used.

Problem		Solutions				Branch and Cut						Total
Data	α_{ij}	Opt Obj	Best Int Obj	Best UB (z_{UB}^{LP})	Gap Rem.(%)	Nodes	Nodes to Best	Time to Best (s)	Cli	GUB	User Cuts	Time (s)
va	0.00	24	24	24	0.0	46	10	1	18	0	20	1
va	0.05	27	27	27	0.0	46	7	1	11	0	23	2
*va	0.15	50	49	49	0.0	14649	8118	830	156	0	21	1421
va	1.00	52	52	52	0.0	2747	1379	71	25	0	29	125
swi	0.00	14	14	14	0.0	388	283	6	50	0	24	7
swi	0.05	30	30	30	0.0	22289	4356	594	60	0	71	2559
swi	0.15	48	48	48	0.0	87702	17596	2939	728	0	71	13620
swi	1.00	55	55	55	0.0	66835	43573	3279	128	0	71	4515
hun	0.00	33	33	33	0.0	137	72	5	198	0	8	7
hun	0.05	108	108	108	0.0	215393	15851	122424	1095	0	168	108976
hun	0.15	159	157	180.6637	39.4	409165	126201	70143	851	0	131	203650
hun	1.00	178	178	178	0.0	69089	8941	3689	581	0	72	23126
cle	0.00	70	70	70	0.0	1270	827	109	492	0	29	153
cle	0.05	(93)	89	151.3746	49.5	217733	8210	4202	1070	0	299	205667
cle	0.15	(133)	130	191.2421	52.0	112077	80916	159291	815	0	864	205398
cle	1.00	188	188	206.9957	25.0	510374	271260	98648	320	0	151	206062

Table 18: Enhanced code without variable fixing or implied cuts. Solving 5-group Model 2 problems with up to 1 maximal clique, 1 odd hole, and 1 maximal hypercliuqe constraint added locally per pass. A branch callback function was used for branching. The conflict 3-hypergraph was generated on-the-fly. The heuristic was applied. The solution strategy was set to emphasize optimality over feasibility, probing was set to the highest level, strong branching was set as the default branching scheme, alternative best estimate node selection procedure was used, and cliques and GUBS were used.

Problem		Solutions				Branch and Cut						Total
Data	α_{ij}	Opt Obj	Best Int Obj	Best UB (z_{UB}^{LP})	Gap Rem.(%)	Nodes	Nodes to Best	Time to Best (s)	Cli	GUB	User Cuts	Time (s)
va	0.00	24	24	24	0.0	61	14	1	8	0	23	2
va	0.05	27	27	27	0.0	58	9	1	11	0	42	2
*va	0.15	50	50	50	0.0	7148	6969	744	152	0	21	753
va	1.00	52	52	52	0.0	3503	190	21	31	0	19	223
swi	0.00	14	14	14	0.0	401	102	4	26	0	37	12
swi	0.05	30	30	30	0.0	28710	3982	875	78	0	123	5455
swi	0.15	48	48	48	0.0	115445	115416	37191	1062	0	68	37198
swi	1.00	55	55	55	0.0	68234	11907	1505	157	0	62	7120
hun	0.00	33	33	33	0.0	464	311	25	134	0	11	34
hun	0.05	108	108	108	0.0	131564	55698	49586	1453	0	518	139705
hun	0.15	159	159	180.4989	39.1	271284	11968	7843	891	0	154	205581
hun	1.00	178	178	178	0.0	80149	21854	8574	661	0	67	29851
cle	0.00	70	70	70	0.0	1487	143	44	283	0	42	184
cle	0.05	(93)	90	156.2028	53.6	126814	89768	135936	973	0	602	205329
cle	0.15	(133)	133	196.0350	56.3	102500	47968	77070	1856	0	304	207517
cle	1.00	188	185	208.9959	27.7	428064	369635	173085	1903	0	176	205821

Table 19: Enhanced code without branching scheme. Solving 5-group Model 2 problems with up to 1 maximal clique, 1 odd hole, 1 maximal hyperclique, and 1 implied constraint added locally per pass. The conflict 3-hypergraph was generated on-the-fly. Variables were fixed to zero before optimization when appropriate. The heuristic was applied. The solution strategy was set to emphasize optimality over feasibility, probing was set at the highest level, strong branching was set as the default branching scheme, alternative best estimate node selection procedure was used, and cliques and GUBS were used.

Problem		Solutions				Branch and Cut						Total
Data	α_{ij}	Opt Obj	Best Int Obj	Best UB (z_{UB}^{LP})	Gap Rem.(%)	Nodes	Nodes to Best	Time to Best (s)	Cli	GUB	User Cuts	Time (s)
va	0.00	24	24	24	0.0	95	84	4	14	0	22	4
va	0.05	27	27	27	0.0	56	5	1	13	0	26	3
va	0.15	50	50	50	0.0	12683	10124	1510	81	0	23	1785
va	1.00	52	52	52	0.0	3146	319	30	30	0	31	198
swi	0.00	14	14	14	0.0	242	55	2	63	0	26	7
swi	0.05	30	30	30	0.0	18437	787	119	72	0	96	3078
swi	0.15	48	48	48	0.0	97620	17118	4809	1025	0	77	31590
swi	1.00	55	55	55	0.0	26279	7795	1516	59	0	76	4220
hun	0.00	33	33	33	0.0	154	23	10	198	0	8	45
hun	0.05	108	108	117.9955	11.4	260207	57960	69608	1137	0	264	205445
hun	0.15	159	159	173.9934	27.3	192065	69653	69663	1183	0	152	204418
hun	1.00	178	178	178	0.0	64077	49629	24661	493	0	64	32750
cle	0.00	70	70	70	0.0	1050	256	38	355	0	31	188
cle	0.05	(93)	91	149.4615	47.9	284949	182680	128752	958	0	379	205449
cle	0.15	(133)	91	190.5666	51.4	0	0	0	2976	0	490	205779
cle	1.00	188	188	200.9963	17.1	141535	42899	36204	5366	0	148	203831

Appendix B

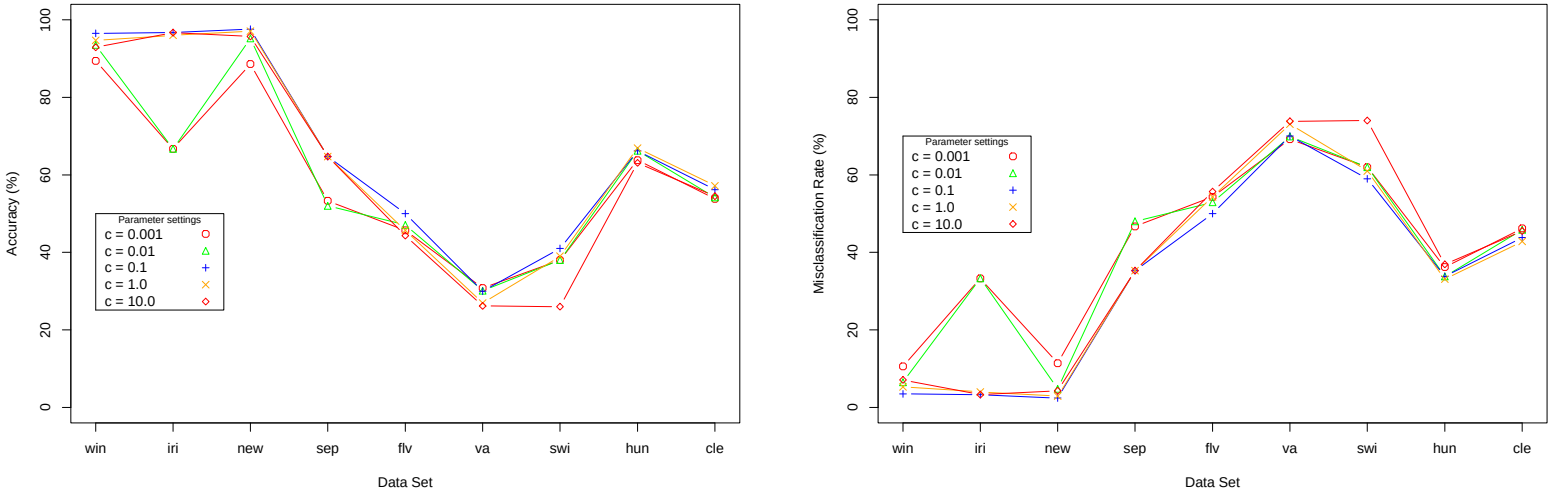
More Classification Accuracy Test Results

This appendix contains additional results for various tests of classification performance. The first section contains results of tests used to determine the optimal parameters for SVM^{multiclass} [47], a multi-class support vector machine code. The second section contains classification matrices for tests of classification performance for various methods on real-world data sets. The third section contains classification matrices for tests of classification performance for various methods on simulated data sets.

B.1 Parameter settings in SVM^{multiclass} [47]

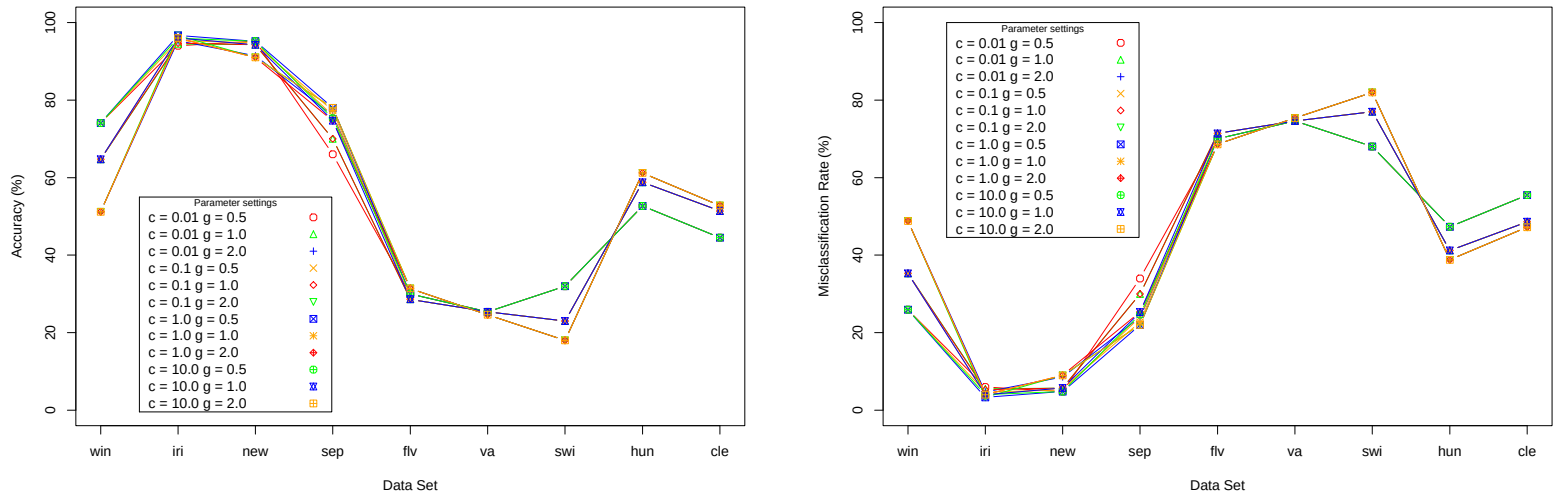
Figures 20 and 21 contain the accuracy and misclassification rates of SVM^{multiclass} [47] using a linear and radial basis function kernel, respectively. For each kernel, the c parameter was tested for various values. The parameter determines the relative emphasis in the objective on minimizing training error and maximizing the margin between groups.

For SVM^{multiclass} [47] with a radial basis function kernel, values for g were tested. The g parameter determines the width of the radial basis function kernel.



(a) (b)

Figure 20: (a) Accuracy and (b) misclassification rates for various settings for SVM^{multiclass} [47] with a linear kernel on real-world data sets. The parameter c determines the relative emphasis of the objective function on reducing training error and increasing the margin between groups. The data sets are described in Section 6.1.



(a) (b)

Figure 21: (a) Accuracy and (b) misclassification rates for various settings for SVM^{multiclass} [47] with a radial basis function kernel on real-world data sets. The parameter c determines the relative emphasis of the objective function on reducing training error and increasing the margin between groups. The parameter g determines the width of the radial basis function kernel. The data sets are described in Section 6.1.

B.2 Real-world data

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
0.0	98.3	1.7	0.0	0.0	98.3	1.7	0.0	0.0	98.3	1.7	0.0	0.0	98.3	1.7	0.0
0.0	1.5	97.1	1.5	0.0	1.5	97.1	1.5	0.0	1.5	97.1	1.5	0.0	1.5	97.1	1.5
0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0

LDF				QDF				CART				SVMlinear				SVMrbf			
0.0	98.3	1.7	0.0	0.0	100.0	0.0	0.0	0.0	89.7	8.6	1.7	0.0	94.8	5.2	0.0	0.0	96.6	1.7	1.7
0.0	1.5	97.1	1.5	0.0	1.5	98.5	0.0	0.0	2.9	94.1	2.9	0.0	0.0	97.1	2.9	0.0	33.8	54.4	11.8
0.0	0.0	0.0	100.0	0.0	0.0	2.3	97.7	0.0	0.0	9.1	90.9	0.0	0.0	2.3	97.7	0.0	43.2	18.2	38.6

(a) *wine*

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0
2.0	0.0	96.0	2.0	0.0	0.0	96.0	4.0	0.0	0.0	96.0	4.0	0.0	0.0	96.0	4.0
8.0	0.0	2.0	90.0	0.0	0.0	6.0	94.0	0.0	0.0	4.0	96.0	0.0	0.0	4.0	96.0

LDF				QDF				CART				SVMlinear				SVMrbf			
0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	100.0	0.0	0.0
0.0	0.0	96.0	4.0	0.0	0.0	94.0	6.0	0.0	0.0	94.0	6.0	0.0	0.0	94.0	6.0	0.0	0.0	92.7	7.3
0.0	0.0	2.0	98.0	0.0	0.0	2.0	98.0	0.0	0.0	6.0	94.0	0.0	0.0	4.0	96.0	0.0	0.0	9.6	90.4

(b) *iris*

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
6.1	93.2	0.7	0.0	0.7	95.3	2.0	2.0	0.0	96.6	2.0	1.4	0.0	96.6	2.0	1.4
12.1	3.0	84.8	0.0	0.0	6.1	93.9	0.0	0.0	9.1	90.9	0.0	0.0	9.1	90.9	0.0
10.3	3.4	0.0	86.2	3.4	3.4	0.0	93.1	0.0	13.8	0.0	86.2	0.0	17.2	0.0	82.8

LDF				QDF				CART				SVMlinear				SVMrbf			
0.0	100.0	0.0	0.0	0.0	98.0	1.4	0.7	0.0	100.0	0.0	0.0	0.0	98.6	0.7	0.7	0.0	98.0	0.0	2.0
0.0	33.3	66.7	0.0	0.0	3.0	97.0	0.0	0.0	15.2	84.8	0.0	0.0	0.0	100.0	0.0	0.0	3.0	97.0	0.0
0.0	24.1	0.0	75.9	0.0	13.8	0.0	86.2	0.0	24.1	0.0	75.9	0.0	10.3	0.0	89.7	0.0	27.6	0.0	72.4

(c) *new-thyroid*

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
2.0	98.0	0.0	0.0	0.0	98.0	2.0	0.0	0.0	98.0	2.0	0.0	0.0	98.0	2.0	0.0
70.0	2.0	24.0	4.0	46.0	4.0	40.0	10.0	36.0	2.0	48.0	14.0	2.0	2.0	64.0	32.0
74.0	0.0	2.0	24.0	62.0	0.0	6.0	32.0	38.0	0.0	18.0	44.0	0.0	0.0	28.0	72.0

LDF				QDF				CART				SVMlinear				SVMrbf			
0.0	98.0	2.0	0.0	0.0	98.0	2.0	0.0	2.0	92.0	6.0	0.0	0.0	98.0	0.0	2.0	0.0	100.0	0.0	0.0
0.0	0.0	70.0	30.0	0.0	0.0	70.0	30.0	2.0	4.0	60.0	34.0	0.0	12.0	0.0	88.0	0.0	22.0	36.0	42.0
0.0	0.0	30.0	70.0	0.0	0.0	36.0	64.0	0.0	6.0	38.0	56.0	0.0	4.0	0.0	96.0	0.0	4.0	22.0	74.0

(d) *sepal*

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
62.5	12.5	12.5	12.5	54.2	16.7	20.8	8.3	20.8	25.0	33.3	20.8	4.2	41.7	33.3	20.8
62.5	12.5	20.8	4.2	45.8	25.0	25.0	4.2	29.2	25.0	37.5	8.3	0.0	41.7	37.5	20.8
54.5	0.0	0.0	45.5	36.4	4.5	0.0	59.1	31.8	4.5	9.1	54.5	0.0	4.5	13.6	81.8

LDF				QDF				CART				SVMlinear				SVMrbf			
0.0	37.5	41.7	20.8	0.0	20.8	58.3	20.8	0.0	41.7	45.8	12.5	0.0	33.3	41.7	25.0	0.0	83.3	16.7	0.0
0.0	50.0	41.7	8.3	0.0	25.0	58.3	16.7	0.0	29.2	58.3	12.5	0.0	41.7	54.2	4.2	0.0	87.5	0.0	12.5
0.0	13.6	18.2	68.2	0.0	18.2	13.6	68.2	0.0	0.0	13.6	86.4	0.0	4.5	31.8	63.6	0.0	86.4	13.6	0.0

(e) *FNlnVN*

Figure 22: Classification matrices (a) *wine* (b) *iris* (c) *new-thyroid* (d) *sepal* and (e) *FNlnVN* data sets for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j+1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

DAMIP 0%						DAMIP 5%						DAMIP 15%					
62.1	6.9	13.8	13.8	3.4	0.0	58.6	13.8	10.3	6.9	10.3	0.0	13.8	34.5	17.2	20.7	13.8	0.0
71.8	12.8	10.3	5.1	0.0	0.0	66.7	12.8	10.3	5.1	5.1	0.0	35.9	17.9	23.1	10.3	10.3	2.6
86.2	3.4	0.0	10.3	0.0	0.0	72.4	3.4	6.9	10.3	6.9	0.0	41.4	10.3	10.3	20.7	13.8	3.4
66.7	0.0	7.4	3.7	14.8	7.4	55.6	3.7	11.1	7.4	14.8	7.4	37.0	0.0	18.5	22.2	11.1	11.1
100.0	0.0	0.0	0.0	0.0	0.0	83.3	0.0	0.0	0.0	0.0	16.7	83.3	0.0	0.0	0.0	0.0	16.7
DAMIP 100%						LDF						QDF					
0.0	24.1	34.5	20.7	20.7	0.0	0.0	28.6	28.6	14.3	14.3	14.3	0.0	35.7	32.1	10.7	10.7	10.7
0.0	25.6	28.2	20.5	25.6	0.0	0.0	26.7	26.7	15.6	15.6	15.6	0.0	30.0	32.5	12.5	12.5	12.5
0.0	10.3	34.5	27.6	20.7	6.9	0.0	11.4	28.6	20.0	20.0	20.0	0.0	18.2	45.5	12.1	12.1	12.1
3.7	0.0	48.1	18.5	18.5	11.1	0.0	5.3	23.7	23.7	23.7	23.7	0.0	20.7	37.9	13.8	13.8	13.8
16.7	0.0	0.0	33.3	50.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	40.0	20.0	20.0	20.0
CART						SVMlinear						SVMrbf					
0.0	21.4	21.4	19.0	19.0	19.0	0.0	58.3	41.7	0.0	0.0	0.0	0.0	60.0	5.7	11.4	11.4	11.4
0.0	26.8	36.6	12.2	12.2	12.2	0.0	20.5	18.2	20.5	20.5	20.5	0.0	60.0	13.3	8.9	8.9	8.9
2.9	5.9	38.2	17.6	17.6	17.6	0.0	15.2	30.3	18.2	18.2	18.2	0.0	57.1	8.6	11.4	11.4	11.4
0.0	22.6	29.0	16.1	16.1	16.1	0.0	26.7	53.3	6.7	6.7	6.7	0.0	66.7	3.3	10.0	10.0	10.0
0.0	0.0	7.7	30.8	30.8	30.8	0.0	50.0	50.0	0.0	0.0	0.0	0.0	80.0	20.0	0.0	0.0	0.0

Figure 23: Classification matrices for *va* data set for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 5×6 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j + 1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

DAMIP 0%						DAMIP 5%						DAMIP 15%					
83.3	0.0	16.7	0.0	0.0	0.0	50.0	0.0	33.3	0.0	16.7	0.0	0.0	16.7	66.7	0.0	16.7	0.0
65.8	5.3	13.2	7.9	5.3	2.6	50.0	7.9	21.1	13.2	5.3	2.6	15.8	21.1	18.4	18.4	18.4	7.9
67.9	3.6	7.1	10.7	3.6	7.1	39.3	3.6	10.7	25.0	7.1	14.3	10.7	3.6	10.7	39.3	17.9	17.9
69.6	0.0	8.7	4.3	8.7	8.7	34.8	0.0	17.4	13.0	17.4	17.4	8.7	0.0	21.7	17.4	34.8	17.4
100.0	0.0	0.0	0.0	0.0	0.0	60.0	0.0	0.0	20.0	20.0	0.0	20.0	0.0	0.0	20.0	20.0	40.0
DAMIP 100%						LDF						QDF					
0.0	0.0	83.3	0.0	16.7	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	40.0	20.0	20.0	20.0
2.6	5.3	55.3	13.2	21.1	2.6	0.0	6.5	47.8	15.2	15.2	15.2	0.0	16.7	50.0	11.1	11.1	11.1
0.0	3.6	28.6	39.3	17.9	10.7	0.0	2.3	15.9	27.3	27.3	27.3	0.0	6.9	20.7	24.1	24.1	24.1
0.0	0.0	43.5	21.7	21.7	13.0	0.0	0.0	40.0	20.0	20.0	20.0	0.0	0.0	28.6	23.8	23.8	23.8
0.0	0.0	20.0	40.0	20.0	20.0	0.0	0.0	0.0	33.3	33.3	33.3	0.0	0.0	0.0	33.3	33.3	33.3
CART						SVMlinear						SVMrbf					
0.0	0.0	36.4	21.2	21.2	21.2	0.0	0.0	62.5	12.5	12.5	12.5	0.0	12.5	12.5	25.0	25.0	25.0
0.0	0.0	41.3	19.6	19.6	19.6	0.0	2.0	55.1	14.3	14.3	14.3	0.0	33.3	31.0	11.9	11.9	11.9
0.0	0.0	23.3	25.6	25.6	25.6	0.0	0.0	36.8	21.1	21.1	21.1	0.0	48.3	20.7	10.3	10.3	10.3
0.0	0.0	14.3	28.6	28.6	28.6	0.0	0.0	52.0	16.0	16.0	16.0	0.0	30.8	11.5	19.2	19.2	19.2
0.0	0.0	25.0	25.0	25.0	25.0	0.0	0.0	10.0	30.0	30.0	30.0	0.0	66.7	33.3	0.0	0.0	0.0

Figure 24: Classification matrices for *switzerland* data set for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 5×6 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j+1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

DAMIP 0%						DAMIP 5%						DAMIP 15%					
87.1	10.4	1.2	0.6	0.0	0.6	44.8	47.9	5.5	0.6	0.6	0.6	27.0	62.6	4.9	3.1	1.8	0.6
82.9	5.7	2.9	2.9	5.7	0.0	68.6	8.6	11.4	5.7	5.7	0.0	45.7	17.1	20.0	5.7	11.4	0.0
78.3	4.3	4.3	0.0	4.3	8.7	65.2	4.3	8.7	0.0	8.7	13.0	39.1	4.3	17.4	4.3	17.4	17.4
68.0	0.0	4.0	4.0	16.0	8.0	60.0	0.0	4.0	12.0	16.0	8.0	56.0	4.0	8.0	4.0	20.0	8.0
71.4	0.0	0.0	7.1	14.3	7.1	57.1	0.0	0.0	0.0	14.3	28.6	42.9	0.0	7.1	0.0	28.6	21.4
DAMIP 100%						LDF						QDF					
0.0	92.6	3.7	1.2	1.2	1.2	0.0	88.7	4.2	2.4	2.4	2.4	0.0	85.6	6.9	2.5	2.5	2.5
2.9	60.0	17.1	11.4	8.6	0.0	0.0	50.0	18.4	10.5	10.5	10.5	0.0	44.2	14.0	14.0	14.0	14.0
0.0	43.5	13.0	0.0	30.4	13.0	0.0	37.5	43.8	6.2	6.2	6.2	0.0	22.2	22.2	18.5	18.5	18.5
0.0	32.0	12.0	20.0	20.0	16.0	0.0	21.4	14.3	21.4	21.4	21.4	0.0	14.8	18.5	22.2	22.2	22.2
0.0	28.6	14.3	14.3	28.6	14.3	0.0	44.4	22.2	11.1	11.1	11.1	0.0	23.5	23.5	17.6	17.6	17.6
CART						SVMlinear						SVMrbf					
0.0	98.1	1.9	0.0	0.0	0.0	0.0	99.4	0.6	0.0	0.0	0.0	0.0	86.8	6.0	2.4	2.4	2.4
0.0	80.6	2.8	5.6	5.6	5.6	0.0	64.7	17.6	5.9	5.9	5.9	0.0	82.4	17.6	0.0	0.0	0.0
0.0	75.0	12.5	4.2	4.2	4.2	0.0	68.4	15.8	5.3	5.3	5.3	0.0	95.2	4.8	0.0	0.0	0.0
0.0	54.2	33.3	4.2	4.2	4.2	0.0	45.8	16.7	12.5	12.5	12.5	0.0	87.5	0.0	4.2	4.2	4.2
0.0	50.0	20.0	10.0	10.0	10.0	0.0	54.5	18.2	9.1	9.1	9.1	0.0	73.3	6.7	6.7	6.7	6.7

Figure 25: Classification matrices for *hungarian* data set for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 5×6 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j + 1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

DAMIP 0%

66.0	33.3	0.6	0.0	0.0	0.0
90.6	3.8	1.9	0.0	3.8	0.0
90.9	0.0	0.0	3.0	6.1	0.0
77.1	2.9	2.9	11.4	2.9	2.9
69.2	0.0	7.7	0.0	15.4	7.7

DAMIP 5%

51.9	42.3	5.1	0.6	0.0	0.0
88.7	5.7	0.0	3.8	1.9	0.0
81.8	0.0	6.1	3.0	6.1	3.0
60.0	2.9	2.9	20.0	5.7	8.6
61.5	0.0	7.7	7.7	23.1	0.0

DAMIP 15%

39.1	51.3	7.7	0.0	0.6	1.3
77.4	11.3	3.8	3.8	3.8	0.0
66.7	0.0	9.1	3.0	15.2	6.1
42.9	2.9	11.4	25.7	8.6	8.6
53.8	0.0	7.7	7.7	30.8	0.0

DAMIP 100%

0.6	91.0	4.5	2.6	0.6	0.6
0.0	49.1	17.0	20.8	9.4	3.8
0.0	18.2	15.2	27.3	21.2	18.2
0.0	5.7	25.7	34.3	14.3	20.0
7.7	7.7	15.4	15.4	46.2	7.7

LDF

0.0	89.8	6.4	1.3	1.3	1.3
0.0	47.1	23.5	9.8	9.8	9.8
0.0	22.7	22.7	18.2	18.2	18.2
0.0	5.6	27.8	22.2	22.2	22.2
0.0	14.3	42.9	14.3	14.3	14.3

QDF

0.0	84.9	7.5	2.5	2.5	2.5
0.0	36.1	19.7	14.8	14.8	14.8
0.0	21.9	12.5	21.9	21.9	21.9
0.0	9.1	36.4	18.2	18.2	18.2
0.0	9.1	36.4	18.2	18.2	18.2

CART

0.0	89.7	8.3	0.6	0.6	0.6
0.0	60.7	17.9	7.1	7.1	7.1
4.7	25.6	20.9	16.3	16.3	16.3
3.0	45.5	33.3	6.1	6.1	6.1
12.5	37.5	50.0	0.0	0.0	0.0

SVMlinear

0.0	96.8	1.3	0.6	0.6	0.6
0.0	72.5	3.9	7.8	7.8	7.8
0.0	52.6	15.8	10.5	10.5	10.5
0.0	21.1	15.8	21.1	21.1	21.1
0.0	30.0	10.0	20.0	20.0	20.0

SVMrbf

0.0	88.8	3.7	2.5	2.5	2.5
0.0	80.0	3.6	5.5	5.5	5.5
0.0	62.5	7.5	10.0	10.0	10.0
0.0	94.1	5.9	0.0	0.0	0.0
0.0	71.4	7.1	7.1	7.1	7.1

Figure 26: Classification matrices for *cleveland* data set for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 5×6 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j + 1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

B.3 Simulated data

B.3.1 Equal group sizes

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
41.2	29.9	14.3	14.6	39.3	30.9	14.6	15.2	39.3	30.9	14.6	15.2	1.2	46.1	26.1	26.6
40.6	13.8	30.6	14.9	39.0	14.2	31.5	15.4	39.0	14.2	31.5	15.4	1.3	25.4	46.9	26.5
40.0	14.2	14.2	31.6	38.3	14.6	14.6	32.5	38.3	14.6	14.6	32.5	1.2	25.6	25.5	47.7

LDF				QDF				CART				SVMlinear				SVMrbf			
0.2	48.1	25.8	25.9	0.1	43.9	27.6	28.4	20.1	37.8	23.8	18.4	0.0	30.6	34.8	34.6	0.0	38.3	29.6	32.1
0.2	25.5	48.2	26.1	0.1	28.9	42.9	28.0	21.7	23.8	37.3	17.2	0.0	14.2	54.1	31.7	0.0	24.6	44.3	31.2
0.2	26.0	24.8	48.9	0.1	29.0	27.2	43.7	24.6	25.0	22.7	27.7	0.0	14.6	31.3	54.2	0.0	23.5	29.0	47.5

(a) $E1$

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
43.0	27.3	15.3	14.3	41.5	27.6	15.7	15.2	41.5	27.6	15.7	15.2	1.4	45.6	26.4	26.6
41.8	13.6	28.8	15.7	40.3	13.9	29.4	16.5	40.3	13.9	29.4	16.5	1.5	25.8	43.4	29.3
42.8	13.7	15.8	27.8	41.0	14.0	16.0	29.0	41.0	14.0	16.0	29.1	1.5	26.0	27.0	45.5

LDF				QDF				CART				SVMlinear				SVMrbf			
0.2	48.7	25.9	25.2	0.1	54.0	23.6	22.3	14.8	46.9	21.6	16.6	0.0	35.7	32.4	31.9	0.0	46.9	25.5	27.5
0.2	26.9	45.1	27.8	0.1	23.2	45.6	31.1	17.3	25.9	35.5	21.2	0.0	20.5	44.6	35.0	0.0	21.2	44.3	34.5
0.2	27.4	27.5	44.8	0.1	23.9	30.1	45.8	19.7	26.3	24.2	29.8	0.0	20.7	35.4	43.9	0.0	21.5	30.9	47.7

(b) $U1$

Figure 27: Classification matrices for (a) $E1$ and (b) $U1$ distributions with 5 training observations in each group. Classification matrices for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j+1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

DAMIP 0%					DAMIP 5%				DAMIP 15%				DAMIP 100%			
	71.0	17.9	5.5	5.6	69.6	18.7	5.8	5.9	24.9	41.5	16.7	16.9	0.9	50.5	24.1	24.5
	70.5	5.4	18.8	5.3	69.3	5.6	19.5	5.6	25.0	17.1	41.7	16.3	0.9	24.3	51.0	23.8
	70.6	5.3	6.0	18.2	69.3	5.4	6.1	19.2	24.7	16.5	17.4	41.4	0.9	23.9	24.7	50.5

LDF				QDF				CART				SVMlinear				SVMrbf			
0.2	52.5	23.5	23.9	0.2	50.1	24.7	25.0	12.9	42.1	23.7	21.4	0.0	29.4	35.3	35.2	0.0	41.5	28.0	30.5
0.2	23.6	52.5	23.6	0.2	24.9	50.0	24.9	12.7	24.6	41.7	21.1	0.0	10.4	59.4	30.2	0.0	22.4	49.1	28.5
0.3	23.2	23.9	52.6	0.2	24.6	25.2	50.0	14.7	24.1	23.6	37.7	0.0	10.3	29.5	60.2	0.0	21.6	27.2	51.3

(a) $E1$

DAMIP 0%					DAMIP 5%				DAMIP 15%				DAMIP 100%			
	81.1	9.6	4.6	4.7	80.1	9.9	5.0	5.0	38.8	30.7	15.1	15.3	0.9	52.1	23.9	23.2
	74.6	4.1	14.6	6.7	73.7	4.5	15.1	6.7	31.7	15.0	35.3	18.0	0.8	27.5	46.0	25.8
	74.2	4.1	6.1	15.6	73.4	4.5	6.1	16.1	31.5	15.1	16.6	36.9	0.7	27.8	24.8	46.7

LDF				QDF				CART				SVMlinear				SVMrbf			
0.3	53.1	23.4	23.3	0.1	64.7	17.7	17.4	13.3	49.7	20.0	17.0	0.0	33.7	32.4	33.9	0.0	56.5	20.3	23.2
0.3	28.2	46.9	24.6	0.1	24.4	48.8	26.8	10.9	22.8	42.7	23.6	0.0	19.1	44.3	36.6	0.0	22.8	45.1	32.1
0.3	28.5	24.3	46.9	0.2	24.1	27.0	48.8	12.1	22.1	26.7	39.1	0.0	18.5	33.8	47.8	0.0	22.3	28.7	49.0

(b) $U1$

Figure 28: Classification matrices for (a) $E1$ and (b) $U1$ distributions with 15 training observations in each group. Classification matrices for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j + 1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
79.8	13.4	3.5	3.4	61.7	24.2	7.1	7.0	32.7	38.9	14.3	14.0	0.4	51.3	23.7	24.6
79.4	3.3	14.1	3.2	62.0	7.0	24.0	7.0	33.3	14.1	38.5	14.1	0.5	23.9	51.1	24.6
79.7	3.3	3.6	13.4	61.8	7.2	7.1	23.9	33.3	14.4	13.8	38.4	0.4	23.5	23.4	52.7

LDF				QDF				CART				SVMlinear				SVMrbf			
0.2	53.3	23.4	23.1	0.2	52.1	24.3	23.4	12.8	40.7	24.7	21.8	0.0	28.0	35.5	36.5	0.0	41.8	27.7	30.5
0.2	23.1	53.6	23.1	0.2	24.0	52.5	23.3	12.1	23.4	42.8	21.7	0.0	8.9	60.8	30.3	0.0	20.7	50.3	29.0
0.3	23.2	23.3	53.2	0.2	24.3	24.1	51.4	13.8	23.2	23.8	39.2	0.0	9.0	28.4	62.7	0.0	20.1	26.1	53.7

(a) $E1$

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
88.4	6.1	2.7	2.8	76.1	12.7	5.5	5.7	48.0	26.7	12.4	12.9	0.5	57.6	20.4	21.5
81.9	2.5	11.6	4.0	65.6	5.7	20.6	8.1	37.2	13.0	34.7	15.1	0.5	31.1	44.7	23.6
82.2	2.4	4.4	11.0	65.8	5.5	8.3	20.4	37.6	12.6	15.3	34.5	0.5	30.6	22.9	46.1

LDF				QDF				CART				SVMlinear				SVMrbf			
0.2	53.7	22.7	23.3	0.2	68.8	15.4	15.7	11.0	51.8	19.3	17.9	0.0	32.8	33.9	33.3	0.0	59.8	18.3	22.0
0.3	29.1	47.6	23.0	0.2	25.2	49.2	25.4	9.9	21.4	43.6	25.1	0.0	18.3	48.2	33.4	0.0	22.8	45.9	31.3
0.3	28.7	23.3	47.7	0.2	25.1	25.2	49.5	10.7	21.4	26.3	41.7	0.0	17.5	35.8	46.7	0.0	22.2	27.3	50.5

(b) $U1$

Figure 29: Classification matrices for (a) $E1$ and (b) $U1$ distributions with 25 training observations in each group. Classification matrices for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j + 1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

DAMIP 0%					DAMIP 5%				DAMIP 15%				DAMIP 100%			
	85.9	9.8	2.2	2.2	61.6	24.5	7.2	6.7	24.9	42.9	16.2	15.9	0.3	52.1	24.4	23.1
	85.6	2.1	10.3	2.0	61.7	6.8	24.9	6.6	24.9	15.7	43.3	16.1	0.3	23.2	53.2	23.3
	85.7	2.1	2.3	9.9	61.9	6.9	6.9	24.3	24.6	16.0	16.0	43.3	0.3	23.4	24.1	52.2

LDF				QDF				CART				SVMlinear				SVMrbf			
0.2	53.6	23.4	22.8	0.2	53.1	23.8	22.9	12.3	41.6	24.7	21.4	0.0	27.8	36.2	36.0	0.0	44.7	27.7	27.6
0.2	22.8	54.2	22.8	0.2	23.4	53.7	22.7	12.7	23.7	42.4	21.2	0.0	8.5	62.7	28.8	0.0	20.4	53.0	26.5
0.3	22.9	23.1	53.8	0.2	23.6	23.6	52.5	13.6	24.1	23.8	38.5	0.0	8.3	28.7	63.0	0.0	20.3	26.4	53.3

(a) $E1$

DAMIP 0%					DAMIP 5%				DAMIP 15%				DAMIP 100%			
	93.6	3.2	1.6	1.6	77.3	12.2	5.1	5.4	40.6	29.8	14.9	14.6	0.4	60.4	19.6	19.7
	87.7	1.3	8.3	2.7	65.1	5.6	21.3	8.0	29.0	15.0	39.1	16.9	0.4	32.6	44.5	22.4
	87.7	1.4	2.8	8.1	65.0	5.7	7.9	21.4	29.1	15.0	16.6	39.4	0.4	32.6	21.5	45.5

LDF				QDF				CART				SVMlinear				SVMrbf			
0.3	53.8	22.8	23.1	0.2	71.3	14.2	14.4	10.2	52.1	20.5	17.2	0.0	31.5	32.5	36.0	0.0	64.4	16.7	18.9
0.3	29.2	47.7	22.9	0.2	26.0	49.0	24.9	9.5	21.2	44.4	24.9	0.0	18.8	44.1	37.1	0.0	24.2	46.0	29.8
0.3	29.1	21.9	48.7	0.2	25.9	24.0	49.9	10.8	20.7	27.5	41.0	0.0	17.6	31.7	50.7	0.0	24.0	26.0	50.1

(b) $U1$

Figure 30: Classification matrices for (a) $E1$ and (b) $U1$ distributions with 40 training observations in each group. Classification matrices for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j + 1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
92.7	5.5	1.0	0.9	65.6	22.7	5.9	5.7	27.6	42.5	14.9	15.0	0.1	53.4	23.6	22.9
92.3	0.9	5.8	0.9	65.3	5.6	23.2	5.8	27.3	15.1	42.6	15.0	0.1	23.1	53.9	22.9
92.7	0.9	1.0	5.5	65.4	5.8	5.9	22.9	27.4	15.0	15.3	42.2	0.1	23.5	23.4	53.0

LDF				QDF				CART				SVMlinear				SVMrbf			
0.2	54.4	23.0	22.5	0.2	54.3	22.9	22.6	12.5	41.8	24.3	21.5	0.0	27.1	36.8	36.1	0.0	46.8	25.6	27.6
0.2	22.6	54.5	22.6	0.2	22.9	54.2	22.8	12.3	23.8	42.4	21.5	0.0	7.4	64.5	28.0	0.0	19.2	54.3	26.5
0.3	22.9	23.0	53.9	0.3	23.3	23.0	53.4	13.0	24.2	24.3	38.6	0.0	7.5	29.0	63.5	0.0	19.3	24.6	56.2

(a) $E1$

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
97.8	0.9	0.6	0.7	81.6	9.8	4.2	4.4	46.3	27.5	13.2	13.1	0.2	64.9	17.0	17.9
93.7	0.6	4.5	1.2	69.0	4.9	19.4	6.7	31.6	14.2	38.2	16.0	0.2	35.0	43.3	21.5
93.7	0.6	1.3	4.4	68.9	4.9	6.8	19.3	31.9	14.3	15.9	38.0	0.2	35.1	20.0	44.7

LDF				QDF				CART				SVMlinear				SVMrbf			
0.2	53.8	22.8	23.2	0.1	73.9	12.9	13.1	9.8	53.4	19.4	17.5	0.0	23.3	36.9	39.8	0.0	70.1	13.7	16.2
0.2	29.6	48.6	21.5	0.2	26.7	49.3	23.8	9.6	20.4	44.4	25.7	0.0	16.3	49.4	34.3	0.0	26.0	47.1	26.9
0.3	29.7	21.7	48.3	0.2	26.9	23.6	49.3	10.4	20.4	27.8	41.5	0.0	14.3	32.9	52.9	0.0	25.8	25.0	49.1

(b) $U1$

Figure 31: Classification matrices for (a) $E1$ and (b) $U1$ distributions with 100 training observations in each group. Classification matrices for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j + 1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

B.3.2 One large group

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
0%				5%				15%				100%			
59.3	38.4	1.1	1.1	53.0	38.3	4.3	4.4	43.0	38.3	9.3	9.5	0.1	97.9	1.0	0.9
74.5	17.0	6.4	2.1	59.4	17.0	16.9	6.7	44.7	16.9	26.4	11.9	0.6	90.6	6.0	2.8
74.6	16.8	2.0	6.6	59.9	16.6	6.2	17.3	44.4	16.5	11.3	27.7	0.6	90.9	2.8	5.7

LDF				QDF				CART				SVMlinear				SVMrbf			
0.0	99.6	0.2	0.2	0.0	99.3	0.4	0.3	0.0	100.0	0.0	0.0	0.0	49.4	26.8	23.8	0.0	98.7	0.6	0.7
0.0	97.0	2.2	0.8	0.0	96.7	2.3	1.0	0.0	100.0	0.0	0.0	0.0	40.3	33.0	26.6	0.0	98.1	1.0	0.9
0.0	97.1	0.8	2.1	0.0	96.9	1.1	2.0	0.0	100.0	0.0	0.0	0.0	40.0	26.8	33.1	0.0	98.1	0.6	1.3

(a) $E1$

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
75.7	22.6	0.9	0.8	72.9	22.4	2.3	2.4	66.9	22.5	5.5	5.2	0.1	98.5	0.7	0.7
63.8	12.4	14.0	9.9	55.8	12.2	18.9	13.1	50.6	12.2	22.6	14.5	1.3	69.2	16.6	12.9
64.4	11.9	10.2	13.6	56.0	11.7	13.7	18.5	51.5	11.7	16.1	20.8	1.2	69.1	13.2	16.5

LDF				QDF				CART				SVMlinear				SVMrbf			
0.0	99.7	0.2	0.2	0.0	99.3	0.4	0.3	0.0	99.8	0.1	0.1	0.0	49.6	28.4	22.0	0.0	97.3	1.2	1.5
0.0	77.8	12.0	10.2	0.0	70.5	17.2	12.3	0.7	94.8	1.4	3.1	0.0	39.5	33.0	27.5	0.0	87.7	6.8	5.5
0.0	77.7	10.5	11.7	0.0	70.7	12.6	16.7	0.7	95.0	1.2	3.1	0.0	39.8	32.2	28.0	0.0	88.0	5.8	6.2

(b) $U1$

Figure 32: Classification matrices for (a) $E1$ and (b) $U1$ distributions with 100 training observations in group 1 and 5 training observations each in groups 2 and 3. Classification matrices for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j + 1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
70.8	26.8	1.2	1.2	64.4	26.6	4.6	4.4	31.0	44.5	12.4	12.1	0.2	96.1	1.9	1.8
82.2	8.9	6.8	2.0	67.7	8.8	17.5	6.0	32.2	18.3	34.7	14.8	0.7	84.2	10.3	4.7
82.0	9.2	1.7	7.1	68.1	9.0	5.4	17.5	32.3	18.7	14.2	34.8	0.8	84.5	4.4	10.4

LDF				QDF				CART				SVMlinear				SVMrbf			
0.0	98.8	0.6	0.6	0.0	98.6	0.7	0.7	0.0	99.7	0.2	0.1	0.0	48.0	28.1	23.9	0.0	97.6	1.0	1.4
0.0	93.3	4.9	1.8	0.0	93.2	4.7	2.1	0.2	99.0	0.6	0.2	0.0	32.6	39.9	27.4	0.0	95.1	2.7	2.3
0.0	93.4	1.6	5.0	0.0	93.5	1.9	4.6	0.1	99.1	0.4	0.4	0.0	33.1	29.8	37.0	0.0	95.1	1.5	3.4

(a) $E1$

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
91.0	7.8	0.6	0.6	87.9	7.7	2.1	2.3	56.5	27.0	8.6	8.0	0.1	97.3	1.4	1.2
73.6	5.3	12.9	8.2	67.6	5.2	17.1	10.0	38.5	13.9	29.1	18.5	0.8	63.3	21.2	14.7
73.4	5.3	7.6	13.7	67.4	5.3	9.5	17.8	39.0	13.8	17.0	30.1	0.7	63.3	14.3	21.6

LDF				QDF				CART				SVMlinear				SVMrbf			
0.0	99.5	0.3	0.3	0.0	98.6	0.7	0.7	0.1	99.1	0.4	0.4	0.0	49.0	26.4	24.6	0.0	95.9	1.5	2.6
0.0	75.2	13.1	11.7	0.1	63.3	22.1	14.5	1.9	79.4	9.5	9.2	0.0	34.4	32.1	33.5	0.0	77.5	12.1	10.4
0.1	75.0	10.6	14.4	0.1	63.3	13.5	23.2	2.0	78.9	8.1	11.1	0.0	35.1	29.6	35.3	0.0	77.6	9.9	12.5

(b) $U1$

Figure 33: Classification matrices for (a) $E1$ and (b) $U1$ distributions with 100 training observations in group 1 and 10 training observations each in groups 2 and 3. Classification matrices for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j + 1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
76.8	20.7	1.2	1.3	70.5	20.6	4.4	4.5	26.7	46.3	13.5	13.5	0.2	94.3	2.7	2.8
84.6	6.2	7.2	2.0	71.2	6.2	17.4	5.3	27.1	19.4	38.1	15.5	0.7	79.5	13.9	5.9
84.5	6.2	1.7	7.6	70.9	6.1	4.8	18.2	27.1	19.0	14.8	39.1	0.7	78.8	5.7	14.8

LDF				QDF				CART				SVMlinear				SVMrbf			
0.0	97.6	1.1	1.3	0.0	97.2	1.3	1.4	0.1	98.9	0.6	0.5	0.0	46.7	27.1	26.2	0.0	96.1	1.6	2.4
0.0	88.7	8.2	3.1	0.0	88.6	8.0	3.5	0.3	96.3	2.2	1.2	0.0	29.3	40.6	30.1	0.0	91.1	5.1	3.8
0.0	88.0	2.9	9.1	0.0	88.2	3.2	8.6	0.3	96.5	1.2	2.0	0.0	26.1	29.0	44.8	0.0	90.9	2.9	6.3

(a) $E1$

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
94.1	4.8	0.6	0.6	90.8	4.7	2.2	2.3	51.8	29.3	9.5	9.3	0.1	96.1	1.9	1.9
78.0	3.9	11.6	6.6	73.2	3.9	14.9	8.0	32.6	14.9	34.3	18.2	0.7	60.0	24.4	14.9
78.1	3.7	6.0	12.2	73.1	3.8	7.2	16.0	33.0	14.8	19.5	32.7	0.8	59.8	15.5	23.9

LDF				QDF				CART				SVMlinear				SVMrbf			
0.0	99.2	0.4	0.4	0.0	97.8	1.1	1.1	0.0	98.7	0.6	0.6	0.0	48.2	27.3	24.5	0.0	94.1	1.9	4.0
0.0	72.7	15.9	11.4	0.0	58.9	26.1	15.0	1.2	73.8	14.6	10.4	0.0	31.4	37.9	30.7	0.0	68.1	17.0	14.9
0.0	72.6	12.0	15.4	0.0	58.8	14.9	26.2	1.2	73.6	10.3	14.9	0.0	31.2	35.6	33.2	0.0	68.1	13.5	18.4

(b) $U1$

Figure 34: Classification matrices for (a) $E1$ and (b) $U1$ distributions with 100 training observations in group 1 and 15 training observations each in groups 2 and 3. Classification matrices for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j + 1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
84.7	13.0	1.2	1.1	66.4	23.5	5.1	5.0	29.0	43.3	13.8	13.8	0.2	86.8	6.5	6.5
88.0	3.0	7.5	1.5	66.8	6.5	21.1	5.6	28.8	16.1	40.3	14.8	0.5	62.7	26.1	10.6
88.3	3.1	1.7	6.9	67.4	6.7	5.7	20.3	28.8	16.3	15.0	39.9	0.5	62.3	11.0	26.2

LDF				QDF				CART				SVMlinear				SVMrbf			
0.0	91.7	4.2	4.1	0.0	91.2	4.4	4.4	0.3	90.4	4.5	4.8	0.0	43.6	28.5	27.9	0.0	85.9	5.6	8.5
0.0	71.2	21.3	7.6	0.0	71.5	20.5	8.0	0.6	78.9	13.6	6.9	0.0	21.6	49.2	29.3	0.0	70.0	17.5	12.5
0.0	71.3	7.9	20.9	0.0	71.2	8.4	20.4	0.7	78.6	7.0	13.7	0.0	21.0	29.6	49.4	0.0	69.9	9.8	20.4

(a) $E1$

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
97.1	1.8	0.5	0.6	87.8	5.9	3.2	3.1	51.5	26.2	11.4	11.0	0.2	92.8	3.6	3.4
85.5	1.9	9.1	3.6	68.7	4.5	18.8	8.0	32.1	13.3	37.1	17.5	0.4	54.5	28.9	16.2
85.5	1.9	3.9	8.7	68.7	4.5	7.9	18.9	32.0	13.4	17.8	36.8	0.4	54.5	15.4	29.8

LDF				QDF				CART				SVMlinear				SVMrbf			
0.0	97.2	1.5	1.3	0.0	95.0	2.5	2.5	0.2	93.7	3.2	3.0	0.0	45.8	28.3	25.9	0.0	90.7	3.7	5.6
0.0	65.8	21.4	12.7	0.0	50.2	33.1	16.6	1.6	58.5	24.9	15.1	0.0	27.2	37.9	34.8	0.0	53.6	26.0	20.3
0.0	65.9	12.9	21.2	0.0	50.1	16.8	33.1	1.7	58.5	14.9	24.9	0.0	26.6	34.6	38.8	0.0	53.4	18.0	28.6

(b) $U1$

Figure 35: Classification matrices for (a) $E1$ and (b) $U1$ distributions with 100 training observations in group 1 and 30 training observations each in groups 2 and 3. Classification matrices for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j + 1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
88.3	9.5	1.1	1.1	66.5	22.5	5.4	5.7	28.2	42.7	14.5	14.7	0.1	76.9	11.3	11.6
90.2	1.9	6.6	1.3	67.0	5.9	21.6	5.5	27.9	15.7	41.7	14.7	0.2	47.4	37.2	15.2
90.3	2.0	1.3	6.4	66.8	5.8	5.7	21.6	28.1	15.4	15.3	41.3	0.2	47.0	15.8	36.9

LDF				QDF				CART				SVMlinear				SVMrbf			
0.1	80.8	9.7	9.5	0.1	79.8	10.1	10.0	5.4	74.8	10.3	9.6	0.0	34.2	32.9	32.9	0.0	71.8	12.5	15.8
0.1	51.2	35.4	13.3	0.1	50.6	35.4	13.8	7.4	56.2	24.4	12.0	0.0	13.2	58.4	28.3	0.0	46.3	34.1	19.6
0.2	51.1	14.0	34.7	0.2	50.6	14.6	34.6	7.8	56.3	13.4	22.5	0.0	13.5	28.9	57.6	0.0	46.1	17.4	36.5

(a) $E1$

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
97.7	1.2	0.6	0.5	86.0	6.8	3.6	3.6	47.8	27.7	12.5	11.9	0.2	87.1	6.3	6.4
89.4	1.1	7.1	2.3	69.5	4.4	19.0	7.1	30.7	14.2	38.4	16.8	0.3	48.8	33.3	17.5
89.8	1.2	2.6	6.5	69.5	4.6	7.2	18.7	30.9	14.3	16.7	38.1	0.3	48.8	16.4	34.5

LDF				QDF				CART				SVMlinear				SVMrbf			
0.1	89.7	5.3	5.0	0.0	90.3	4.8	4.9	4.3	81.4	7.5	6.9	0.0	42.8	29.3	27.8	0.0	86.7	6.0	7.3
0.2	53.7	30.6	15.5	0.1	42.0	39.0	18.9	6.0	43.4	31.6	19.1	0.0	23.3	40.7	36.0	0.0	43.1	33.8	23.1
0.2	53.9	14.9	31.0	0.1	42.0	19.0	38.9	6.5	43.4	19.6	30.5	0.0	25.8	34.7	39.4	0.0	43.0	20.6	36.4

(b) $U1$

Figure 36: Classification matrices for (a) $E1$ and (b) $U1$ distributions with 100 training observations in group 1 and 50 training observations each in groups 2 and 3. Classification matrices for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j + 1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

B.3.3 Two large groups

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
88.0	9.7	1.6	0.6	63.5	26.4	6.4	3.8	34.9	40.6	13.9	10.6	0.7	67.4	31.2	0.7
87.8	1.6	9.9	0.7	63.6	6.6	26.0	3.8	35.2	14.3	40.0	10.6	0.8	31.5	67.1	0.7
90.6	2.6	2.7	4.1	66.8	9.2	9.4	14.6	36.9	16.5	16.8	29.8	1.6	47.0	47.7	3.7

LDF				QDF				CART				SVMlinear				SVMrbf			
0.1	69.1	30.7	0.1	0.1	69.0	30.7	0.1	4.2	59.4	36.4	0.0	0.0	36.5	49.5	13.9	0.0	58.7	41.1	0.2
0.1	31.2	68.6	0.1	0.1	31.7	68.1	0.1	4.3	36.5	59.2	0.0	0.0	11.4	83.7	5.0	0.0	23.4	76.5	0.2
0.1	49.2	50.0	0.7	0.1	49.4	49.7	0.8	5.2	47.0	47.7	0.0	0.0	20.9	67.9	11.2	0.0	41.3	58.0	0.7

(a) $E1$

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
96.3	2.1	0.9	0.7	81.0	11.5	4.5	2.9	54.0	26.3	10.7	9.0	0.2	80.7	18.3	0.7
79.5	1.0	18.9	0.7	61.0	5.8	29.1	4.0	40.5	12.9	35.7	10.8	0.1	39.9	59.5	0.5
83.4	1.1	12.8	2.6	65.2	6.1	18.0	10.7	43.0	12.8	20.7	23.5	0.3	45.9	51.9	2.0

LDF				QDF				CART				SVMlinear				SVMrbf			
0.1	74.7	25.2	0.0	0.1	82.5	17.4	0.1	2.9	68.5	28.6	0.0	0.0	33.1	48.1	18.9	0.0	71.3	27.5	1.1
0.1	33.8	66.0	0.1	0.1	31.1	68.6	0.2	2.5	28.7	68.8	0.0	0.0	12.2	78.4	9.4	0.0	24.1	75.1	0.8
0.1	40.6	59.2	0.1	0.1	39.5	59.5	0.9	2.8	35.5	61.7	0.0	0.0	18.7	73.7	7.6	0.0	33.1	66.3	0.6

(b) $U1$

Figure 37: Classification matrices for (a) $E1$ and (b) $U1$ distributions with 100 training observations each in groups 1 and 2 and 5 training observations groups 3. Classification matrices for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j + 1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
88.2	9.5	1.6	0.7	65.7	23.9	6.1	4.2	29.3	43.3	15.3	12.2	0.1	66.6	32.1	1.2
88.0	1.6	9.8	0.7	65.9	6.0	23.7	4.3	29.3	15.4	42.9	12.4	0.2	30.7	68.1	1.1
90.8	2.5	2.6	4.1	67.9	7.7	7.5	16.8	30.2	17.9	17.5	34.3	0.7	45.6	47.7	5.9

LDF				QDF				CART			SVMlinear				SVMrbf			
0.1	68.6	31.1	0.2	0.1	68.2	31.4	0.3	4.6	59.5	35.9	0.0	0.0	34.4	49.5	16.1	0.0	58.1	41.5	0.4
0.1	30.8	68.9	0.2	0.1	31.0	68.7	0.2	4.6	36.5	58.8	0.0	0.0	10.7	83.6	5.7	0.0	23.0	76.7	0.4
0.2	48.8	49.5	1.6	0.1	48.6	49.6	1.6	6.1	47.2	46.7	0.0	0.0	20.5	66.5	13.0	0.0	40.5	57.8	1.7

(a) $E1$

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
96.9	1.6	0.8	0.6	83.8	9.5	4.1	2.7	47.1	30.0	12.6	10.3	0.2	81.1	17.6	1.1
82.9	1.0	15.3	0.8	67.0	5.3	23.1	4.6	32.9	14.7	40.6	11.7	0.1	40.6	58.5	0.9
87.1	1.2	8.5	3.2	70.2	5.5	11.4	13.0	34.7	14.8	22.1	28.4	0.3	46.1	50.0	3.6

LDF				QDF				CART			SVMlinear				SVMrbf			
0.1	74.9	24.9	0.1	0.1	82.6	17.2	0.1	3.5	68.4	28.1	0.0	0.0	33.1	48.5	18.4	0.0	71.8	27.1	1.1
0.1	34.1	65.7	0.0	0.1	31.4	68.2	0.3	3.2	28.8	68.0	0.0	0.0	12.4	78.6	9.1	0.0	24.5	74.7	0.8
0.1	40.5	59.3	0.1	0.1	39.3	58.7	2.0	3.7	36.1	60.2	0.0	0.0	18.4	74.0	7.6	0.0	33.5	66.0	0.6

(b) $U1$

Figure 38: Classification matrices for (a) $E1$ and (b) $U1$ distributions with 100 training observations each in groups 1 and 2 and 10 training observations groups 3. Classification matrices for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j+1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

	DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
	88.4	9.3	1.5	0.8	69.0	20.9	5.4	4.7	27.7	43.2	15.6	13.4	0.2	66.6	31.3	1.9
	88.7	1.6	8.9	0.8	68.6	5.2	21.4	4.9	27.0	15.4	44.2	13.3	0.2	31.0	67.0	1.9
	90.6	2.4	2.1	4.9	68.4	6.2	6.3	19.1	27.2	17.4	18.0	37.4	0.6	44.9	45.3	9.2

	LDF				QDF				CART				SVMlinear				SVMrbf			
0.1	68.3	31.1	0.4		0.1	67.9	31.4	0.5	4.8	58.8	36.4	0.0	0.0	34.4	49.6	16.0	0.0	58.2	41.0	0.8
0.1	30.9	68.6	0.4		0.1	31.0	68.4	0.5	4.9	36.4	58.7	0.0	0.0	10.7	83.3	5.9	0.0	23.4	75.9	0.7
0.1	48.3	48.4	3.2		0.2	48.1	48.5	3.3	6.0	46.9	47.1	0.1	0.0	20.3	64.9	14.8	0.0	40.8	55.9	3.3

(a) $E1$

	DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
	97.1	1.6	0.7	0.7	85.3	7.6	3.9	3.2	45.1	30.4	13.1	11.4	0.3	79.9	18.5	1.4
	85.5	1.0	12.6	0.9	71.5	4.4	19.0	5.1	31.7	14.7	41.0	12.6	0.1	40.0	58.8	1.2
	88.8	1.1	6.5	3.6	72.8	4.4	8.2	14.6	32.1	15.5	21.3	31.1	0.4	45.7	48.8	5.0

	LDF				QDF				CART				SVMlinear				SVMrbf			
0.1	74.1	25.5	0.2		0.1	82.6	17.1	0.2	4.3	67.8	27.9	0.0	0.0	32.7	48.6	18.7	0.0	71.9	27.1	1.0
0.1	33.9	66.0	0.1		0.1	31.4	67.9	0.6	3.8	28.5	67.7	0.0	0.0	12.3	78.7	8.9	0.0	24.5	74.7	0.7
0.1	40.8	58.8	0.3		0.1	39.3	57.3	3.4	4.9	35.3	59.7	0.1	0.0	18.7	73.5	7.8	0.0	33.8	65.6	0.6

(b) $U1$

Figure 39: Classification matrices for (a) $E1$ and (b) $U1$ distributions with 100 training observations each in groups 1 and 2 and 15 training observations groups 3. Classification matrices for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j + 1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

DAMIP 0%					DAMIP 5%				DAMIP 15%				DAMIP 100%			
	90.3	7.7	1.3	0.7	66.2	23.0	5.7	5.1	27.9	42.9	14.9	14.3	0.2	65.5	30.1	4.2
	90.1	1.3	7.8	0.8	66.4	5.8	22.6	5.2	27.8	15.2	42.9	14.1	0.2	30.0	65.8	4.1
	92.0	1.7	1.7	4.6	67.0	6.3	6.0	20.7	27.7	16.1	16.1	40.1	0.4	41.0	41.9	16.6

LDF				QDF				CART				SVMlinear				SVMrbf			
0.1	67.9	30.2	1.9	0.1	67.4	30.4	2.0	6.1	57.9	34.9	1.1	0.0	31.5	49.2	19.2	0.0	58.3	38.7	3.0
0.1	30.5	67.4	1.9	0.1	30.7	67.2	2.0	6.1	35.1	57.6	1.2	0.0	9.4	82.8	7.7	0.0	24.0	73.4	2.6
0.1	44.8	44.6	10.4	0.2	45.1	44.8	9.9	7.2	44.8	44.5	3.5	0.0	16.3	61.6	22.1	0.0	39.2	51.4	9.4

(a) $E1$

DAMIP 0%					DAMIP 5%				DAMIP 15%				DAMIP 100%			
	97.7	1.1	0.6	0.6	84.3	8.0	4.3	3.5	47.0	28.4	12.8	11.8	0.3	79.1	17.4	3.2
	89.2	0.9	8.8	1.1	68.5	4.8	20.5	6.2	31.7	14.4	39.7	14.2	0.2	40.1	56.5	3.2
	91.5	0.9	3.5	4.0	70.0	4.7	8.0	17.3	32.4	14.3	18.4	34.9	0.4	45.6	42.2	11.7

LDF				QDF				CART				SVMlinear				SVMrbf			
0.1	73.3	25.4	1.2	0.1	81.9	17.1	0.9	4.1	67.2	27.8	0.9	0.0	31.1	48.8	20.1	0.0	73.2	25.7	1.1
0.1	33.7	65.9	0.3	0.1	30.8	66.4	2.8	3.9	27.9	65.9	2.3	0.0	11.7	79.2	9.0	0.0	25.3	73.8	0.9
0.1	40.9	57.4	1.6	0.1	37.8	50.9	11.2	4.5	34.5	55.0	6.1	0.0	17.5	73.1	9.4	0.0	35.1	63.6	1.3

(b) $U1$

Figure 40: Classification matrices for (a) $E1$ and (b) $U1$ distributions with 100 training observations each in groups 1 and 2 and 30 training observations groups 3. Classification matrices for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j + 1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
90.7	7.3	1.1	0.9	66.4	22.4	5.6	5.5	27.3	43.0	15.2	14.5	0.1	62.4	28.6	8.9
91.0	1.3	6.8	0.9	66.4	5.9	22.2	5.4	27.0	15.4	42.9	14.6	0.1	28.4	62.6	8.9
91.9	1.5	1.3	5.4	66.2	5.9	5.8	22.2	27.0	15.8	15.6	41.5	0.2	35.6	35.0	29.2

	LDF				QDF				CART				SVMlinear				SVMrbf			
0.1	64.8	28.7	6.4	0.1	64.3	28.8	6.8	7.5	55.4	32.8	4.3	0.0	29.5	47.4	23.0	0.0	56.0	35.6	8.4
0.1	28.7	64.7	6.5	0.1	28.6	64.4	6.9	7.7	33.3	54.6	4.4	0.0	8.3	80.6	11.1	0.0	23.2	69.1	7.8
0.2	37.4	37.3	25.1	0.2	37.4	37.3	25.1	9.5	39.6	39.0	12.0	0.0	10.8	54.6	34.6	0.0	33.2	43.6	23.2

(a) $E1$

DAMIP 0%				DAMIP 5%				DAMIP 15%				DAMIP 100%			
97.7	1.0	0.6	0.6	83.2	8.5	4.5	3.8	46.0	28.5	13.3	12.1	0.4	75.9	17.9	5.8
91.3	0.7	6.8	1.2	69.6	4.7	19.6	6.1	31.9	14.0	38.7	15.4	0.3	38.9	53.3	7.4
92.4	0.8	2.4	4.4	70.3	4.8	7.1	17.8	31.4	14.8	16.9	37.0	0.5	43.3	34.5	21.8

	LDF				QDF				CART				SVMlinear				SVMrbf			
0.2	70.2	25.4	4.3	0.1	81.3	15.8	2.8	5.4	65.1	26.2	3.3	0.0	28.8	49.1	22.2	0.0	74.7	23.7	1.6
0.1	33.4	64.6	1.8	0.1	30.6	62.1	7.2	5.6	26.6	60.8	7.0	0.0	11.6	79.8	8.6	0.0	26.4	71.5	2.1
0.2	40.2	50.6	9.1	0.1	35.6	41.4	22.9	7.2	31.3	46.6	14.9	0.0	16.7	71.7	11.7	0.0	36.1	58.7	5.2

(b) $U1$

Figure 41: Classification matrices for (a) $E1$ and (b) $U1$ distributions with 100 training observations each in groups 1 and 2 and 50 training observations groups 3. Classification matrices for linear discriminant functions (LDF), quadratic discriminant functions (QDF), classification trees (CART), support vector machines with a linear kernel (SVMlinear), support vector machines with a radial basis function kernel (SVMrbf), and DAMIP with training misclassification limits of 0, 5, 15, and 100% (DAMIP0, DAMIP5, DAMIP15, and DAMIP100, resp.). The rows of each 3×4 matrix correspond to the known group membership of observations, and the columns correspond to the allocated group. The percentage in row i and column $j + 1$ denotes the percentage of observations from group i allocated by the method to group j. The first column corresponds to the reserved judgment group; these observations are not classified. The data set and methods are described in Chapter 6.

References

[1] Anderson, J., "Constrained discrimination between k populations," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 31, pp. 123–139, 1969.

[2] Applegate, D., , Bixby, B., Chvátal, V., and Cook, W., "Finding cuts in the TSP (A preliminary report)," Tech. Rep. 95-05, DIMACS Technical Report, 1995.

[3] Atamturk, A., *Conflict graphs and flow models for mixed-integer linear optimization problems*. PhD thesis, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, 1998.

[4] Atamturk, A., "Strong formulations of robust mixed 0-1 programming," tech. rep., University of California at Berkeley, 2003.

[5] Atamtürk, A., Nemhauser, G. L., and Savelsbergh, M. W., "Conflict graphs in solving integer programming problems," *European Journal of Operations Research*, vol. 121, pp. 40–55, 2000.

[6] Banks, W. J. and Abad, P. L., "An efficient optimal solution algorithm for the classification problem," *Decision Sciences*, vol. 22, pp. 1008–1023, 1991.

[7] Ben-Tal, A. and Nemirovski, A., "Robust convex optimization," *Mathematics of Operations Research*, vol. 23, pp. 769–805, 1998.

[8] Ben-Tal, A. and Nemirovski, A., "Robust solutions of uncertain linear programs," *Operations Research Letters*, vol. 25, pp. 1–13, 1999.

[9] Ben-Tal, A. and Nemirovski, A., "Robust solutions of linear programming problems contaminated with uncertain data," *Mathematical Programming, Series A*, vol. 88, pp. 411–424, 2000.

[10] Berge, C., *Graphs and hypergraphs*. Elsevier, 1976. Translated by Edward Minieka.

[11] Bertsimas, D. and Sim, M., "Robust discrete optimization and network flows," *Mathematatical Programming, Series B*, vol. 98, pp. 49–71, 2003.

[12] Bixby, R. E. and Lee, E. K., "Solving a truck dispatching scheduling problem using branch-and-cut," *Operations Research*, 1998.

[13] Bollobás, B., *Graph Theory: An Introductory Course*. Springer-Verlag, 1979.

[14] Borndörfer, R., *Aspects of set packing, partitioning and covering*. PhD thesis, Technischen Universität Berlin, Berlin, Germany, 1997.

[15] Borndörfer, R. and Weismantel, R., "Set packing relaxations of some integer programs," *Mathematical Programming*, vol. 88, pp. 425–450, 2000.

[16] Breiman, L., "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.

[17] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J., *Classification and Regression Trees*. Wadsworth and Brooks/Cole, 1984.

[18] Breiman, L., "Heuristics of instability an stabilization in model selection," *The Annals of Statistics*, vol. 6, pp. 2350–2383, 1996.

[19] Breiman, L., "Arcing classifiers," *Annals of Statistics*, vol. 26, pp. 801–824, 1998.

[20] Broffit, J., Randles, R., and Hogg, R., "Distribution-free partial discriminant analysis," *Journal of the American Statistical Association*, vol. 71, pp. 934–939, 1976.

[21] Bron, C. and Kerbosch, J., "Algorithm 457: Finding all cliques of an undirected graph," *Communications of the ACM*, vol. 16, pp. 575–577, 1973.

[22] Chen, C. and Mangasarian, O., "Hybrid misclassification minimization," *Advances in Computational Mathematics*, vol. 5, pp. 127–136, 1996.

[23] Chow, C., "On optimum recognition error and reject tradeoff," *IEEE Transactions on Information Theory*, 1970.

[24] Cormen, T. H., Leiserson, C. E., and Rivest, R. L., *Introduction to Algorithms*. McGraw-Hill, 1990.

[25] Cortes, C. and Vapnik, V., "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[26] Devroye, L., Györfi, L., and Lugosi, G., *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

[27] Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern Classification*. Wiley, 2001.

[28] Easton, T., Hooker, K., and Lee, E., "Facets of the independent set plytope," *Mathematical Programming, Series B*, vol. 98, pp. 177–199, 2003.

[29] Euler, R., Jünger, M., and Reinelt, G., "Generalizations of cliques, odd cycles and anticycles and their relation to independence system polyhedra," *Mathematics of Operations Research*, vol. 12, pp. 451–462, 1987.

[30] 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.

[31] Fisher, R. A., "The use of multiple measurements in taxonomy problems," *Ann. Eugenics*, vol. 7, pp. 179–188, 1936.

[32] Freed, N. and Glover, F., "A linear programming approach to the discriminant problem," *Decision Sciences*, vol. 12, pp. 68–74, 1981.

[33] Fulkerson, D., "Blocking and anti-blocking pairs of polyhedra," *Mathematical Programming*, vol. 1, pp. 168–194, 1971.

[34] G. David Forney, J., "Exponential error bounds for erasure, list, and decision feedback schemes," *IEEE Transactions on Information Theory*, vol. 14, pp. 206–220, 1968.

[35] Gallagher, R. J., Lee, E. K., and Patterson, D. A., "Constrained discriminant analysis via 0/1 mixed integer programming," *Annals of Operations Research*, vol. 74, pp. 65–88, 1997.

[36] Garey, M. R. and Johnson, D. S., *Computers and Intractability*. Freeman, 1979.

[37] Gehrlein, W., "General mathematical programming formulations for the statistical classification problem," *Operations Research Letters*, vol. 5, pp. 299–304, 1986.

[38] Gessaman, M. and Gessaman, P., "A comparison of some multivariate discrimination procedures," *Journal of the American Statistical Association*, vol. 67, pp. 468–472, 1972.

[39] Gochet, W., Stam, A., Srinivasan, V., and Chen, S., "Multigroup discriminant analysis using linear programming," *Operations Research*, vol. 45, pp. 213–225, 1997.

[40] Golumbic, M., Rotem, D., and Urritia, J., "Comparability graphs and intersection graphs," *Discrete Mathematics*, vol. 43, pp. 37–46, 1983.

[41] Györfi, L., Györfi, Z., and Vajda, I., "Bayesian decision with rejection," *Problems of Control and Information Theory*, vol. 8, pp. 445–452, 1979.

[42] Habbema, J., Hermans, J., and Burgt, A. V. D., "Cases of doubt in allocation problems," *Biometrika*, vol. 61, pp. 313–324, 1974.

[43] Hoffman, K. and Padberg, M., "Solving airline crew-scheduling problems by branch-and-cut," *Management Science*, vol. 39, pp. 667–682, 1993.

[44] Hsu, C.-W. and Lin, C.-J., "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, pp. 415–425, 2002.

[45] ILOG, *ILOG CPLEX 8.1 Reference Manual*, 2001-2002.

[46] ILOG, *ILOG CPLEX 8.1 Advanced Reference Manual*, 2002.

[47] Joachims, T., *Advances in Kernel Methods - Support Vector Learning, B*, ch. Making large-scale SVM learning practical. B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

[48] Joachimsthaler, E. A. 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.

[49] Johnson, D. and Preparata, F., "The densest hemisphere problem," *Theoretical Computer Science*, vol. 6, pp. 93–107, 1978.

[50] Johnson, E. and Padberg, M., "Degree-two inequalities, clique facets, and biperfect graphs," *Annals of Discrete Mathematics*, vol. 16, pp. 169–187, 1982.

[51] Koehler, G. J. and Erengue, S. S., "Minimizing misclassifications in linear discriminant analysis," *Decision Sciences*, vol. 21, pp. 63–85, 1990.

[52] Kouvelis, P. and Yu, G., *Robust Discrete Optimization and Its Applications*. Kluwer, 1997.

[53] Laurent, M., "A generalization of antiwebs to independence systems and their canonical facets," *Mathematical Programming*, vol. 45, pp. 97–108, 1989.

[54] Lee, E. K., *Solving a truck dispatching scheduling problem using branch-and-cut*. PhD thesis, Computational and Applied Mathematics, Rice University, Houston, Texas, 1993.

[55] Lee, E. K. and Maheshwary, S., "Conflict hypergraphs in integer programming," tech. rep., Georgia Institute of Technology, 2004.

[56] Lee, E. K., Fung, A. Y., Brooks, J. P., and Zaider, M., "Automated planning volume definition in soft-tissue sarcoma adjuvant brachytherapy," *Biology in Physics and Medicine*, vol. 47, pp. 1891–1910, 2002.

[57] Lee, E. K., Gallagher, R. J., Campbell, A. M., and Prausnitz, M. R., "Prediction of ultrasound-mediated disruption of cell membranes using machine learning techniques and statistial analysis of acoustic spectra," *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 1–9, 2004.

[58] Lee, E. K., Gallagher, R. J., and Patterson, D. A., "A linear programming approach to discriminant analysis with a reserved-judgment region," *INFORMS Journal on Computing*, vol. 15, pp. 23–41, 2003.

[59] 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.

[60] Li, R.-H. and Belford, G. G., "Instability of decision tree classification algorithms," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 570–575, ACM Press, 2002.

[61] 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.

[62] Lovász, L., "Normal hypergraphs and the perfect graph conjecture," *Discrete Mathematics*, vol. 2, pp. 253–267, 1972.

[63] Mangasarian, O. L., "Linear and nonlinear separation of patterns by linear programming," *Operations Research*, vol. 13, pp. 444–452, 1965.

[64] Mangasarian, O. L., "Multi-surface method of pattern separation," *IEEE Transactions on Information Theory*, vol. 14, pp. 801–807, 1968.

[65] Müller, K.-R., Mika, S., Rätsch, G., Tsuda, K., and Schölkopf, B., "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, pp. 181–201, March 2001.

[66] Müller, R. and Schulz, A., "Transitive packing: A unifying concept in combinatorial optimization," *SIAM Journal on Optimization*, vol. 13, pp. 335–367, 2002.

[67] Nemhauser, G. and Trotter, L., "Properties of vertex packing and independence system polyhedra," *Mathematical Programming*, vol. 6, pp. 48–61, 1974.

[68] Nemhauser, G. L. and Wolsey, L. A., *Integer and Combinatorial Optimization*. Wiley, 1999.

[69] Neyman, J. and Pearson, E., "Contributions to the theory of testing statistical hypotheses," *Stat. Res. Mem.*, vol. 1, pp. 1–37, 1936.

[70] Ng, T.-H. and Randles, R., "Distribution-free partial discrimination procedures," *Computers and Mathematics with Applications*, vol. 12A, pp. 225–234, 1986.

[71] O'Hagan, A., *Kendall's Advanced Theory of Statistics: Bayesian Inference*, vol. 2B. Halsted Press, 1994.

[72] Padberg, M., "$(1, k)$-configurations and facets for packing problems," *Mathematical programming*, vol. 18, pp. 94–99, 1973.

[73] Padberg, M., "On the facial structure of set packing polyhedra," *Mathematical Programming*, vol. 5, pp. 199–215, 1973.

[74] Papadimitriou, C. H. and Steiglitz, K., *Combinatorial Optimization: Algorithms and Complexity*. Dover, 1998.

[75] Quesenberry, C. and Gessaman, M., "Nonparametric discrimination using tolerance regions," *Annals of Mathematical Statistics*, vol. 39, pp. 664–673, 1968.

[76] Rao, C. R., *Advanced Statistical Methods in Biometric Research*. Wiley, 1952.

[77] Rencher, A. C., *Multivariate Statistical Inference and Application*. Wiley, 1998.

[78] Sekiguchi, Y., "A note on node packing polytopes on hypergraphs," *Operations Research Letters*, vol. 2, pp. 243–247, 1983.

[79] Smith, C. A. B., "Some examples of discrimination," *Ann. Eugenics*, vol. 13, pp. 272–282, 1947.

[80] Stam, A., "Nontraditional approaches to statistical classification: Some perspectives on l_p-norm methods," *Annals of Operations Research*, vol. 74, pp. 1–36, 1997.

[81] Stam, A. and Joachimsthaler, E. A., "Solving the classification problem in discriminant analysis via linear and nonlinear programming methods," *Decision Sciences*, vol. 20, pp. 285–293, 1989.

[82] Vapnik, V., *Statistical Learning Theory*. Wiley, 1998.

[83] Vapnik, V. and Chervonenkis, A., "On the uniform convergence of relative frequencies of events to their probabilities," *Theory of Probability and its Applications*, vol. 16, pp. 264–270, 1971.

[84] Vapnik, V. and Chervonenkis, A., "Necessary and sufficient conditions for the uniform convergence of means to their expectations," *Theory of Probability and its Applications*, vol. 26, pp. 532–553, 1981.

[85] Vapnik, V. and Chervonenkis, A. Y., "On a class of pattern-recognition learning algorithms," *Automation and Remote Control*, vol. 25, pp. 838–845, 1964.

[86] Vapnik, V. and Lerner, A. Y., "Pattern recognition using generalized portraits," *Automation and Remote Control*, vol. 24, pp. 709–715, 1963.

[87] Wolsey, L. A., *Integer Programming*. Wiley, 1998.

[88] Wright, A. H., *The role of integrins in the differential upregulation of tumor cell motility by endothelial extracellular matrix proteins*. PhD thesis, Georgia Institute of Technology, December 1999.

[89] Yanev, N. and Balev, S., "A combinatorial approach to the classification problem," *European Journal of Operational Research*, vol. 115, pp. 339–350, 1999.

[90] Zopounidis, C. and Doumpos, M., "Multicriteria classification and sorting methods: A literature review," *European Journal of Operational Research*, vol. 138, pp. 229–246, 2002.