Journal of Classification 21:255-277 (2004)
DOI: 10.1007/s00357-004-0019-8

Oscillation Heuristics for the Two-group Classification Problem

Ognian Asparouhov

MEDai, Inc.

Paul A. Rubin

Michigan State University

Abstract: We propose a new nonparametric family of oscillation heuristics for improving linear classifiers in the two-group discriminant problem. The heuristics are motivated by the intuition that the classification accuracy of a separating hyperplane can be improved through small perturbations to its slope and position, accomplished by substituting training observations near the hyperplane for those used to generate it. In an extensive simulation study, using data generated from multivariate normal distributions under a variety of conditions, the oscillation heuristics consistently improve upon the classical linear and logistic discriminant functions, as well as two published linear programming-based heuristics and a linear Support Vector Machine. Added to any of the methods above, they approach, and frequently attain, the best possible accuracy on the training samples, as determined by a mixed-integer programming (MIP) model, at a much smaller computational cost. They also improve expected accuracy on the overall populations when the populations overlap significantly and the heuristics are trained with large samples, at least in situations where the data conditions do not explicitly favor a particular classifier.

Keywords: Discriminant analysis; Linear discriminant function; Nonparametric classification; Mathematical programming.

The authors thank Professor Wojtek Krzanowski of Exeter University for providing helpful insights regarding the experimental design and analysis.

Authors' Addresses: P. Rubin, Department of Management, The Eli Broad Graduate School of Management, Michigan State University East Lansing, MI 48824-1122, USA, tel.: 517-432-3509, facs.: 517-432-1111, e-mail: rubin@msu.edu, http://www.msu.edu/~rubin/; O. Asparouhov, MEDai, Inc., Millennia Park One Building, 4901 Vineland Road, Suite 450, Orlando, FL 32811, USA, tel.: 321-281-4526, facs.: 321-281-4499, e-mail: OAsparoukhov@MEDai.com

1. Introduction

Consider the classical sample-based two-group linear discriminant problem (McLachlan 1992): Given two distinct groups G_1 and G_2, and a training sample of $n = n_1 + n_2$ observations (n_1 from G_1 and n_2 from G_2), each a vector of K attributes $\mathbf{x} = (x_1, \ldots, x_K)$, construct a linear classifier $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - w_0$ ($\mathbf{w} \in \Re^K$, $w_0 \in \Re$) such that $f(\mathbf{x}) > 0$ for $\mathbf{x} \in G_1$ and $f(\mathbf{x}) < 0$ for $\mathbf{x} \in G_2$ with as few exceptions as possible. Here and in the sequel $\mathbf{x} \cdot \mathbf{y}$ denotes the inner product of vectors $\mathbf{x}$ and $\mathbf{y}$. The linear classifier $f()$ corresponds to a hyperplane $H = \{\mathbf{x} \in \Re^K : \mathbf{w} \cdot \mathbf{x} = w_0\}$ that (ideally) separates the two groups. We refer to $f(\mathbf{x})$ as the classification *score* of $\mathbf{x}$.

In addition to a variety of statistical approaches to this problem, there has been considerable interest in methods based not on the joint distributions of the attributes but rather on the geometry of the separating hyperplane. In this paper, we focus on what Stam (1997) calls the *L_0-norm approach*, which minimizes the (possibly weighted) total number of misclassifications on the training sample. The majority of attacks on the L_0-norm problem to date have involved mixed-integer programming (MIP) models similar in substance to the following:

$$\begin{array}{lrcl} \text{minimize} & \sum_{i=1}^{n} c_i y_i & & \\ \text{s.t.} & \mathbf{x}_i \cdot \mathbf{w} - w_0 + M y_i & > & 0 \quad (i = 1, \ldots, n_1)\ [G_1] \\ & \mathbf{x}_i \cdot \mathbf{w} - w_0 - M y_i & < & 0 \quad (i = n_1 + 1, \ldots, n_1 + n_2)\ [G_2] \end{array} \tag{1}$$

where the classifier coefficients w_j ($j = 0, 1, \ldots K$) are unrestricted in sign, the 0-1 variable y_i takes value 1 if the i^{th} observation is misclassified and 0 otherwise, M is a sufficiently large positive constant, and the objective coefficient c_i incorporates the prior probability of $\mathbf{x}_i$ occurring and the cost of misclassifying it.

In most studies, including this one, equal prior probabilities and equal misclassification costs are assumed, so that the c_i are identical for all i and the objective value z is proportional to the total number of misclassifications in the combined training sample. Since mathematical programming models with strong inequality constraints are not well posed, analysts must relax the constraints in this generic formulation to weak inequalities, and either accept the ambiguity of an observation for which $\mathbf{w} \cdot \mathbf{x} - w_0 = 0$ or change one or both right hand sides from 0 to $\pm\varepsilon$ for some fixed $\varepsilon > 0$.

There are several studies (Banks and Abad 1991; Duarte Silva and Stam 1997; Koehler and Erenguc 1990; Rubin 1997; Soltysik and Yarnold 1994) devoted to specific MIP classification algorithms that minimize the number of misclassifications in the general case of continuously distributed attributes. Unfortunately, all known MIP classification formulations are very time consum-

ing, since the problem is *NP*-hard, and in fact is hard to approximate (Amaldi and Kann 1995), so there is no hope for polynomial-time algorithms unless $P = NP$. This computational complexity is the main reason for the development of heuristic classification procedures (Abad and Banks 1993; Conway, Cabot, and Venkataramanan 1998; Gouljashki and Asparoukhov 1999; Koehler and Erenguc 1990; Rubin 1990) that yield classifiers with accuracy competitive to that of the MIP optimal classifiers at substantially less computation cost.

While not a traditional approach, the L_0-norm approach is grounded in theory. Let $R_{\min}$ and $R(\mathbf{x};n)$ denote respectively the expected error rate of the best linear classifier given the populations and the objective value of any optimal solution to (1) for a given sample $\mathbf{x}$ of size n. Vapnik (1999) has demonstrated, based on earlier work by Vapnik and Chervonenkis (1971), that the set of linear classifiers has finite VC-dimension, and consequently $R(\mathbf{x};n)$ is a consistent estimator of $R_{\min}$. Moreover, convergence is rapid in the sense that a positive constant γ exists for which the following statement holds:

$$\forall \varepsilon > 0 \exists n_0 \ni n > n_0 \Longrightarrow \Pr\{R(\mathbf{x};n) - R_{\min} > \varepsilon\} < e^{-\gamma \varepsilon^2 n}.$$

Unfortunately, the *NP*-hard nature of (1) likely trumps the asymptotically rapid convergence.

The purpose of this paper is to propose a set of nonparametric heuristic methods to improve a linear classifier constructed by any means other than solving some variation of (1). By "improve" we mean reduce the error rate on the training sample, and thus hopefully on the population. Improvement heuristics are a recent trend, with boosting (Schapire (1999)) a prime example. Boosting adaptively weights observations that are prone to misclassification, either by altering their impact on the objective function or by adjusting their frequencies in a resampling scheme. Our approach focuses on ambiguous observations, specifically combinations of training set observations nearest (in the Euclidean sense) to either (a) the current best separating hyperplane or (b) the K points that generated the current best hyperplane. Where boosting will consistently apply a high weight to an observation from one population lying deep in the other population's territory, our heuristics will ignore that observation (other to count it as misclassified when evaluating competing classifiers) because it will lie away from the separating hyperplane.

In subsequent sections, we will: discuss relevant prior work; present our approach and give a geometric motivation for it; describe specific improvement heuristics based on this approach; describe a series of computational experiments testing the efficacy of the heuristics in improving classifiers developed by several established procedures; and present our analysis of the results of those experiments.

2. Prior Developments

In this section, we first describe a previously developed method with similar geometric roots and then discuss a procedure used within our heuristics for improving discriminant functions.

2.1 The Warmack-Gonzalez Algorithm

Warmack and Gonzalez (1973) proposed an algorithm (referred to as W-G below) for selecting a maximal consistent subset from a set of n inequalities, which Soltysik and Yarnold (1994) subsequently adapted to the linear discriminant problem under the name MultiODA. In our notation, the initial system of inequalities would be $\mathbf{w} \cdot \mathbf{x} - w_0 > 0$ ($\mathbf{x} \in G_1$), $\mathbf{w} \cdot \mathbf{x} - w_0 < 0$ ($\mathbf{x} \in G_2$). Constructing a linear classifier that minimizes the number of misclassified training observations is equivalent to determining a minimal set of those inequalities that must be violated. Finding this minimal set is a discrete optimization problem that, in general, need not have a unique solution.

Each inequality determines an open half-space in the attribute space, whose bounding hyperplane is obtained by changing the inequality to an equality. Warmack and Gonzalez focus on edges formed by the intersection of K such hyperplanes. An edge is described by a system of K homogeneous linear equations in the $K+1$ variables $w_0, \ldots, w_K$. Warmack and Gonzalez assume that their system of inequalities satisfies the *Haar condition*: every $(K+1) \times (K+1)$ matrix formed by the coefficient vectors of $K+1$ of the inequalities is nonsingular. In our context, the Haar condition means that no subset of $K+1$ training observations lies in a $K-1$ dimensional affine subspace (hyperplane) of $\Re^K$, and further that no k training observations lie in an affine subspace of dimension $k-2$ for any $k \leq K$. This condition is not restrictive if observations are not repeated, the attributes have a continuous joint distribution and there is no redundancy (multicollinearity) among them; under other circumstances, it may prove to be an issue (Rubin 1999).

The W-G algorithm iterates through sequences of edges, actively exploiting the Haar condition. In each iteration, Warmack and Gonzalez consider a set of K inequalities whose corresponding hyperplanes intersect in an edge. Although coefficient values $(\mathbf{w}, w_0)$ on the edge result in those K inequalities being satisfied as equalities, the Haar condition allows the final solution to be perturbed so that those K inequalities are satisfied strictly without violation of any previously satisfied inequality. In terms of the discriminant problem, the W-G/MultiODA method considers sets of K observations, selects discriminant functions that would classify those observations ambiguously ($f(\mathbf{x}) = 0$), and then perturbs the final discriminant function to correctly classify those K train-

ing observations without misclassifying any observation previously classified correctly.

Our oscillation approach will similarly search for suitable combinations of K training observations, and use them to generate discriminant functions. As with the W-G method, we require that the Haar condition hold. Among studies devoted to L_0-norm classification algorithms, some, such as Soltysik and Yarnold (1994), explicitly assume that the Haar condition applies, while others (Banks and Abad 1991; Duarte Silva and Stam 1997; Koehler and Erenguc 1990) implicitly assume the Haar condition by using weak inequalities and assuming that observations falling on the corresponding hyperplanes can be correctly classified. The only algorithms known to the authors that explicitly do not require the Haar condition are the decomposition algorithm proposed by Rubin (1997) and the PHASER heuristic of Stam and Ragsdale (1992). Hence assumption of the Haar condition, while a bit restrictive, is consistent with most prior work in the area.

2.2 Refinement of Solutions

Where Warmack and Gonzalez sought an algorithm that would be guaranteed to find an optimal solution, we seek a heuristic procedure, and so will not explore as many combinations of K points as W-G does. In exchange for the time saved by considering fewer combinations, we will exert a modest amount of effort with each combination to improve the classification error for the hyperplane corresponding to that combination. We use the "refinement" procedure of Yarnold *et al.* (Yarnold, Hart, and Soltysik 1994; Yarnold and Soltysik 1991), which does a one-dimensional brute force search for the constant term w_0 of the discriminant function that minimizes training misclassifications. It also incorporates the option of reversing the roles of G_1 and G_2, by changing $f()$ to $-f()$.

To refine the discriminant function $f()$, we begin by sorting and reindexing the observations into ascending order of $f(\mathbf{x})$, so that $f(\mathbf{x}_1) \leq \ldots \leq f(\mathbf{x}_n)$. We then examine sequentially every gap between two consecutive observations from different groups whose scores (in the sorted order) are unequal. For each such gap, we select a candidate cutoff value c within the gap (Yarnold *et al.* suggest the midpoint) and count the number of observations from each group with scores above and below that cutoff. If either $f() - c$ or $-f() - c$ misclassifies fewer observations than the best adjusted function found so far, that becomes the new incumbent classifier.

Obviously, there will be multiple optimal cutoff values, since any choice within the optimal gap is equivalent to any other choice in that gap in terms of accuracy on the training sample; moreover, there may be more than one gap

that is optimal for the training sample. Geometrically, refinement amounts to a parallel displacement of the separating hyperplane H. Note that the choice of a cutoff score between consecutive sorted scores of the original function implies that no training observation will receive a zero score from the refined function. The refinement procedure has relatively low computational cost (linear in n other than the sorting, whose effort is on the order of $n\log(n)$ for large n).

3. Oscillation Heuristics

In this section, we present the general framework for oscillation and describe some specific implementations.

3.1 Oscillation

Suppose that H is the unique hyperplane containing a particular subset $B = \{\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_K}\}$ of K training observations. The Haar condition both guarantees the uniqueness of H and precludes there being more than K training observations on H, so all observations falling on H belong to B. We refer to those observations as the *generators* of H. We will perform local searches, selecting a new hyperplane in a neighborhood of H. As with any local search method, our hope is that we can construct a chain of candidate hyperplanes, each in some sense near to its predecessor, and each improving on the classification accuracy of its predecessor.

Our notion of a neighborhood of H is the set of hyperplanes obtainable by replacing one of the generators $\mathbf{x}_{i_j} \in B$ with a nearby nongenerating observation $\mathbf{x}_k \notin B$. We term such a shift an *oscillation* of the original hyperplane. One plausible measure of proximity for nongenerating points is their Euclidean distance from H. Alternatively, we may elect to use training observations close in Euclidean distance to the generators they replace, rather than to H.

There is a precedent, in other nonparametric discriminant procedures, for this type of local search. Similar notions underlie kernel and nearest neighbor estimators (McLachlan 1992). Regarding neural networks, Ney (1995) notes: "In other words, the model parameters are estimated by giving emphasis to the training samples close to class boundaries, which is a well known heuristic principle in statistical pattern recognition." (Ney cites Fukunaga (1972, p. 106).)

Suppose that hyperplane H_0 results from refinement (parallel displacement) of hyperplane H. H_0 will not contain any training observations, so we must reconsider our idea of a neighborhood of H_0. We could revert to the generator set B of the prerefinement hyperplane H, thereby losing any benefit of refinement. Our preference, however, is to replace B with the K training ob-

servations whose Euclidean distance to H_0 is smallest; they, in turn, determine a new hyperplane H_1 near H_0, which we term the *postrefinement* hyperplane. When we speak of oscillating a refined hyperplane H_0, then, we will actually mean applying the oscillation process to the postrefinement hyperplane H_1.

3.2 Framework for Improvement Heuristics

Before discussing specific heuristics based on oscillation, we state a general framework for them, beginning with an initial hyperplane corresponding to a linear classifier generated by some other method.

Step 1: Initialize the heuristic. Select and refine an initial separating hyperplane. Decide how many oscillations (K_1) of the initial hyperplane to perform, and how many oscillations (K_2) of subsequent hyperplanes to perform.

Step 2: Construct a generator set for the initial refined hyperplane, select K_1 nearby nongenerator points, and examine every one of the $K \times K_1$ hyperplanes obtained by replacing one of the K generators with one of the K_1 nongenerators. (Note that the Haar condition guarantees that the hyperplanes are all distinct.) Also examine the refinements of each of those hyperplanes. Record separately the most accurate unrefined and refined hyperplanes, along with their generator sets. Several hyperplanes of the same type (refined or unrefined) may tie for most accurate, in which case all are recorded here and oscillated in step 3.

Step 3: Choose either the best unrefined or best refined hyperplanes and oscillate each of them by forming $K \times K_2$ new hyperplanes, replacing each generator in turn with one of the K_2 nearest nongenerator points. As each new hyperplane is encountered, compute its accuracy both before and after refinement, recording the best hyperplanes in each category (unrefined, refined) along with their generator sets. When oscillating a refined hyperplane, also check the accuracy of the postrefinement hyperplane.

Step 4: Based on some combination of recent improvement (or lack thereof) and number of iterations performed, decide whether to stop. If so, proceed to step 5. If not, repeat steps 3 and 4.

Step 5: If the best hyperplane encountered so far was unrefined (and hence contained K generators), perturb it slightly, in the manner described by Warmack and Gonzalez, so that the generators are classified correctly without losing accuracy at any other points. (If the best hyperplane was refined, it contains no sample points and so does not need perturbation.)

Besides the specific choice of K_1 and K_2, what will distinguish specific implementations of the oscillation framework are the decision rule used in step 3 to select either unrefined or refined hyperplanes for improvement, the criterion for selecting observations for substitution into the generator set (proximity to the hyperplane or to the generator set), and the stopping criterion.

3.3 Implementation of Details

In this section, we describe a few specific details of our implementation of the oscillation framework.

3.3.1 Number of Oscillations to Perform

In step 1 of the framework, we must decide how many times to oscillate the initial hyperplane (K_1), and how many times to oscillate subsequent hyperplanes (K_2). Along with the iteration limits used as stopping criteria (section 3.3.3 below), these parameters involve a tradeoff between accuracy of the final solution on the training sample and execution time. Larger oscillation counts and larger iteration limits will tend to produce more accurate solutions, at the cost of longer execution times. In our experiments, we set $K_1 = 3K$ and $K_2 = 2K$.

3.3.2 Choice of Hyperplanes to Oscillate

In step 3 of the framework, we must select either the best unrefined or best refined hyperplanes from the previous stage as targets for oscillation. Our general preference is to work with whichever hyperplanes give the lower error rate, with the proviso that if the error rate of the better set of hyperplanes has not improved since the last iteration, we will try the other set.

At first blush, it might seem that the best refined hyperplane will always produce no more classification errors than the best unrefined hyperplane, since the refinement process yields the lowest possible training error rate for a hyperplane with given normal vector. As it turns out, however, the best unrefined hyperplane can outperform the best refined hyperplane. Consider an unrefined hyperplane whose generator set B contains observations from both groups. Those observations are considered correctly classified by the unrefined hyperplane, since the Warmack-Gonzalez perturbation, which *alters the normal vector slightly*, will convert it to a hyperplane that classifies those K observations correctly without changing the classification of any other observation. The refinement procedure, however, is applied to the hyperplane before the Warmack-Gonzalez adjustment. Since all the points in B had equal discrimi-

nant scores (zero) before refinement, they all have equal (nonzero) discriminant scores after refinement, which means they will all be classified into the same group, and so some of them will be classified incorrectly. If refinement does not improve the accuracy at other training observations enough to compensate for this, the refined hyperplane will actually be less accurate than the unrefined hyperplane.

3.3.3 Stopping Criteria

We use a mixture of three stopping criteria. The first halts the heuristic if, at any iteration of step 3, it fails to locate a single hyperplane (refined or unrefined) at least as good as the current incumbent. The second criterion halts the heuristic if it matches but fails to improve on the incumbent error rate for a certain number of consecutive iterations. The third criterion for stopping is that a preselected iteration limit has been reached. We recommend that this limit permit at least $K - 1$ repetitions of step 3, so that, including step 2, at least K oscillations are attempted. Since K generators determine the initial hyperplane, this allows for the entire generator set to turn over if beneficial.

3.3.4 Criteria for Generator Replacement

As noted earlier, the replacements for current generators can be selected based on either of two criteria. In our experiments, we test three possible implementations:

Step 1: version OH, which selects nongenerator points nearest to the hyperplane being oscillated;

Step 2: version OP, which selects nongenerator points based on their proximity to the generator points they will replace; and

Step 3: version OHP, which combines the two criteria.

The method for selecting the k ($= K_1$ or K_2) nearest points to use in oscillating a hyperplane varies with the selection criterion. When the guideline is proximity to the hyperplane, the calculation hinges on the fact that the Euclidean distance of the point $\mathbf{y}$ from the hyperplane $H = \{x : f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - w_0 = 0\}$ is given by $|f(\mathbf{y})|/\|\mathbf{w}\|$ and is proportional to $|f(\mathbf{y})|$. Since we must sort the values $f(\mathbf{y})$ as part of the refinement process, we simply select the k nongenerating points whose function values are closest to zero in magnitude.

When the proximity criterion is nearness to the point being replaced, we proceed a bit differently. The list of k nearest points to a given observation $\mathbf{x}$

is not dependent on a particular hyperplane, although the list of $K-1$ generators (other than $\mathbf{x}$) to exclude from selection is. To facilitate the calculation of nearest points, we compute at the outset the distances between all pairs of observations, and store a vector of dimension n, whose i^{th} entry is a list of the indices of the $K-1+\max(K_1,K_2)$ training observations closest to $\mathbf{x}_i$, sorted according to distance from $\mathbf{x}_i$. Note that at worst $K-1$ of the observations closest to $\mathbf{x}_i$ will be generators of any hyperplane containing $\mathbf{x}_i$, so by recording the $K-1+\max(K_1,K_2)$ closest neighbors, we are assured of having at least $\max(K_1,K_2)$ nongenerators among them.

The hybrid OHP heuristic combines the nearness criteria of the OH and OP heuristics. To implement it, we apply the two oscillation procedures in succession, first oscillating the best previous hyperplane using the OH method, then oscillating the best resulting hyperplane from that step using OP. Other methods of combining the two heuristics remain to be investigated.

3.3.5 Redundancy

If $\mathbf{y}$ is one of the k observations nearest to $\mathbf{x}$, there is a good chance that $\mathbf{x}$ is one of the k nearest to $\mathbf{y}$. Similarly, if $\mathbf{x}$ is a generator of hyperplane H and $\mathbf{y}$ is near H, and if H' is the hyperplane obtained by replacing $\mathbf{x}$ with $\mathbf{y}$ in the generating set, then there is a good chance that $\mathbf{x}$ is among the nearest points to H'. Thus if oscillating H produces H', it is entirely possible that oscillating H' will produce H.

Therefore, there will be a tendency for hyperplanes to recur during the oscillation heuristic. Since we stop if there is no improvement within a certain number of iterations, indefinite cycling is not a danger. To avoid wasting effort, we can store a list of hyperplanes already tested (for instance, by storing their normalized coefficient vectors), and skip any hyperplanes already tested. We did so during our experiments, and in a small number of trials of the same data with and without screening of duplicates found that elimination of redundant hyperplanes did appear to compensate for the effort of testing for them. Screening out redundant hyperplanes raises the possibility that a given iteration generates no new hyperplanes and thus leaves the heuristic with no hyperplanes to oscillate; this condition becomes an additional criterion for stopping.

4. Experiments

We tested the oscillation heuristics on data generated using a Monte Carlo sampling procedure. To evaluate the heuristics, we applied them to two statistical procedures, two L_0-norm procedures and a support vector machine, examining their effect on accuracy in both the training sample and the overall

population, as well as computational effort. Each of those five procedures was applied five times to each test problem: alone; followed by the ODA refinement of the constant term (w_0); and followed by each of the oscillation heuristics (OH, OP, OHP). The OHP heuristic dominated ODA and the other two oscillation variants in training sample accuracy, at a modest cost in execution time. In the interest of brevity, only results for the method alone and augmented by OHP will be given below.

As a benchmark, we applied a mixed-integer programming method to most (but not all) of the problems. Discriminant functions determined by this procedure are guaranteed to have optimal accuracy on the training samples, and so neither the ODA refinement nor the oscillation heuristics were applicable in conjunction it.

4.1 Methods Tested

The two statistical procedures employed were the Fisher linear discriminant function (LDF) and the logistic discriminant function (LOG). Although LDF is one of the oldest known discriminant procedures, it continues to be studied and modified (Friedman 1989; Hastie, Tibshirani, and Buja 1994). It and LOG are the most widely used linear classifiers. LOG's performance, and its close affinity to the LDF, have been studied by several authors (McLachlan 1992, ch. 8). The general consensus (Krzanowski 1988) is that LOG is preferable to LDF when the distributions are clearly not Gaussian or the dispersion matrices are clearly unequal. However, LDF is simpler and requires less computational effort. LOG is also more sensitive to sample size and number of attributes; in particular, for fixed sample size its effectiveness decreases as the number of attributes increases.

The two L_0-norm procedures were a linear programming heuristic (Glover *et al.* 1988), denoted GKD, and the linear programming relaxation (MIPLP) of the MIP benchmarking model (listed below). The GKD heuristic constructs the classifier $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - w_0$ by solving the following linear program (which includes a slightly modified version of the normalization constraint posed in Glover (1990)):

$$
\begin{array}{lll}
\text{minimize} & h_0\alpha_0 + \sum_{i=1}^{n} h_i\alpha_i - k_0\beta_0 - \sum_{i=1}^{n} k_i\beta_i & \\
\text{s.t.} & \mathbf{x}_i \cdot \mathbf{w} - w_0 - \alpha_0 - \alpha_i + \beta_0 + \beta_i = 0 \quad (i = 1,\ldots,n_1)\ [G_1] & \\
& \mathbf{x}_i \cdot \mathbf{w} - w_0 + \alpha_0 + \alpha_i - \beta_0 - \beta_i = 0\ (i = n_1+1,\ldots,n)\ [G_2] & \\
& 2n_1n_2(\beta_0 - \alpha_0) + n_2\sum_{i=1}^{n_1}(\beta_i - \alpha_i) & \\
& \quad + n_1\sum_{i=n_1+1}^{n}(\beta_i - \alpha_i) = n_1n_2 & \\
& \alpha_i, \beta_i \geq 0 \qquad (i = 0,\ldots,n) & \\
& \mathbf{w}, w_0 \quad \text{free} &
\end{array}
\tag{2}
$$

The variables α_0 and β_0 denote respectively the smallest score violation of any incorrectly classified point and the smallest score "cushion" of any correctly classified point; the variables α_i and β_i denote respectively any additional violation or cushion for observation i. The corresponding objective coefficients h_0, k_0, h_i and k_i are all nonnegative, and must obey certain inequality relations for the linear program to be well-formed. (For instance, excessively rewarding cushion can lead to a solution that willingly misclassifies points in order to raise the cushion of other points.) Variables α_0 and β_0 should be deleted if their corresponding objective coefficients h_0 and k_0 are zero.

The support vector machine Vapnik (1999, p. 997), denoted SVM, can be expressed as a quadratic program:

$$\begin{array}{lllll} \text{minimize} & \mathbf{w}\cdot\mathbf{w}+\lambda\sum_{i=1}^{n} e_i & & & \\ \text{s.t.} & \mathbf{x}_i\cdot\mathbf{w}-w_0-e_i & \leq & -1 & (i=1,\ldots,n_1)\ [G_1] \\ & \mathbf{x}_i\cdot\mathbf{w}-w_0+e_i & \geq & 1 & (i=n_1+1,\ldots,n)\ [G_2] \\ & e_i & \geq & 0 & (i=1,\ldots,n) \\ & \mathbf{w}, w_0 & \text{free} & & \end{array} \tag{3}$$

where $\lambda > 0$ is a parameter to be chosen. The linear term of the objective can optionally be modified to incorporate prior probabilities and weights, as for instance $(\lambda\pi_1 c_1/n_1)\sum_{i=1}^{n_1} e_i + (\lambda\pi_2 c_2/n_2)\sum_{i=n_1+1}^{n} e_i$, where π_g, c_g and n_g are as defined for (4).

Benchmarking was done via a mixed-integer programming algorithm (Rubin 1997), denoted MIP:

$$\begin{array}{lllll} \text{minimize} & \frac{\pi_1 c_1}{n_1}\sum_{i=1}^{n_1} z_i + \frac{\pi_2 c_2}{n_2}\sum_{i=n_1+1}^{n} z_i - \varepsilon d & & & \\ \text{s.t.} & \mathbf{x}_i\cdot\mathbf{w}-w_0+d-M_i z_i & \leq & 0 & (i=1,\ldots,n_1)\ [G_1] \\ & \mathbf{x}_i\cdot\mathbf{w}-w_0-d+M_i z_i & \geq & 0 & (i=n_1+1,\ldots,n)\ [G_2] \\ & d & \geq & \delta & \\ & \mathbf{w}, w_0 & \text{free} & & \\ & z_i & \in\{0,1\} & & (i=1,\ldots,n) \end{array} \tag{4}$$

This model is a somewhat fuller version of (1). The misclassification costs c_i in (1) have been replaced by $\pi_g c_g/n_g$ $(g=1,2)$, where π_g is the prior probability of an observation belonging to group g, c_g is the cost of misclassifying an observation from group g, and n_g is the frequency of group g in the combined sample. Variable d represents the minimum score "cushion" of any correctly classified point, and is included in the objective with a small reward $\varepsilon > 0$ to encourage selection, among multiple solutions with equal misclassification costs, of one best separating the correctly classified points. Parameter $\delta > 0$ represents the minimum cushion for a correctly classified point; this has the effect of treating observations whose scores have the correct sign but are too close to zero

as being misclassified, reducing the likelihood of an ambiguously scored point being treated as correctly classified solely due to rounding errors. Finally, the parameters $M_i \gg 0$ are chosen to be sufficiently large that setting $z_i = 1$ renders the corresponding constraint vacuous. A formula for selecting the M_i is given in Rubin (1997). If run to completion, the MIP model is assured of producing a discriminant function with optimal accuracy on the training samples. The MIPLP heuristic is simply model (4) with the integrality restrictions on the z_i dropped.

4.2 Data Generation

The data sets used to evaluate the various procedures were generated by sampling multivariate normal distributions. Since the LDF and the Smith quadratic discriminant function theoretically provide optimal accuracy on multivariate normal populations with equal and unequal covariance matrices, respectively, L_0-norm methods are primarily studied for their efficacy on non-Gaussian data. That notwithstanding, experimental evaluations of heuristics typically begin with Gaussian data: since Gaussian or near-Gaussian data is common in practice, it would be undesirable to employ a method that did poorly with it. We fixed the number of attribute variables (K) at six, and assumed equal prior probabilities ($\pi_1 = \pi_2 = 0.5$) and misclassification costs ($c_1 = c_2 = 1$) for the two groups, so that the accuracy criterion can be expressed as the total number of misclassifications over the training sample. Our experimental design had four factors, each with two levels:

- The first factor was the sample size, either 150 (level 1, $n_1 = n_2 = 75$) or 300 (level 2, $n_1 = n_2 = 150$).
- The second factor was the average magnitude of correlation between each attribute and the subspace spanned by the preceding attributes. Its two levels were 0.2 (level 1) and 0.5 (level 2), representing relatively low and relatively high degrees of attribute correlation. Within each group in each replication, we generated a set of five correlations, independently sampled from a uniform distribution over $(-\rho - 0.1, -\rho + 0.1) \bigcup (\rho - 0.1, \rho + 0.1)$, where ρ was either 0.2 or 0.5. The first sampled value was the correlation between x_1 and x_2, the second was the correlation between x_3 and the linear subspace spanned by x_1 and x_2, and so on.
- The third factor governed the similarity between the covariance matrices of the two groups. Having computed the covariance matrix $\mathbf{S}_1$ for the first group, we set $\mathbf{S}_2 = \mathbf{D}\mathbf{S}_1\mathbf{D}$, where $\mathbf{D}$ was a diagonal matrix with diagonal entries D_{ii} independently generated such that $\log(D_{ii})$ was uniformly distributed over $(-\sigma, \sigma)$. The two levels for σ were 0.25 (level 1) and 1.25

(level 2), representing relatively similar and relatively dissimilar covariance structures between groups.

- The fourth factor dictated the separation between the groups. The quantiles of a multivariate normal distribution form K-dimensional ellipsoids. For some unique value $p \in (0,1)$, the ellipsoids $E_j = \{\mathbf{x} : pdf_j(\mathbf{x}) = p\}$ $(j = 1,2)$ will be tangent, where $pdf_j()$ is the probability density function for group G_j. Having established the covariance matrices of both groups, we arbitrarily placed the mean of the first group (G_1) at the origin, generated a direction $\mathbf{d}$ uniformly distributed over the unit sphere, located the point $\tilde{\mathbf{x}}$ on the p-quantile ellipsoid of G_1 along the ray in direction $\mathbf{d}$ emanating from the origin, and then set the mean of G_2 so that the p-quantile ellipsoid of G_2 would be tangent to the p-quantile ellipsoid of G_1 at $\tilde{\mathbf{x}}$. The levels for p were 0.1 (level 1, relatively great overlap) and 0.6 (level 2, relatively little overlap). Note that the greater the overlap, the less accurate classification will be.

Thus the cell coded 1212, for instance, used training samples of 75 observations from each group, with high correlation among the attributes, similar covariance matrices, and high separation. We generated 25 independent pairs of samples for each of the 16 experimental cells, for a total of 400 data sets.

4.3 Computational Considerations

All discriminant methods were coded in C++ and run on a personal computer powered by a 733 MHz Pentium III processor. The LDF and LOG functions were calculated directly from well known formulas, using the Matrix Template Library (Siek and Lumsdaine 1999) for matrix operations, and using the SolvOpt library (Kuntsevich and Kappel 1997) for maximum likelihood estimation of the logistic discriminant. OHP also used the Matrix Template Library. With the exception of LOG, methods involving optimization problems (GKD, MIP, SVM) employed the CPLEX 8.1 Callable Library (ILOG, Inc. 2002) to solve those problems. Data generation and evaluation of Bayes (population) error rates were done in Mathematica 5.0 (Wolfram Research Inc. 2003).

4.3.1 Parameter Selection

We performed pilot experiments on a few small samples to select parameters values for those methods with configurable parameters. Based on those experiments, we set parameters as follows:

- LDF and LOG require no user inputs.

- The SVM model (3) requires a single parameter λ. We initialized this at 1.0 and then performed, separately for each sample, a search for the parameter setting that minimized the objective function.
- For GKD, we set $h_0 = k_0 = 0$ (and deleted α_0 and β_0 accordingly), and set $\alpha_i = 1$ and $\beta_i = 0.5$ for $i = 1, \ldots, n$.
- For MIP, which decomposes the training samples into subsamples relegated to distinct subproblems, we specified the use of four subproblems when $n = 150$ and six subproblems when $n = 300$. We set the minimum separation parameter δ to 0.001.
- For OHP, we set the overall iteration limit to $2K$, with at most three iterations without improvement, and set $K_1 = 3K$ and $K_2 = 2K$.

4.3.2 MIP Solutions

While solution times for most of the methods grow modestly with sample size, solution time for the MIP algorithm increases dramatically as the sample size grows. For practical reasons, we were unable to solve all 400 test problems to optimality. Instead, we ordered the 400 data sets according to increasing number of misclassifications in the best solution found by one of the heuristics, and then solved the first 305 data sets in that sequence, which included all 200 of the small ($n_1 = n_2 = 75$) cases. Though not monotonic, MIP run time generally increased nonlinearly with heuristic error rate, so we feel we included most of the problems that MIP could have solved in a reasonable amount of time.

One other complication arose involving MIP. The raw data contained both small and large values, which created some rounding problems within the MIP code. To improve accuracy, we applied a scaling transformation to the data before applying the MIP algorithm to it. The scaling transformation amounts to multiplication of the data by a diagonal matrix with all positive diagonal entries.

4.4 Results

We preface the analysis of the results with reminders of several factors that influence the interpretation of those results:

- Accuracy results are quoted separately for the training samples and the overall populations. Since the actual group distributions were multivariable normal with known parameters, and since the discriminant functions were linear, we were able to compute population error rates exactly by integration, and did not have to rely on hold-out samples.

- Equal weight is given to misclassifications from either group, and the groups had equal prior probabilities. Error rates quoted below are averaged across the two groups.

- The problems for which we have MIP results tend to be those with the lowest optimal (training sample) error rates. It is possible that the performance of some methods, relative to that of MIP, would be different if we had MIP results for the harder test problems.

- Support vector machines are commonly implemented using kernel methods, which embed the original feature space (here $\Re^K$) in a higher dimensional space. Similarly, it is possible to generalize any of the other methods tested to certain types of nonlinear classifiers by expanding the feature space. (A distinction of SVMs is that they can work with an implicit embedding, whereas the other methods would require an explicit embedding.) The study here, however, is restricted to linear classifiers.

We must also stress that our purpose here is not to compare the various classification methods but to demonstrate that the OHP heuristic consistently improves their training sample accuracy, and to investigate conditions under which the OHP heuristic improves population accuracy.

4.4.1 Execution Time

We first dispense with the issue of execution time. We measured execution time (in seconds) from the start of algorithm execution to its conclusion, including neither the time to load the program and the problem data nor the time to display results. Background operating system activity inevitably makes the execution times somewhat random.

Four of the base heuristics (LDF, LOG, GKD and SVM) required less than two seconds to process each of our test cases. The fifth (MIPLP) required at most 2.25 seconds. Refinement of the classifier using OHP added an average of approximately 30 seconds, with a worst case increase of just over six minutes (when combined with GKD).

With the possible exception of certain "real-time" applications, we believe that the problem of classifier *construction*, as opposed to classifier *application*, is not terribly time-sensitive. Users will typically be willing to trade a reasonable amount of computing time for an improvement in accuracy. OHP execution times are clearly tied to problem dimensions (K, n), but are also tied to the stopping criteria and the number of oscillations performed (K_1, K_2). Thus the tradeoff between execution time and accuracy improvement is largely under the control of the user.

MIP execution times were considerably more variable, ranging from well under a second (in trivial cases, where both samples could be classified with perfect accuracy) to just over 62 hours for the hardest problem we solved. The difficulty of solving MIP classification models is well documented; less clear is whether they provide a commensurate improvement in accuracy.

4.4.2 Accuracy

The upper portions of tables 1 and 2 show the classification error rates achieved by each of the five basic procedures, both alone and supplemented by the oscillation heuristic ("OHP"), on the training samples for the small and large data sets respectively. Tables 3 and 4 show the population error rates for the classifiers obtained. The best accuracy in each treatment is printed in italics. Accuracy figures for each treatment are averaged across the 25 replications of that treatment. Since OHP is an improvement heuristic, its error rates on the training samples are never worse than those of the base methods; the only question is whether substantial improvement occurs. On the population, however, it is possible that OHP can actually reduce the classification accuracy of the function, and indeed this occurs in some cases.

Examination of these tables suggests several conclusions. The OHP heuristic improved classification accuracy on the training samples across the board, by generally substantial margins. For four of the methods, OHP yielded remarkably consistent error rate reductions: 14%–38% for LDF; 15%–35% for LOG; 13%–30% for GKD; and 12%–27% for SVM. (Reductions quoted are the change from the unenhanced error rate to the OHP-enhanced error rate, expressed as a percentage for the unenhanced error rate for that treatment.) Only in the case of MIPLP was the impact of OHP relatively minor (error rate reductions between 1.1% and 4.5% relative to MIPLP alone). On the other hand, the MIPLP+OHP combination produced the best average training sample results in 14 of 16 treatments.

OHP did not consistently improve population classification accuracy of any of the methods tested. Comparing population error rates in Tables 3 and 4 by treatment, we find relatively slight differences between the best and worst methods. In almost half the treatments, the best of the variants tested shaved not more than 3% off the error rate of the worst variant; in only two cases (treatment 1222 and 2222) did the best method reduce the population error rate of the worst method by more than 10%.

Treatments 2121, 2221 and 2222 provide encouraging results regarding the population accuracy of methods enhanced by OHP. These treatments all involve the larger ($n = 300$) training samples and populations with dissimilar covariance matrices; the former two are treatments with greater overlap among

Table 1. Training Error Rates ($n = 150$)

	Treatment							
Method	1111	1112	1121	1122	1211	1212	1221	1222
LDF	0.423	0.386	0.215	0.149	0.389	0.369	0.223	0.123
+OHP	0.330	0.298	0.154	0.101	0.303	0.282	0.157	**0.076**
LOG	0.422	0.385	0.219	0.146	0.390	0.369	0.225	0.121
+OHP	0.330	0.299	0.155	0.102	0.305	0.284	0.157	0.079
GKD	0.406	0.367	0.210	0.136	0.373	0.350	0.210	0.112
+OHP	0.330	0.301	0.153	0.103	0.304	0.285	**0.154**	0.079
MIPLP	0.337	0.298	0.156	0.103	0.314	0.284	0.159	0.079
+OHP	**0.324**	**0.294**	**0.152**	**0.100**	**0.300**	**0.281**	0.155	**0.076**
SVM	0.399	0.363	0.195	0.136	0.368	0.349	0.203	0.107
+OHP	0.329	0.296	**0.152**	0.101	0.302	0.285	0.157	0.078
Number	25	25	25	25	25	25	25	25
MIP	0.314	0.287	0.151	0.099	0.292	0.276	0.151	0.075
MIPLP	0.337	0.298	0.156	0.103	0.314	0.284	0.159	0.079
+OHP	0.324	0.294	0.152	0.100	0.300	0.281	0.155	0.076

Table 2. Training Error Rates ($n = 300$)

	Treatment							
Method	2111	2112	2121	2122	2211	2212	2221	2222
LDF	0.426	0.389	0.267	0.138	0.410	0.354	0.237	0.128
+OHP	0.362	0.333	0.214	0.104	0.348	0.302	**0.176**	**0.093**
LOG	0.426	0.390	0.266	0.135	0.411	0.354	0.235	0.125
+OHP	0.363	0.333	0.217	**0.103**	0.348	0.301	0.177	**0.093**
GKD	0.416	0.381	0.265	0.134	0.401	0.345	0.237	0.123
+OHP	0.360	0.330	0.217	0.105	0.347	0.299	0.177	0.094
MIPLP	0.364	0.334	0.219	0.105	0.353	0.306	0.183	0.183
+OHP	**0.358**	**0.329**	**0.212**	**0.103**	**0.342**	**0.298**	**0.176**	0.176
SVM	0.415	0.379	0.255	0.125	0.397	0.347	0.213	0.117
+OHP	0.363	0.333	0.214	**0.103**	0.348	0.302	**0.176**	**0.093**
Number	-	4	20	24	2	11	21	23
MIP	-	0.255	0.188	0.093	0.258	0.247	0.149	0.072
MIPLP	-	0.259	0.194	0.096	0.265	0.251	0.153	0.073
+OHP	-	0.258	0.190	0.094	0.265	0.249	0.152	0.072

Table 3. Population Error Rates ($n = 150$)

	Treatment							
Method	1111	1112	1121	1122	1211	1212	1221	1222
LDF	**0.470**	**0.438**	0.264	0.170	0.448	**0.414**	0.245	0.131
+OHP	0.477	0.443	0.261	0.179	0.451	0.429	0.248	0.139
LOG	**0.470**	**0.438**	0.265	**0.169**	0.448	**0.414**	0.246	0.131
+OHP	0.478	0.443	0.262	0.181	0.453	0.422	0.249	0.133
GKD	0.471	0.439	0.276	0.179	0.45	0.415	0.257	0.143
+OHP	0.474	0.444	0.263	0.181	0.449	0.427	0.246	0.134
MIPLP	0.473	0.447	0.263	0.181	0.456	0.433	0.245	0.135
+OHP	0.473	0.448	0.265	0.181	0.454	0.433	0.245	0.132
SVM	**0.470**	0.441	**0.257**	**0.169**	**0.444**	0.415	**0.241**	**0.127**
+OHP	0.476	0.449	0.259	0.183	0.449	0.426	0.246	0.131
Number	25	25	25	25	25	25	25	25
MIP	0.479	0.452	0.266	0.183	0.454	0.43	0.247	0.135
MIPLP	0.473	0.447	0.263	0.181	0.456	0.433	0.245	0.135
+OHP	0.473	0.448	0.265	0.181	0.454	0.433	0.245	0.132

Table 4. Population Error Rates ($n = 300$)

	Treatment							
Method	2111	2112	2121	2122	2211	2212	2221	2222
LDF	**0.461**	**0.424**	0.284	0.146	0.443	**0.376**	0.251	0.140
+OHP	0.465	0.428	**0.272**	0.150	0.445	0.383	0.242	**0.136**
LOG	**0.461**	**0.424**	0.286	0.144	0.443	**0.376**	0.251	0.138
+OHP	0.463	0.428	**0.272**	0.149	0.447	0.383	**0.238**	**0.136**
GKD	**0.461**	**0.424**	0.294	0.155	0.443	0.377	0.260	0.146
+OHP	0.462	0.427	0.275	0.148	0.448	0.384	0.240	0.137
MIPLP	0.467	0.430	0.279	0.148	0.452	0.387	0.241	0.183
+OHP	0.465	0.431	0.275	0.148	0.450	0.387	**0.238**	0.176
SVM	**0.461**	0.425	0.276	**0.141**	**0.442**	0.377	0.241	0.137
+OHP	0.463	0.430	0.273	0.149	0.448	0.384	**0.238**	0.137
Number	-	4	20	24	2	11	21	23
MIP	-	0.346	0.254	0.139	0.396	0.347	0.212	0.117
MIPLP	-	0.348	0.254	0.139	0.400	0.346	0.211	0.118
+OHP	-	0.348	0.254	0.139	0.400	0.346	0.211	0.118

the populations. As noted in the introduction, theoretical justification for improvement heuristics such as OHP can be drawn from consistency of $R(\mathbf{x};n)$ as an estimator of $R_{\min}$. Since consistency is an asymptotic property, it is reasonable to expect the benefit of OHP to manifest more in larger samples. Our data was normally distributed, and LDF is known to be theoretically optimal when the covariance matrices are equal, so it is not surprising that in treatments with similar covariance matrices the unenhanced LDF method dominated. Finally, OHP focuses exclusively on observations near the separating hyperplane, and larger samples from populations with greater overlap will be richer in such observations. In the three treatments 2121, 2221 and 2222, all of which involve large samples with dissimilar covariance matrices, OHP-enhanced methods provided the best population accuracy, and in fact OHP improved the accuracy of every method to which it was added. In the fourth treatment with large samples and dissimilar covariances, 2122, OHP did not fare as well. A possible explanation is that this treatment involved relatively great separation of the populations, which may tend to starve OHP of observations near the candidate hyperplanes.

The lower portion of each table compares the MIPLP with and without OHP to the results of the mixed-integer programming model (MIP) on those replications we were able to solve. MILP was chosen because MILP+OHP exhibited the best training sample accuracy among the heuristics tested. The first row in this section shows the number of replications (out of 25) for which we obtained MIP solutions. Results in the lower portions of the tables are averages across only those replications. We see that the MIPLP+OHP solutions have accuracy very similar to that of the MIP solutions in both the training samples and populations, obtained in a (frequently tiny) fraction of the computation time. MIPLP alone produces training sample error rates close to the optimum, which helps explain the rather modest gains of MIPLP+OHP over MIPLP.

5. Conclusions

We have proposed a new family of oscillation heuristics to improve linear classifiers, constructed by other methods, for the two-group classification problem. They are motivated by the intuition that small perturbations in the slope and position of the separating hyperplane can be used to improve its classification accuracy, and that those perturbations can be accomplished by substituting training observations near the hyperplane for those used to generate it. Experiments using data generated from multivariate normal distributions with a variety of intragroup and intergroup structures show the best of the oscillation heuristics, OHP, to consistently improve the training sample accuracy of the initial discriminant function, at modest computational cost. Initiated with the solution

to the linear relaxation of mixed-integer model (4), the OHP heuristic achieves near-optimal training sample accuracy at a fraction of the computational cost of the MIP model. Furthermore, there is evidence that the OHP heuristic improves expected accuracy of the discriminant functions on the underlying populations when applied to larger samples, particularly when population overlap is greater, at least in situations where data conditions do not explicitly favor a particular classifier (as the combination of normal distributions and similar covariances favors LDF here).

Multivariable Gaussian test data is a logical place to start the examination of the oscillation heuristics, but the necessary next step is to assess their efficacy on non-Gaussian data, and on outlier-contaminated data. Stam (1997) speculates that L_p-norm methods, and by extension heuristics such as OHP that focus on the geometry of the separating hyperplane, might be preferable to traditional parametric methods such as the Fisher LDF when the data comes from highly skewed distributions, when the underlying distributions are difficult to estimate (and by implication are substantially non-Gaussian), or when the samples are contaminated by outliers (to which the former methods are more robust). Data with discrete-valued (particularly binary) attributes would certainly challenge the distributional assumptions of parametric methods. Experiments with non-Gaussian data may better answer the question of whether pursuit of greater training sample accuracy results in greater population accuracy, or constitutes overfitting.

Another direction to investigate is the use of nonlinear classifiers. All the nonparametric methods tested can be extended to nonlinear classifiers, by embedding the original attribute space in a space of higher dimension either explicitly or, in the case of support vector machines, implicitly via kernel functions. (Parametric methods such as the Fisher LDF may admit this sort of embedding in a mechanical sense, but distributional assumptions will typically be stretched to the snapping point in the process.) Again, the issue will be whether the oscillation heuristics improve classification accuracy, and, if so, at what computational cost. Also, it is not immediately apparent what the Haar condition implies in that larger space.

Our approach to oscillation was based on selecting observations whose Euclidean distance from the separating hyperplane was small. An alternative would be to select observations whose *angular displacement* from the hyperplane was minimal. Selection of such points is computationally more intense than when distance is the criterion, which explains why we selected the approach we did. Nonetheless, it would be interesting to see if small angular perturbations produced better results than shifts to nearby observations. Another avenue to explore is adoption of a weighted combination of several of the best classifiers found, similar to what is done in AdaBoost (Freund and Schapire (1997)).

References

ABAD, P., and BANKS, W. (1993), "New LP-based Heuristics for the Classification Problem," *European Journal of Operational Research, 67*, 88–100.

AMALDI, E., and KANN, V. (1995), "The Complexity and Approximability of Finding Maximum Feasible Subsystems of Linear Relations," *Theoretical Computer Science, 147*, 181–210.

BANKS, W., and ABAD, P. (1991), "An Efficient Optimal Solution Algorithm for the Classification Problem," *Decision Sciences, 22*, 1008–1023.

CONWAY, D., CABOT, A., and VENKATARAMANAN, M. (1998), "A Genetic Algorithm for Discriminant Analysis," *Annals of Operations Research, 78*, 71–82.

DUARTE SILVA, A., and STAM, A. (1997), "A Mixed Integer Programming Algorithm for Minimizing the Training Sample Misclassification Cost in Two-Group Classification," *Annals of Operations Research, 74*, 129–157.

FREUND, Y., and SCHAPIRE, R. (1997), "A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting," *Journal of Computer and System Sciences, 55*, 119–139.

FRIEDMAN, J. (1989), "Regularized Discriminant Analysis," *Journal of the American Statistical Association, 84*, 165–175.

FUKUNAGA, K. (1972), *Introduction to Statistical Pattern Recognition*, New York: Academic Press.

GLOVER, F. (1990), "Improved Linear Programming Models for Discriminant Analysis," *Decision Sciences, 21*, 771–785.

GLOVER, F., KEENE, S., and DUEA, B. (1988), "A New Class of Models for the Discriminant Problem," *Decision Sciences, 19*, 269–280.

GOULJASHKI, V., and ASPAROUKHOV, O. (1999), "A Heuristic Procedure for a Two-Group Classification Problem," *Problems of Engineering Cybernetics and Robotics, 48*, 45–52.

HASTIE, T., TIBSHIRANI, R., and BUJA, A. (1994), "Penalized Discriminant Analysis," *Annals of Statistics, 23*, 73–102.

ILOG, INC. (2002), *CPLEX Callable Library*, Incline Village, NV: ILOG, Inc., version 8.1 ed.

KOEHLER, G., and ERENGUC, S. (1990), "Minimizing Misclassifications in Linear Discriminant Analysis," *Decision Sciences, 21*, 63–85.

KRZANOWSKI, W. (1988), *Principles of Multivariate Analysis*, New York: Oxford University Press.

KUNTSEVICH, A., and KAPPEL, F. (1997), "The solver for local nonlinear optimization problems (version 1.1)," Tech. rep., University of Graz.

MCLACHLAN, G. (1992), *Discriminant Analysis and Statistical Pattern Recognition*, New York: Wiley.

NEY, H. (1995), "On The Probabilistic Interpretation of Neural Network Classifiers and Discriminative Training Criteria," *IEEE Transactions on Pattern Analysis and Machine Intelligence, 17*, 107–119.

RUBIN, P. (1990), "Heuristic Solution Procedures for a Mixed-Integer Programming Discriminant Model," *Managerial and Decision Economics, 11*, 255–266.

RUBIN, P. (1997), "Solving Mixed Integer Classification Problems by Decomposition," *Annals of Operations Research, 74*, 51–64.

RUBIN, P. (1999), "Adapting the Warmack-Gonzalez Algorithm to Handle Discrete Data," *European Journal of Operational Research, 113*, 632–642.

SCHAPIRE, R. (1999), "A Brief Introduction to Boosting," in *Proceedings of the Sixteenth International Conference on Artificial Intelligence.*

SIEK, J., and LUMSDAINE, A. (1999), "The Matrix Template Library: Generic Components for High-Performance Scientific Computing," *Computing in Science & Engineering, 1*, 70–79.

SOLTYSIK, R., and YARNOLD, P. (1994), "The Warmack-Gonzalez Algorithm for Linear Two-Category Multivariable Optimal Discriminant Analysis," *Computers and Operations Research, 21*, 735–745.

STAM, A. (1997), "Nontraditional Approaches to Statistical Classification: Some Perspectives on Lp-Norm Methods," *Annals of Operations Research, 74*, 1–36.

STAM, A., and RAGSDALE, C. (1992), "On the Classification Gap in Mathematical-Programming-Based Approaches to the Discriminant Problem," *Naval Research Logistics, 39*, 545–559.

VAPNIK, V. (1999), "An Overview of Statistical Learning Theory," *IEEE Transactions on Neural Networks, 10*, 988–999.

VAPNIK, V., and CHERVONENKIS, A. (1971), "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities," *Theory of Probability and its Applications, 16*, 264–280.

WARMACK, R., and GONZALEZ, R. (1973), "An Algorithm for the Optimal Solution of Linear Inequalities and Its Application to Pattern Recognition," *IEEE Transactions on Computers, C22*, 1065–1075.

WOLFRAM RESEARCH INC. (2003), *Mathematica*, Champaign, IL: Wolfram Research, Inc., version 5.0 ed.

YARNOLD, P., HART, L., and SOLTYSIK, R. (1994), "Optimizing the Classification Performance of Logistic Regression and Fisher's Discriminant Analyses," *Educational and Psychological Measurement, 54*, 73–85.

YARNOLD, P., and SOLTYSIK, R. (1991), "Refining Two-Group Multivariable Classification Models using Univariate Optimal Discriminant Analysis," *Decision Sciences, 22*, 1158–1164.