Feature Selection for Multiclass Discrimination via Mixed-Integer Linear Programming

Frank J. Iannarilli Jr., *Member*, *IEEE*, and Paul A. Rubin

Abstract—We reformulate branch-and-bound feature selection employing L_∞ or particular L_p metrics, as mixed-integer linear programming (MILP) problems, affording convenience of widely available MILP solvers. These formulations offer direct influence over individual pairwise interclass margins, which is useful for feature selection in multiclass settings.

Index Terms—Feature selection, discrimination, classification, mixed-integer linear programming, branch-and-bound.

1 Introduction

1.1 Overview of Feature Selection

PATTERN recognition and classification problems occur across many application domains. A generic problem is to formulate a classifier $f(\vec{x}) \rightarrow \omega$ that assigns a class (category) label ω from the set of class labels $\{\omega_1, \ldots, \omega_Q\}$ to a vector-valued observation (measurement) $\vec{x}$, composed of features (scalar measurements) x_j. To be definite, let there be a total of Q class labels. Let there also be a maximum of N available features that may compose an observation vector $\vec{x}$. At an abstract level, $f(\vec{x})$ is a partitioning of an N-dimensional vector space (*feature space*) X, into a set of appropriately labeled hypervolume elements. The crux of pattern classification methodologies is to formulate or "learn" $f(\vec{x})$ from a combination of a priori domain knowledge, as well as a set $Y = \{\vec{x}_1, \ldots, \vec{x}_T\}$ of training observations.

Motivated by purposes of economy, and for deeper reasons related to the learnability of $f(\vec{x})$ from Y, **feature selection** (also known as "variable selection," "measurement selection," and "data selection") determines the "best" r out of N available features to compose reduced-dimensionality observations $\vec{x}$ for formulating $f(\vec{x})$ from Y. Feature selection continues to be a heavily studied area. For additional background on feature selection, see [1] and [2, Section 8.5]. A useful survey of feature selection methodologies is shown in [3]. Comprehensive empirical evaluations of various techniques are contained in [4], [5]. In addition to classical approaches that often appeal to normal distributions parameterized by low-order statistics, more recent "distribution-free" mathematical programming-based approaches have been developed to formulate classifiers. Some of these approaches have been augmented to perform feature selection. A useful overview is provided in [6].

1.2 Realm of Applicability

Feature selection is a combinatorial optimization problem, attempting to find $\vec{z}$, which maximizes some criterion function $J(\vec{z})$, where $\vec{z} \in \{0,1\}^N$ is a vector designating inclusion (1) or exclusion (0) of each feature. The pertinent quality measure of feature selection is the off-training-set Bayes error of the resulting classifier $f(\vec{x})$. Unfortunately, incorporating this measure within the feature selection optimization is typically computationally infeasible. Instead, to enable our technique, we impose two simplifications on the form of the class-conditional probability densities $p(\vec{x}|\omega_k)$.

Our first simplification is to model the densities $p(\vec{x}|\omega_k)$ as multivariate *unimodal* distributions (for example, Gaussians), with possibly distinct covariance matrices. (This modeling can be extended to *mixtures* of unimodal distributions by treating each mode of $p(\vec{x}|\omega_k)$ for the given class ω_k, as a pseudoclass.) Two favorable consequences ensue from this simplification. First, for the two-class case ($Q = 2$), the computationally-tractable Kullback-Leibler (K-L) divergence between $p(\vec{x}|\omega_m)$ and $p(\vec{x}|\omega_n)$ (which reduces to the Mahalanobis distance for equal covariance matrices) is a monotone function of the Bayes error [7] and, therefore, comprises a pertinent criterion $J(\vec{z})$. Second, for given $\vec{z}$, which specifies some chosen subset of r features, the K-L divergence between multivariate Gaussians $p(\vec{x}|\omega_m)$ and $p(\vec{x}|\omega_n)$ can only increase upon admitting any additional feature into the subset. In other words, the K-L criterion $J(\vec{z})$ is monotonic in $\vec{z}$: $J(\vec{z}) \leq J(\vec{z} + \vec{z}')$ (where $+$ is the inclusive-OR operator). This enables employment of implicit enumeration to optimize $J(\vec{z})$, finding the optimal combination of r features by (usually) evaluating dramatically fewer than the full set of $\binom{N}{r} = \frac{N!}{r!(N-r)!}$ combinations that would otherwise be required. (Note that, in general scenarios, because there is no a priori means of ascertaining the behavior of the Bayes error as a function of the chosen feature subset, the strictly valid means of optimal feature selection is to exhaustively evaluate the Bayes error over all $\binom{N}{r}$ possible feature subsets [8].) The classical method for such optimal feature selection via implicit enumeration is the branch-and-bound (B&B) method of Narendra and Fukunaga [9].

Our second simplification is to model the $p(\vec{x}|\omega_k)$ with *diagonal* covariance structure, i.e., where the principal axes of an underlying distribution in fact coincide with the coordinate measurement axes, or where we otherwise ignore (zero) the cross-moment terms. This consequently implies that the classifier $f(\vec{x})$ will employ the "naive Bayes" assumption, namely, the within-class independence of the individual features x_j in $\vec{x}$. This assumption allows the pairwise K-L divergence between $p(\vec{x}|\omega_m)$ and $p(\vec{x}|\omega_n)$ to be expressed as a linear combination of $\vec{z}$—$\sum_j c_j \cdot z_j$, with fixed (precomputed) c_j—and, thereby, to be compatible with our employment of mixed-integer **linear** programming (MILP) based solution, discussed later. This simplification is not as objectionable as it might appear: Often, there are not enough training samples available to estimate the parameters of the full (general) covariance with sufficient reliability (low variance) and, moreover, the "naive Bayes" assumption often works well in practice [10, Section 6].

1.3 Criteria for Multiclass Feature Selection

In the two-class problem, the Bayes error is a monotonically decreasing function of the K-L divergence d and, thus, the divergence is a suitable criterion to optimize. For instance, the two-class probability of error P_e in the case of equal covariance matrices and prior class probabilities is [11]:

$$P_e = g\left(\frac{\sqrt{d}}{2}\right) \quad g(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt. \tag{1}$$

Unlike the two-class case, *multiclass* criteria $J(\vec{z})$ that are both monotonic to the Bayes error and computationally tractable, remain elusive [12]. Instead, we can proceed to form a multiclass criterion that is an aggregate of the pairwise interclass divergences $d_{i,j}$, thereby retaining tractability and relationship to Bayes error.

- *F. Iannarilli Jr. is with Aerodyne Research, Inc., 45 Manning Rd., Billerica, MA 01821. E-mail: franki@aerodyne.com.*
- *P.A. Rubin is with the Department of Management, The Eli Broad Graduate School of Management, N442 North Business Complex, Michigan State University, East Lansing, MI 48824-1122. E-mail: rubin@msu.edu.*

Manuscript received 15 Jan. 2002; revised 18 Nov. 2002; accepted 22 Nov. 2002.
Recommended for acceptance by R. Beveridge.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 115709.

For equal prior class probabilities, the average probability of error P_e can be expressed as:

$$P_e = \frac{1}{Q}\sum_i \sum_{j \neq i} \int_{X_j} p(\vec{x}|\omega_i), \tag{2}$$

where X_j is the feature space decision region for class j. However, we and others [12] recognize that, since the presence of multiple classes reduces the decision region X_j relative to what it would be for the two-class case, the two-class error term $g\left(\frac{\sqrt{d_{i,j}}}{2}\right)$ sets an upper bound on the pairwise interclass contribution to the multiclass Bayes error P_e, as follows:

$$P_e \leq \frac{1}{Q}\sum_i \sum_{j \neq i} g\left(\frac{\sqrt{d_{i,j}}}{2}\right). \tag{3}$$

Equations (2) and (3) indicate two useful guidelines for an effective *multiclass* criterion $J(\vec{z})$. One is the employment of a saturating transform (akin to $g()$) on each pairwise divergence (or other measure of interclass margin), so the unbounded divergence $d_{i,j}$ behaves like a bounded Bayes error term. This provides rationale for the common use of various "separability index" measures crafted for multiclass discrimination, such as the transformed divergence, Jeffreys-Matusita (J-M) distance, etc., [12]. The use of saturating transforms is also consistent with the second guideline, that all pairwise margins be attended to, and possibly constrained by, lower bounds, so as not to allow a subset of these margins to dominate the optimization.

1.4 Motivation and Contribution of Proposed Methods

Motivated by the aforementioned desiderata for a multiclass feature selection criterion, we initially formulated (Section 2.1.1) an L_∞ metric-based integer nonlinear program (INLP) as a means to maximize the number as well as the magnitude of pairwise interclass margins. We use the term "metric" to denote a measure of the pairwise "distance" (margin) between classes based on the class-conditional distributions of the reduced feature vectors. This L_∞ metric employs a saturating transform on the pairwise margins, as motivated in the previous section (Section 1.3). Integer nonlinear programs can be tricky to solve. By linearizing the INLP model into a mixed-integer linear programming (MILP) formulation, we enable employment of efficient off-the-shelf MILP solvers. Our MILP formulation can easily be modified to substitute an L_p metric on the pairwise interclass margins for the L_∞ metric, so long as the metric is "diagonal" (i.e., excludes cross-moment terms). In accordance with our original motivations for multiclass feature selection (Section 1.3), this L_p metric version can be additionally augmented with optional specific lower bound constraints on *each* of the pairwise interclass margins.

Thus, the contribution of our proposed methods is the advantageous reformulation of B&B feature selection as an MILP problem, albeit for interclass distance measures complying with the "naive Bayes" assumption (class-conditional feature independence). Our criterion formulations for multiclass feature selection are not novel (see Section 1.2) and, in fact, are a restricted subset of those amenable to a branch-and-bound solution. Nevertheless, many practitioners may welcome their ready representation and solution using off-the-shelf MILP solvers. Moreover, advances in MILP solver technology readily accrue to the feature selection application.

The outline of our paper is as follows: In Section 2, we present our proposed methods of feature selection. In Section 3, we provide guidelines for practical application, including an example problem, discussion of computational complexity, and demonstration of application against some standard data sets.

2 Mixed-Integer Programming Feature Selection

2.1 The L_∞ Model

2.1.1 Initial Integer Nonlinear Programming (INLP) Formulation

Employing the L_∞ distance metric, we can cast our feature selection problem into the following form:

$$\max_{\vec{z}} J(\vec{z}) = \sum_{(m,n)\in C} \max_{j\in F}\left(a_j^{(m,n)} \cdot z_j\right) \text{ s.t.} \sum_{j\in F} z_j \leq r, \tag{4}$$

where $\vec{z} = \vec{z}(r)$ denotes the value of the binary indicator vector (defined earlier) that maximizes the criterion function given r, C is the set of all pairwise combinations (m, n) of classes, with cardinality $|C| = \binom{Q}{2}$, and $F = \{1, \ldots, N\}$ indexes the features.

The $a_j^{(m,n)}$ are component terms of a *diagonal* distance metric (one that ignores cross-moments between features) for pairwise combinations (m, n) of classes. For example, using a weighted L_1 metric applied to the class means, we might use

$$a_j^{(m,n)} = h\left(c \cdot \frac{\left|\mu_{m,j} - \mu_{n,j}\right|}{\frac{\sigma_{m,j}\sigma_{n,j}}{\sigma_{m,j}+\sigma_{n,j}}}\right), \tag{5}$$

with $\mu_{m,j}$ and $\sigma_{m,j}$ the training set (sample) conditional mean and standard deviation of the jth feature given class ω_m, $h()$ a bounded function, and c a positive scale parameter. The use of a saturating transform function $h()$, such as the sigmoidal function $\tanh()$, reduces the likelihood of selecting a set of features that produces a few large interclass distances at the expense of distinguishing between a number of other class pairs (see Section 1.3).

The $a_j^{(m,n)}$ component terms may alternatively be defined in accordance with an L_2 metric for the class means and dispersions, for instance, the K-L divergence, which, given our assumption of diagonal covariance, reduces to

$$a_j^{(m,n)} = h\left(\begin{array}{c} \frac{c}{2}\left(\mu_{m,j} - \mu_{n,j}\right)^2\left(\frac{1}{\sigma_{m,j}^2} + \frac{1}{\sigma_{n,j}^2}\right) \\ + \frac{c}{2}\left(\frac{\sigma_{n,j}^2}{\sigma_{m,j}^2} + \frac{\sigma_{m,j}^2}{\sigma_{n,j}^2} - 2\right)\end{array}\right). \tag{6}$$

For equal covariance matrices Σ_m and Σ_n, the K-L divergence reduces to the Mahalanobis distance. For unequal (distinct) covariance matrices, the K-L divergence consists of two terms: $\frac{1}{2}\left(\vec{\mu}_m - \vec{\mu}_n\right)^T\left(\Sigma_m^{-1} + \Sigma_n^{-1}\right)\left(\vec{\mu}_m - \vec{\mu}_n\right)$, an average Mahalanobis distance sensitive to the difference in means $\left(\vec{\mu}_m - \vec{\mu}_n\right)$, and $\frac{1}{2}\text{tr}\left(\Sigma_m^{-1}\Sigma_n + \Sigma_n^{-1}\Sigma_m - 2I\right)$, a term gauging the "shape disparity" of the distributions, which may be nonzero even for a zero difference in means.

Model (4) is an integer nonlinear program (INLP), with the nonlinearity due to the $\max$ operator in the objective summand. This operator is employed to preclude the selection of several columns j of large-valued $a_j^{(m,n)}$ that occur within a small number of rows of $A = \left(a_j^{(m,n)}\right)$, at the expense of neglecting the remaining rows (interclass pairs) that might possess a feature yielding satisfactory margin but whose selection would be otherwise suppressed.

2.1.2 Mixed-Integer Linear Programming (MILP) Reformulation

Unfortunately, INLP problems are notoriously difficult to solve, due to computational burden as well as convergence problems. Fortunately, we can recast this formulation into an equivalent mixed-integer **linear** program (MILP), which can be solved by generic MILP solvers, e.g., linear programming (LP)-based

branch-and-bound (B&B) solvers [13], [14]. Furthermore, since the B&B solver implicitly enumerates all $\binom{N}{r}$ possible values of $\overrightarrow{z}$, it is sure to find the optimal $\overrightarrow{z}$ (subject to time and memory constraints). We recast our initial INLP into an MILP as follows:

$$\max_{\overrightarrow{z},\overrightarrow{w}} J(\overrightarrow{z},\overrightarrow{w}) = \sum_{(m,n)\in C}\sum_{j\in F}\left(a_j^{(m,n)}\cdot w_j^{(m,n)}\right) \tag{7}$$

$$\text{s.t.}\sum_{j\in F} z_j \le r \tag{8}$$

$$\sum_{j\in F} w_j^{(m,n)} \le 1 \forall (m,n)\in C \tag{9}$$

$$w_j^{(m,n)} \le z_j \forall (m,n)\in C, j\in F, \tag{10}$$

where the z_j are the binary variables as before, and the $w_j^{(m,n)}$ are continuous, nonnegative auxillary variables. For a given interclass pair $(m,n)\in C$, the corresponding objective term is maximized by allocating all the available $(=1)$ weight across the N variables $w_j^{(m,n)}$ to the particular $w_j^{(m,n)}$, corresponding to the largest $a_j^{(m,n)}$ among the chosen features j. Note that this relies implicitly on our choice of nonnegative weights $a_j^{(m,n)}$ in (5) or (6).

Upon completion of the optimization, the maximum number of nonzero terms among the $w_j^{(m,n)}$ is limited by $q=\binom{Q}{2}$. Thus, in cases where $q<r$, this method will pick at most q features intelligently (and will either choose only q features, or will choose additional features purely by chance). Another drawback is that the model, by construction, may select only $r'<r$ features intelligently, if there is a strong ranking among features with respect to their corresponding "column sums" $\sum_{(m,n)\in C}\left(a_j^{(m,n)}\right)$. This can be counteracted by setting the constraint (8) to be an equality or, alternatively, employing our L_p model (with $\kappa=r$), which we discuss next.

2.2 The L_p Model

A simple augmentation, that we designate as our L_p model, overcomes the limitation of the L_∞ model, which is unable to (intelligently) choose more than $\min(q,r)$ features $(q=\binom{Q}{2})$. This is accomplished merely by modifying (9) as follows:

$$\sum_{j\in F} w_j^{(m,n)} \le \kappa \ \forall (m,n)\in C, \tag{11}$$

where the (row of) auxiliary variables summation bound of 1 is replaced by the integer $\kappa \le r$. The L_p model is able to intelligently choose $\min(q\cdot\kappa,\ r)$ features.

Now, for a given interclass pair $(m,n)\in C$, the corresponding objective term is maximized by allocating the available $(=\kappa)$ weight evenly among those κ variables $w_j^{(m,n)}$, corresponding to the κ largest $a_j^{(m,n)}$ among the N features j. Now that the available weight is distributed among κ features instead of just a single feature (for given interclass pair $(m,n)\in C$), it is appropriate to succinctly denote this model as an L_p model, in accordance with the elected definition of $a_j^{(m,n)}$ (see (5), (6), and Section 2.1.1). However, the admissible L_p metric is restricted to "diagonal" metric tensors.

2.3 The Constrained L_p Model

With the L_p model, as κ is chosen to be increasingly greater than 1, the advantage of the original L_∞ model in precluding a subset of $a_j^{(m,n)}$ from dominating the optimization is increasingly forfeited. Consequently, there is decreased advantage to retaining the individual rows (interclass pairs) (m,n) and their auxillary variables $w_j^{(m,n)}$, which form the criterion $J(\overrightarrow{z},\overrightarrow{w})$. In response to this, we offer an alternative formulation of the L_p model, which we denote as the "constrained L_p model," possessing three modifications. The first modification replaces the criterion $J(\overrightarrow{z},\overrightarrow{w})$ with the *average* interclass margin components. Maximization of such a criterion would be trivial (could be done by inspection), were it not for the second modification, which introduces lower-bound constraints on each of the pairwise interclass margins. These, in turn, make the use of the $\max$ operator in the objective unnecessary and, thus, make the auxiliary variables w superfluous. This model is formulated as follows:

$$\max_{\overrightarrow{z}}\ J(\overrightarrow{z}) = \sum_{j\in F}(\overline{a}_j\cdot z_j) \tag{12}$$

$$\text{s.t.}\sum_{j\in F} z_j \le r \tag{13}$$

$$\sum_{j\in F}\left(a_j^{(m,n)}\cdot z_j\right) \ge \lambda^{(m,n)}\ \forall (m,n)\in C, \tag{14}$$

where

$$\overline{a}_j = \frac{1}{q}\sum_{(m,n)\in C} a_j^{(m,n)}. \tag{15}$$

The added constraints force the solution to heed user-specified lower bounds $\lambda^{(m,n)}$ on the individual interclass margins. Although the feasible magnitudes of such bounds may require discovery through trial and error, an MILP solver can immediately report whether a model with the stated constraints is completely infeasible in its most relaxed (noninteger) form. The modifications reduce the size of the model and, thus, its computational burden. The lower-bound constraints retain and more directly enforce the desirable property for multiclass discrimination exhibited by our L_∞ model, namely, maximization of the number of occurrences, as well as magnitudes of acceptable pairwise interclass margins (see Sections 2.1.1 and 1.3).

3 Practical Application Guidelines

3.1 Example Application

We provide a concrete example in which we employ our multiclass feature selection methods to find the best two features. Fig. 1a is a representation of the feature space for a 10-class discrimination problem containing 32 features. The class-conditional mean values of the features ("feature value") are plotted as a function of the feature index, one plotted curve for each of the 10 classes, assuming a common feature standard deviation (s.d.) $\sigma=0.03$, and diagonal covariance structure (uncorrelated features). In applying our L_∞ method, we conditioned the interclass distances employing $\tanh()$ and using $c=0.1$ in (5). The $\tanh()$ function saturates at about three s.d. of margin. The corresponding author will provide the raw feature data to readers wishing to reproduce the results.

The best two features found by our L_∞ method for discriminating among the 10 classes are features #10 and #18. This is not surprising, given visual inspection of Fig. 1a. The resultant interclass separations are shown in Fig. 1b, as plotted within the coordinate subspace defined by features #10 and #18, with each plotted point representing a class. The classification error (numerically sampling Gaussian distributions as posed, to gauge the fraction misclassified, assuming equal a priori class probabilities) for a (quadratic) Bayes classifier employing these features is 0.31.

The best two features found by our L_p method, with $p=1$ and $\kappa=2$, are features #17 and #18. When compared to Fig. 1a, the resultant interclass separations shown in Fig. 1c reveal that the L_p method has maximized the *sum* of interclass margins at the expense of some individual margins. **Consequently, the Bayes error for employing these features is 0.75, substantially greater**.

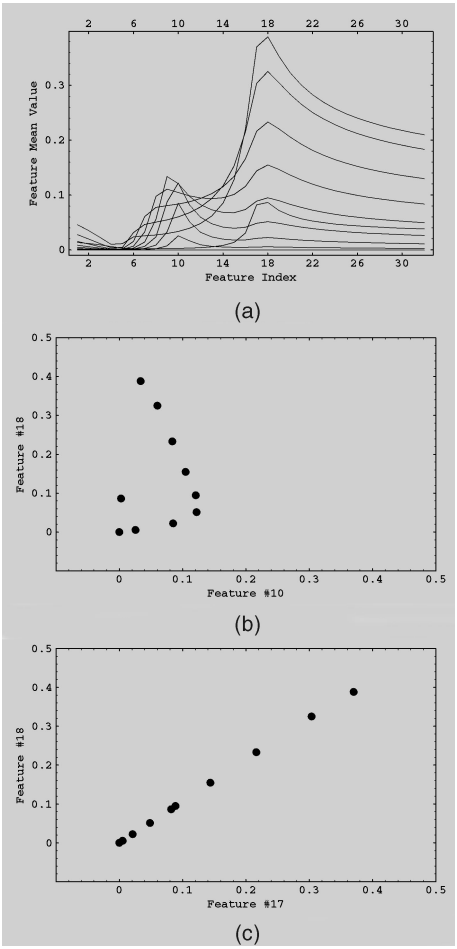

Fig. 1. (a) Feature values versus feature index plotted for 10 classes (one plot per class). (b) Plot of the value of feature index #18 versus index #10, for each of the 10 classes. These are the two best features for discriminating between the 10 classes as selected by out L_∞ and constrained L_1 MILP models. (c) Plot of the value of feature index #18 versus index #17, for each of the 10 classes. These are the two best features for discriminating between the 10 classes as selected by our L_1 MILP model.

Applying the constrained L_p method ($p = 1$ and $\kappa = 2$) with a common instance $\lambda^{(m,n)} = 0.14$ of constraint (14) asserted for all interclass distances, selects features #10 and #18, the same features selected by our L_∞ method. This is expected on the following basis. The minimum interclass mean disparity $|\mu_{m,j} - \mu_{n,j}|$ in Fig. 1b, is between 0.01 and 0.03 (by visual inspection). This corresponds to a transformed interclass distance component (given $\sigma = 0.03$, and using $\tanh()$ and $c = 0.1$ in (5)) between 0.05 and 0.2. Therefore, a value of the constraint parameter(s) $\lambda^{(m,n)}$ between 0.05 and 0.2 should force the constrained L_p method away from selecting features #17 and #18 and toward selecting features #10 and #18, which in fact occurs.

3.2 Computational Complexity

The parameters Q (number of classes) and N (number of features) determine the problem size and, together with r (maximum number of features desired), determine the solution time. The intrinsic MILP branch-and-bound solution timing is typically driven by the number of discrete (binary) variables, which for our formulations is N, rather than the number of continuous variables. Both the L_∞ and L_p models are sparse: either has $N(1+q)$ variables and $1+q+Nq$ constraints with $N(1+3q)$ nonzeros in the constraint matrix, where $q = \binom{Q}{2}$, so the constraint matrix density is

TABLE 1
Comparison of MILP and Baseline B&B Methods Against UCI Glass Data set

Method	$r = 3; (\epsilon)$	$r = 4; (\epsilon)$
L_∞ (L_1)	2,3,6; (0.22)	2,3,5,6; (0.15)
L_∞ (L_2)	3,6,8; (0.19)	3,6,7,8; (0.12)
L_1	3,4,8; (0.21)	2,3,4,8; (0.15)
L_2	3,6,8; (0.19)	3,6,7,8; (0.12)
constr. L_1	2,3,5 ($\lambda = 0.4$); (0.25)	1,2,3,5 ($\lambda = 0.45$); (0.15)
constr. L_2	3,7,8 ($\lambda = 0.44$); (0.22)	3,4,7,8 ($\lambda = 0.5$); (0.14)
Baseline B&B $\min L_2$	1,3,5; (0.25)	1,4,5,7; (0.17)
Baseline B&B L_2	3,6,7; (0.19)	1,3,6,7; (0.15)

$$\frac{N(1+3q)}{N(1+q)\cdot(1+q+Nq),} \tag{16}$$

which is asymptotically 0, as either $N \to \infty$ or $Q \to \infty$. Contemporary commercial MILP solvers can exploit this sparseness. In contrast, the constrained L_p model has N variables and, including all pairwise interclass distance constraints (14), has $1+q$ constraints with $N(1+q)$ nonzeros in the constraint matrix, giving it a density of 1. Commercial solvers can also exploit constraints (8) and (13), each of which defines a specially ordered set (SOS) [15], [16]. We note that there may be an opportunity to reduce the number of binary variables in the L_∞ model by preprocessing. If $a_j^{(m,n)} \le a_{j'}^{(m,n)}\ \forall (m,n) \in C$, feature j will never be selected in preference to j', and once feature j' is selected, constraint (9) will force $w_j^{(m,n)} = 0$.

The classical B&B approach [9] and our MILP formulations, both share the same number $\frac{N!}{r!(N-r)!}$ of potential solution (leaf) nodes in their B&B enumeration trees. The enumeration tree of [9] is balanced, as it starts with all N features, and proceeds by successively removing features to attain the best r-out-of-N feature subset. An MILP solver initially solves the linear relaxation LP, wherein the integer variables are allowed to be real-valued. Its B&B tree traversal successively constrains the integer variables to integrality. Thus, when employing "variable dichotomy" [15] branching decisions, the MILP enumeration tree for the binary decision variables $\vec{z}$ is unbalanced [13], which may or may not be advantageous. Tree balancing may be achieved by employing SOS branching, which is a capability provided by some MILP solvers. A notable attraction to [9] is an efficient means for incrementally updating the quadratic distance measure as features are successively added or deleted from the feature vector. A similar efficiency is intrinsically gained for our linearized criterion functions when using most MILP solvers, which employ rapid incremental reoptimization of the LP when traversing to a descendant node [13].

3.3 Empirical Confirmation

Table 1 presents results of trials against the standard "Glass" data set from the UCI repository [17], comparing the versions of our MILP method against a baseline B&B feature selection code within PRTools [18]. The PRTools B&B code is essentially a Narendra-Fukunaga method, although it recomputes from scratch the quadratic distances at each node rather than employing their efficient incremental update [9]. The "Glass" data set comprised five classes, using the nine numeric features (we excluded class label "6," as its nine samples inexplicably had three zero-valued

TABLE 2
Comparison of MILP and Baseline B&B Methods Against UCI Covtype Data Set

Method	$r = 2; (\epsilon)$	$r = 3; (\epsilon)$	$r = 4; (\epsilon)$
L_∞ (L_2)	1,5; (0.42)	1,5: 10; (0.39)	1,5: 9,10; (0.37)
L_2	1,6; (0.41)	1,6,10; (0.38)	1,4,6,10; (0.36)
Baseline B&B $L_2, \min L_2$	1,7; (0.39)	1,8,9; (0.37)	1,5,8,9; (0.35)

features). The table entries indicate the selected features and their corresponding Bayes error ϵ, as a function of method and subset size r. Assuming equal prior class probabilities, the Bayes error is gauged by numerically sampling the actual (full covariance) Gaussian distributions to gauge the fraction misclassified, for a (quadratic) Bayes classifier employing these features.

In the method notation, L_1 or L_2 indicate which form of the $a_j^{(m,n)}$ was employed (see (5) and (6); $c = 0.2$ in all cases). For the L_p (L_1 or L_2) model methods, we chose $\kappa = r$ (see Section 2.2). For the constrained L_p (L_1 or L_2) model methods, we indicate the lower-bound constraint value λ employed (common across all interclass pairs, see Section 2.3). Different features are nearly always selected for lower-bounds notably smaller than indicated. The baseline B&B method [18] employed a class-average covariance in computing as its criterion, either the sum of or the minimum of, the pairwise interclass (squared) Mahalanobis distances. The use of the minimum pairwise distance potentially helps in multiclass settings, as motivated in Section 1.3.

For this data set, our L_p and L_∞ MILP models, using L_2 distance definitions, yielded the best results because scatterplots indicate the actual data better matching our L_2 measure (which registers variance mismatch) than it does our L_1. Our constrained L_p MILP model yielded nearly as good results. All methods exhibited similar execution times.

We applied the same methodology, albeit for abbreviated comparisons shown in Table 2, for the UCI "Covtype" data set, which has seven classes (forest ground-cover-type). The 10 numeric features were employed (ignoring binary features), and we used the L_2 form of $a_j^{(m,n)}$ (see (6); $c = 0.8$). The quality of the L_p and L_∞ MILP model solutions remained competitive with the baseline B&B code. Our L_∞ MILP model chose only two features, even when three or four were allowed (desired). It chose the desired number r of features only when we set the constraint (8) to be an equality (see Section 2.1.2).

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their suggestions, which substantially improved our presentation. They also acknowledge use of data sets made available by the UCI Repository of machine learning databases maintained by the University of California Irvine Department of Information and Computer Science, and the use of the PRTools software, made available by the Pattern Recognition Group at the Delft University of Technology.

REFERENCES

[1] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification.* second ed. New York: John Wiley and Sons, Inc., 2001.
[2] C.M. Bishop, *Neural Networks for Pattern Recognition.* New York: Oxford Univ. Press, 1995.
[3] M. Dash and H. Liu, "Feature Selection for Classification," *Intelligent Data Analysis,* vol. 1, pp. 131-156, 1997.
[4] M. Kudo and J. Sklansky, "Comparison of Algorithms That Select Features for Pattern Classifiers," *Pattern Recognition,* vol. 33, pp. 25-41, 2000.
[5] A. Jain and D. Zongker, "Feature Selection: Evaluation, Application, and Small Sample Performance," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 2, pp. 153-158, Feb. 1997.
[6] J. Glen, "Classification Accuracy in Discriminant Analysis: A Mixed Integer Programming Approach," *J. Operational Research Soc.,* vol. 52, pp. 328-339, 2001.
[7] D.J. Hand, *Discrimination and Classification,* chapter 6. John Wiley and Sons, 1981.
[8] T.M. Cover and J.M.V. Campenhout, "On the Possible Orderings in the Measurement Selection Problem," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 7, no. 9, pp. 657-661, 1977.
[9] P. Narendra and K. Fukunaga "A Branch and Bound Algorithm for Feature Subset Selection," *IEEE Trans. Computers,* vol. 26, no. 9, pp. 917-922, Sept. 1977.
[10] T.M. Mitchell, *Machine Learning.* New York: McGraw-Hill, 1997.
[11] J.T. Tou and R.C. Gonzalez, *Pattern Recognition Principles.* London: Addison- Wesley, 1974.
[12] L. Bruzzone and S.B. Serpico, "A Technique for Feature Selection in Multiclass Problems," *Int'l J. Remote Sensing,* vol. 21, no. 3, pp. 549-563, 2000.
[13] L.A. Wolsey, *Integer Programming.* New York: John Wiley and Sons, Inc., 1998.
[14] R.J. Vanderbei, *Linear Programming,* second ed. Boston: Kluwer Academic Publishers, 2001.
[15] J. Mitchell and E. Lee, "Branch-and-Bound Methods for Integer Programming," *Encyclopedia of Optimization,* Kluwer Academic Publishers, 2001.
[16] J.T. Linderoth and M.W.P. Savelsbergh, "A Computational Study of Branch and Bound Search Strategies for Mixed Integer Programming," *INFORMS J. Computing,* vol. 11, pp. 173-187, 1999.
[17] C. Blake and C. Merz, "UCI Repository of Machine Learning Databases," http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.
[18] R. Duin, *PRTools 3, A Matlab Toolbox for Pattern Recognition.* Delft Univ. of Technology, 2000.