Feature Minimization within Decision Trees

Erin J. Bredensteiner

Kristin P. Bennett *

December 1996

Abstract

Decision trees for classification can be constructed using mathematical programming. Within decision tree algorithms, the feature minimization problem is to construct accurate decisions using as few features or attributes within each decision as possible. Feature minimization is an important aspect of data mining since it helps identify what attributes are important and helps produce accurate and interpretable decision trees. In feature minimization with bounded accuracy, we minimize the number of features using a given misclassification error tolerance. This problem can be formulated as a parametric bilinear program and is shown to be NP-complete. A parametric Frank-Wolfe method is used to solve the bilinear subproblems. The resulting minimization algorithm produces more compact, accurate, and interpretable trees. This procedure can be applied to many different error functions. Formulations and results for two error functions are given. One method, FM_RLP-P, dramatically reduced the number of features of one dataset from 147 to 2 while maintaining an 83.6% testing accuracy. Computational results compare favorably with the standard univariate decision tree method, C4.5, as well as with linear programming methods of tree construction.

Key Words: Data mining, machine learning, feature minimization, decision trees, bilinear programming.

*Knowledge Discovery and Data Mining Group, Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180. Email bredee@rpi.edu, bennek@rpi.edu. Telephone (518) 276-6899. FAX (518) 276-4824. This material is based on research supported by National Science Foundation Grant 949427.

1 Introduction

We consider the fundamental problem in machine learning of the discrimination between elements of two sets $\mathcal{A}$ and $\mathcal{B}$ in the n-dimensional real space R^n. Each dimension of the space represents a feature or attribute of the elements of the set. Commonly, the method of discrimination involves determining a linear function which consists of a linear combination of the attributes of the two given sets. In general it is not possible for a single linear function to completely separate these sets of points. Thus, some error criterion is minimized to determine the linear discriminant. To obtain a more accurate discrimination, many linear separators can be used as the decisions within a decision tree. In a decision tree, several linear discriminants are applied recursively to form a nonlinear separation of the space R^n into disjoint regions, each corresponding to set $\mathcal{A}$ or set $\mathcal{B}$. The goal is to obtain a decision tree, with one or more decisions, which generalizes well, i.e., correctly classifies future points.

Feature minimization is an important aspect of multivariate decision tree construction. The goal of feature minimization is to construct good decisions using as few features as possible. By minimizing the number of features used at each decision, understandability of the resulting tree is increased and the number of data evaluations is decreased [7]. Feature minimization is not necessary in univariate decision tree algorithms in which each decision in the tree is based on a single feature or attribute. Note that in this paper we use the term feature and attribute interchangeably. For example, in a credit card approval application a univariate decision may be: "Is income $>$ \$50,000?" A multivariate decision uses a linear combination of features, for example: "Is 3*debt $>$ income?" A tree with multivariate decisions can represent more complex relationships using fewer decisions than univariate

trees. However multivariate decisions with too many attributes can be difficult to interpret. Our goal is to make both a small number of decisions and to utilize only necessary attributes in each decision.

Feature minimization is especially important in data mining applications where the resulting decision tree is used not only to classify future points, but also to understand the underlying characteristics of the sets being studied. An added benefit is increased generalization at each node which may assist in better decision tree construction by avoiding overfitting. There is a trade off between the complexity of each decision and the number of decisions required in the tree. Multivariate decision trees typically have many fewer decisions than univariate decision trees constructed using one attribute per decision. Univariate decision trees have the advantages that single attribute decisions help avoid over-parameterization and the resulting trees are more readily interpretable provided the number of decisions is not excessive. Examples of univariate decision tree algorithms are C4.5 and CART [24, 6]. Reducing the number of features at each decision allows the inclusion of all of the benefits of multivariate decisions while maintaining the simplicity of univariate decisions.

The goal of this paper is to obtain a compact and accurate decision that includes as few attributes as possible while maintaining a specific level of accuracy. Our approach is to use mathematical programming to minimize the number of attributes used while maintaining a minimum level of accuracy as measured by some error metric. The resulting mathematical program constructs the discriminant and minimizes the number of features used simultaneously. Other mathematical programming approaches to feature minimization have been proposed in [23, 20]. The problem formulations and method of solutions are different than our

feature minimization methods. All of these recent mathematical programming approaches contrast with popular feature selection methods in both machine learning and statistics that do not explicitly consider the number of features used as part of the discrimination algorithm. Instead these approaches wrap an algorithm that adds and/or deletes attributes around an existing decision construction algorithm that minimizes some measure of the classification error. We propose directly changing the underlying discrimination algorithm. Some common approaches to feature minimization are based on greedy heuristics [16, 7]. Sequential Backward Elimination (SBE) and Sequential Forward Elimination (SFE) [7] involve searching the feature space for features that do not contribute (SBE) or contribute (SFE) to the quality of the decision. In SBE an initial discriminant function is constructed using all of the features and then features are removed sequentially from the problem until some stopping criterion is satisfied. In SFE, a discriminant is constructed using a single feature and then features are added one at a time. At each iteration in both methods, the best feature to add or remove is determined by finding the best discriminant for each possible attribute. Similar stepwise discrimination procedures (Forward Selection and Backward Elimination) are very popular in statistics [14]. They use statistical measures to determine which features to add or delete. The wrapper methods of Kohavi and John [15] provide a less greedy search of the feature space. But none of these approaches change the underlying discrimination algorithms.

In Section 2, we will discuss the background and formulation of our feature minimization method. Specifically we examine how feature minimization can be applied to two different discrimination methods. Both approaches require the solution of a parametric bilinear program. We then prove in Section 3 that our feature minimization problem is NP-complete.

In Section 4, we propose an algorithm based on the Frank-Wolfe method discussed in [5] for solving the parametric bilinear programming problem. Section 5 contains a computational comparison of our feature minimization method to C4.5 and two linear programming approaches to decision tree construction. Results on a number of practical problems are given.

The following notation is used. Let $\mathcal{A}$ and $\mathcal{B}$ be two sets of points in the n-dimensional real space R^n with cardinality m and k respectively. Let A be an $m \times n$ matrix whose rows are the points in $\mathcal{A}$. Let B be a $k \times n$ matrix whose rows are the points in $\mathcal{B}$. The i^{th} point in $\mathcal{A}$ and the i^{th} row of A are both denoted A_i. Likewise, B_j is the j^{th} point in $\mathcal{B}$ and the j^{th} row of B. For two vectors in R^n, xy denotes the dot product. Let e denote a vector of ones of the appropriate dimension. The set of minimizers of $f(x)$ on the set $\mathcal{S}$ is denoted by $arg\min_{x \in \mathcal{S}} f(x)$. For a vector x in R^n, x_+ will denote the vector in R^n with components $(x_+)_i := max\{x_i, 0\},\ i = 1, \ldots, n$. The step function x_* will denote the vector in $[0,1]^n$ with components $(x_*)_i := 0$ if $x_i \leq 0$ and $(x_*)_i := 1$ if $x_i > 0,\ i = 1, \ldots, n$.

2 Feature Minimization

At each decision we are interested in finding a linear function that separates the two sets. Mathematically, this corresponds to finding the plane

$$wx = \gamma \tag{1}$$

such that

$$Aw > e\gamma \qquad e\gamma > Bw \tag{2}$$

where $w \in R^n$ is the normal to the separating plane and γ determines the distance of the plane from the origin.

Upon normalization, this becomes

$$Aw - e\gamma - e \geq 0 \qquad -Bw + e\gamma - e \geq 0 \tag{3}$$

Equation (3) is feasible if and only if the two sets are linearly separable. In the event that the sets are not linearly separable, we must choose a plane that minimizes some error function.

Our formulations of the feature minimization problem can be applied to many different error functions. In this paper, we apply the feature minimization method to two error functions. The first error function minimizes the average magnitude of misclassified points within each class. The underlying problem without feature minimization is a linear program. This robust linear program (RLP) [4] has been used for decision tree construction [1]. RLP combined with the greedy sequential backward elimination method for feature minimization, a simplified version of SBE, forms the basis of a breast cancer diagnosis system [28, 27]. The second error function is a slight modification of the first. In addition to decreasing the average magnitude of misclassified points, it also decreases the maximum classification error. This problem can also be written as a linear program [3]. We will refer to it as the Perturbed Robust Linear Program (RLP-P). Our feature minimization method could also be applied

to algorithms that minimize the number of points misclassified such as [2, 18] or to other successful linear programming approaches [12, 25], but we leave these extensions for future work.

2.1 Feature Minimization Applied to RLP

The following robust linear programming problem, RLP [4], minimizes a weighted average of the sum of the distances from the misclassified points to the separating plane.

$$\begin{array}{ll} \min\limits_{w,\gamma,u,v} & \frac{1}{m}eu + \frac{1}{k}ev \\ subject\ to & u + Aw - e\gamma - e \geq 0 \\ & v - Bw + e\gamma - e \geq 0 \\ & u \geq 0, \quad v \geq 0 \end{array} \tag{4}$$

We are interested in minimizing the number of features at each decision in an effort to balance the amount of separation achieved versus the number of features used. The step function x_* will be used to count the number of nonzero elements in the vector w. We replace w with $(w_+) - (w_-)$ where $w_+, w_- \geq 0$. At optimality, $w_+ = (w)_+$ and $w_- = (-w)_+$. Thus the number of nonzero elements in the vector w is now $e((w_+) + (w_-))_*$. Adding this term to the objective function yields the following multiobjective optimization problem:

$$\begin{array}{ll} \min\limits_{w_+,w_-,\gamma,u,v} & \frac{1}{m}eu + \frac{1}{k}ev + \lambda e(w_+ + w_-)_* \\ subject\ to & u + A(w_+ - w_-) - e\gamma - e \geq 0 \\ & v - B(w_+ - w_-) + e\gamma - e \geq 0 \\ & u \geq 0, \quad v \geq 0, \quad w_+ \geq 0, \quad w_- \geq 0 \end{array} \tag{5}$$

where $\lambda > 0$ is constant.

The first issue we will confront in the above problem is the elimination of the step function. The step function is removed from problem (5) using properties found in [18] and [19]. The details are contained in the appendix. The resulting linear program (6) with equilibrium constraints [17] is equivalent to the original problem (5).

$$\begin{array}{ll} \min\limits_{w_+,w_-,\gamma,u,v,r} & \frac{1}{m}eu + \frac{1}{k}ev + \lambda er \\ \textit{subject to} & u + A(w_+ - w_-) - e\gamma - e \geq 0 \\ & v - B(w_+ - w_-) + e\gamma - e \geq 0 \\ & (w_+ + w_-)(e - r) = 0 \\ & 0 \leq r \leq e \\ & u \geq 0, \quad v \geq 0, \quad w_+ \geq 0, \quad w_- \geq 0 \end{array} \tag{6}$$

Note that at optimality $r = (w_+ + w_-)_*$, thus er counts the number of features used.

The second issue to confront is how to choose the parameter λ. The solution of problem (6) yields optimal decisions dependent on the value of λ. The choice of λ is not intuitively obvious. We propose two variants of the problem that eliminate the parameter λ, move the complementarity constraints to the objective function, and allow the problem to be solved using bilinear programming.

One possible approach is to minimize the number of features while satisfying a specific misclassification error bound. In our effort to achieve this goal, we propose removing λ from the problem by bounding the error function in a constraint. Problem (7) removes features while maintaining accuracy within some tolerance, δ. A similar concept was used by [7] and

[27] in their feature elimination methods. In [7], feature elimination was allowed to continue as long as a specific error tolerance was maintained. Street [27] computed planes for all feature counts and then used a tuning set to determine the best plane. We call this problem feature minimization with bounded accuracy and formulate it as follows.

Find the positive integer $\bar{\nu}$ such that

$$\bar{\nu} = \min_{\nu>0}\{\nu | f(\nu) = 0\} \tag{7}$$

where $f(\nu) =$

$$\begin{array}{ll} \min\limits_{w_+,w_-,\gamma,u,v,r} & (w_+ + w_-)(e - r) \\ \textit{subject to} & \frac{1}{m}eu + \frac{1}{k}ev \leq \delta \\ & u + A(w_+ - w_-) - e\gamma - e \geq 0 \\ & v - B(w_+ - w_-) + e\gamma - e \geq 0 \\ & 0 \leq r \leq e \quad er \leq \nu \quad \nu \in [1, n] \\ & u \geq 0, \quad v \geq 0, \quad w_+ \geq 0, \quad w_- \geq 0 \end{array} \tag{8}$$

For each fixed value of ν, problem (8) finds a linear separator within a specific error rate. If for any given ν, $f(\nu) \neq 0$, then no linear discriminant exists with the error tolerance that uses at most ν features. Theorem 3.2 proves this problem is NP-complete for the error function that counts the number of points misclassified.

An alternate approach is to rephrase the problem as follows: What is the best decision that can be made using at most ν variables? This limited feature minimization problem then becomes:

$$
\begin{aligned}
\min_{w_+,w_-,\gamma,u,v,r} \quad & \tfrac{1}{m}eu + \tfrac{1}{k}ev + \alpha(w_+ + w_-)(e - r) \\
\textit{subject to} \quad & u + A(w_+ - w_-) - e\gamma - e \geq 0 \\
& v - B(w_+ - w_-) + e\gamma - e \geq 0 \\
& 0 \leq r \leq e \quad er \leq \nu \quad \nu \in [1, n] \\
& u \geq 0, \quad v \geq 0, \quad w_+ \geq 0, \quad w_- \geq 0
\end{aligned}
\tag{9}
$$

Here $\alpha > 0$ must be chosen sufficiently large in order to force the complementarity constraints of (6), $(w_+ + w_-)(e - r) = 0$, to be satisfied at optimality.

Limiting the maximum number of features is appealing in practice because if the number of features is small the interpretability of the tree may be greatly enhanced.

For example, the number of features per decision could be limited to three, then each decision may be viewed graphically as a three dimensional plot. The extreme case of univariate trees, those limited to a single feature, has been demonstrated to work very well on numerous applications. However, such trees may require an excessive number of decisions.

2.2 Feature Minimization Applied to RLP-P

The following Perturbed Robust Linear Program (RLP-P) [3] is a linear programming modification of the Generalized Optimal Plane problem of Cortes and Vapnik [8]. This method is constructed to reduce a weighted average of the sum of the distances from the misclassified points to the separating plane and to decrease the classification error.

$$
\begin{array}{ll}
\min\limits_{w,\gamma,u,v,s} & (1-\epsilon)(\frac{1}{m}eu + \frac{1}{k}ev) + \epsilon es \\
\textit{subject to} & u + Aw - e\gamma - e \geq 0 \\
& v - Bw + e\gamma - e \geq 0 \\
& -s \leq w \leq s \\
& u \geq 0, \quad v \geq 0, \quad s \geq 0
\end{array}
\tag{10}
$$

where $0 < \epsilon < 1$ is a fixed constant. Note that at optimality $s = |w|$.

To minimize the number of features used in each decision, we proceed as before and substitute $w = w_+ - w_-$ and add the term $\lambda e(w_+ + w_-)_*$ to the objective function. The variable s can be removed from the problem since at optimality $s = |w| = |w_+ - w_-| = w_+ + w_-$.

$$
\begin{array}{ll}
\min\limits_{w_+,w_-,\gamma,u,v} & [(1-\epsilon)(\frac{1}{m}eu + \frac{1}{k}ev) + \epsilon e(w_+ + w_-)] + \lambda e(w_+ + w_-)_* \\
\textit{subject to} & u + A(w_+ - w_-) - e\gamma - e \geq 0 \\
& v - B(w_+ - w_-) + e\gamma - e \geq 0 \\
& u \geq 0, \quad v \geq 0, \quad w_- \geq 0, \quad w_+ \geq 0
\end{array}
\tag{11}
$$

where $0 < \epsilon < 1$ and $\lambda > 0$ are constant.

The step function is removed by using the same procedure as Section 2.1 and also contained in the appendix. The resulting linear program with equilibrium constraints follows:

$$
\begin{aligned}
\min_{w_+,w_-,\gamma,u,v,r} \quad & [(1-\epsilon)(\tfrac{1}{m}eu + \tfrac{1}{k}ev) + \epsilon e(w_+ + w_-)] + \lambda er \\
\textit{subject to} \quad & u + A(w_+ - w_-) - e\gamma - e \geq 0 \\
& v - B(w_+ - w_-) + e\gamma - e \geq 0 \\
& (w_+ + w_-)(e - r) = 0 \\
& 0 \leq r \leq e \\
& u \geq 0, \quad v \geq 0, \quad w_- \geq 0, \quad w_+ \geq 0
\end{aligned}
\tag{12}
$$

Similarly as Section 2.1, this problem can be solved in a variety of ways. In particular we propose solving the problem as a feature minimization with bounded accuracy problem.

Find the positive integer $\bar{\nu}$ such that

$$
\bar{\nu} = \min_{\nu>0}\{\nu | f(\nu) = 0\} \tag{13}
$$

where $f(\nu) =$

$$
\begin{aligned}
\min_{w_+,w_-,\gamma,u,v,r} \quad & (w_+ + w_-)(e - r) \\
\textit{subject to} \quad & (1-\epsilon)(\tfrac{1}{m}eu + \tfrac{1}{k}ev) + \epsilon e(w_+ + w_-) \leq \delta \\
& u + A(w_+ - w_-) - e\gamma - e \geq 0 \\
& v - B(w_+ - w_-) + e\gamma - e \geq 0 \\
& 0 \leq r \leq e \quad er \leq \nu \quad \nu \in [1, n] \\
& u \geq 0, \quad v \geq 0, \quad w_- \geq 0, \quad w_+ \geq 0
\end{aligned}
\tag{14}
$$

3 Computational Complexity

For this paper we will concentrate on the feature minimization with bounded accuracy problem. In this section, we will show that this problem is NP-complete. We begin by giving a formal definition of a problem titled "bounded accuracy with limited features". We then prove this problem to be NP-complete. The feature minimization with bounded accuracy problem is then defined and the bounded accuracy with limited features problem is used to prove it is NP-complete.

Definition 3.1 (Bounded Accuracy with Limited Features) *Let $\mathcal{X}$ be a finite subset of vectors in R^{n+1}. Let the vector $\bar{x} \in \mathcal{X}$ have integer valued entries. Is there a vector $\bar{y} \in R^{n+1}$ such that at most ν $(0 < \nu \leq n)$ entries $\bar{y}_i,\ i = 1, \ldots, n$, are nonzero and such that $\bar{x}\bar{y} > 0$ for at least K vectors $\bar{x}$?*

Specifically, $\mathcal{X}$ contains vectors of the form $[A_i, -1]$ and $[-B_i, 1]$. Also, $\bar{y}_i = (w_+ - w_-)_i$ for $i = 1, \ldots, n$ and $\bar{y}_{n+1} = \gamma$.

Theorem 3.1 *The Bounded Accuracy with Limited Features Problem is NP-complete.*

Proof. It is easy to show that this problem is in NP. We need only choose a vector $\bar{y} \in R^{n+1}$ and check in polynomial time whether $\bar{x}\bar{y} > 0$ for at least K vectors $\bar{x} \in \mathcal{X}$ and if at most ν elements $\bar{y}_i,\ i = 1, \ldots, n$, are nonzero.

To show that the above problem is NP-complete the Open Hemisphere problem of [11] can be easily transformed into a single instance of the bounded accuracy with limited features problem. The Open Hemisphere problem is the problem of determining if there is a vector $\bar{y}$

such that $\bar{x}\bar{y} > 0$ for at least K vectors $\bar{x} \in \mathcal{X}$. Thus, solving the Open Hemisphere problem is exactly solving the instance of our problem when $\nu =$ n. □

The feature minimization with bounded accuracy problem is precisely the problem of bounded accuracy with limited features with the added condition that the number of features be minimized. The formal definition of this problem is as follows.

Definition 3.2 (Feature Minimization with Bounded Accuracy) *Let $\mathcal{X}$ be a finite subset of vectors in R^{n+1}. Let the vector $\bar{x} \in \mathcal{X}$ have integer valued entries. Find a vector $\bar{y} \in R^{n+1}$ such that the number of nonzero elements $\bar{y}_i,\ i = 1, \ldots, n$, is* **minimized** *and such that $\bar{x}\bar{y} > 0$ for at least K vectors $\bar{x}$.*

Theorem 3.2 *The Feature Minimization with Bounded Accuracy Problem is NP-complete.*

Proof. We can show that this problem is in NP by relating it to at most two instances of the bounded accuracy with limited features problem which is in NP by Theorem 3.1. There exists a solution for the feature minimization with bounded accuracy problem with exactly p nonzero elements $\bar{y}_i,\ i = 1, \ldots, n$, if and only if there exists a solution for the bounded accuracy with limited features problem for $\nu = p$, but no solution exists for $\nu = p - 1$. Thus, given a vector $\bar{y}$ with p nonzero elements $\bar{y}_i,\ i = 1, \ldots, n$, we can check whether or not $\bar{y}$ is a solution to the feature minimization with bounded accuracy problem by solving the bounded accuracy with limited features problem for $\nu = p - 1$ and verifying that $\bar{y}$ satisfies $\bar{x}\bar{y} > 0$ for at least K vectors $\bar{x} \in \mathcal{X}$. Therefore, the feature minimization with bounded accuracy problem is in NP.

We will now show that this problem is NP-complete by reducing the bounded accuracy with limited features problem to the feature minimization with bounded accuracy problem. We solve the feature minimization with bounded accuracy problem and obtain the solution $\bar{y}$ which contains exactly p nonzero elements $\bar{y}_i,\ i = 1, \ldots, n$. A solution exists to the bounded accuracy with limited features problem if and only if $p \leq \nu$. □

In the next two sections, we describe a practical algorithm for solving the feature minimization with bounded accuracy problem and provide computational results.

4 Feature Minimization Algorithm

In this section we provide the algorithm used in solving our RLP feature minimization problem (7). All of the procedures discussed can be applied directly to the RLP-P feature minimization problem (13). The first step in solving problem (7) is to determine values for the parameters δ and ν. To determine δ, we solve the linear program (4) and allow for a 10% error on the value of the objective function. This value of δ will stay constant for the remainder of the solution of this program. The parameter ν is allowed to change in the process of solving this parametric bilinear program. Subsection 4.2 contains a complete description of how ν is chosen. For fixed values of δ and ν, several approaches are available to find a solution of program (8). Some possibilities are to apply branch and bound techniques, cutting plane methods, or the Frank-Wolfe method. The approach implemented in this paper uses a Frank-Wolfe type algorithm used successfully to solve bilinear programs in [5, 2]. This algorithm reduces the original bilinear program into two linear programs. One of these linear programs has a closed form solution as shown in [2]. A complete description of our algorithm

is given in the following two subsections.

4.1 Bilinear Subproblems

The parametric bilinear programming formulation (8) is an uncoupled bilinear program. It has been shown that a Frank-Wolfe algorithm [10] applied to an uncoupled bilinear program will converge to a global solution or a stationary point [5]. Applying this Frank-Wolfe algorithm to problem (8) we obtain the following algorithm:

Algorithm 4.1 (Frank-Wolfe algorithm for uncoupled bilinear programs) *For fixed* ν,

Step 1: $(w_+^{i+1}, w_-^{i+1}, \gamma^{i+1}, u^{i+1}, v^{i+1}) \in$

$$arg \min_{w_+, w_-, \gamma, u, v} \quad (w_+ + w_-)(e - r^i)$$

$$\tfrac{1}{m}eu + \tfrac{1}{k}ev \leq \delta$$

$$u + A(w_+ - w_-) - e\gamma - e \geq 0 \quad v - B(w_+ - w_-) + e\gamma - e \geq 0$$

$$u \geq 0 \quad v \geq 0 \qquad\qquad w_+ \geq 0 \quad w_- \geq 0$$

Step 2: $(r^{i+1}) \in$

$$arg \min_{r} \quad (w_+^{i+1} + w_-^{i+1})(e - r)$$

$$0 \leq r \leq e \qquad er \leq \nu$$

Step 3: Repeat until no improvement in objective.

It can easily be shown that the subproblem contained in step 2 has a closed form integer solution namely $r_j = 1$ for the ν largest components of $|w^{i+1}| = (w_+^{i+1} + w_-^{i+1})$ otherwise

$r_j = 0$.

4.2 The Feature Minimization Bilinear Program

The parametric bilinear program (7) searches for the minimum number of features such that a specific error criterion is met. There are various methods available for choosing which values of ν should be explicitly solved. For each ν a series of linear programs must be solved, thus it is computationally valuable to solve for as few values of ν as possible. We have used a modification of the secant method, similar to that used in [2], in the following algorithm:

Algorithm 4.2 (Feature Minimization with Bounded Accuracy) *Let ν_{max} denote the smallest number of features such that the error tolerance is* **satisfied** *thus far. Let ν_{min} denote the largest number of features attempted so far in Algorithm 4.1 such that the error tolerance is* **violated**. *All calculations for ν and p are rounded to the nearest integer.*

Step 0: Solve the robust LP (4) to find the best linear discriminant using all of the features.

Let lp error $= \frac{1}{m}eu + \frac{1}{k}ev$.

Let $\delta =$ *1.1(lp error).*

Let $\nu_{max} = n$. *Let* $\nu_{min} = 1$.

Step 1: Solve bilinear subproblem (8) using Algorithm 4.1 for $\nu = 1$.

If $f(\nu) = 0$ *then return* $\nu = 1$

else let $f(\nu_{min}) = f(\nu)$ *and* $\nu = \frac{1}{2}(n)$

Step 2: Solve bilinear subproblem (8) using Algorithm 4.1.

Step 3: If $f(\nu) = 0$

then let $\nu_{max} = \nu$ *and* $\nu = \frac{1}{2}(\nu_{min} + \nu_{max})$

else calculate secant method update

$$p = \nu - f(\nu)\frac{(\nu - \nu_{min})}{(f(\nu) - f(\nu_{min}))}$$

Let $\nu_{min} = \nu$ *and* $f(\nu_{min}) = f(\nu)$.

If $p \in (\nu_{min}, \nu_{max})$

then let $\nu = p$

else let $\nu = \frac{1}{2}(\nu_{min} + \nu_{max})$

Step 4: If $\nu_{max} > \nu_{min} + 1$ *Go to Step 2*

Else return ν_{max}.

To apply this algorithm to the RLP-P (13) simply substitute Problem (10) for Problem (4) and Problem (14) for Problem (8) in the above algorithm. In the remaining two sections, we refer to our implementation of the feature minimization with bounded accuracy program as Feature Minimization.

5 Computational Method

To evaluate the effectiveness of our mathematical programming methods, Feature Minimization applied to the RLP (FM_RLP) and Feature Minimization applied to the RLP-P (FM_RLP-P), we have implemented the RLP (4) and RLP-P (10) problems for comparison. C4.5, a popular univariate decision tree procedure, is used to form a baseline comparison. Several experiments on real world data sets are reported. Section 5.1 describes our exper-

imental method and the data sets used. Computational results on single linear separators are contained in Section 5.2.

5.1 Experimental Method

Feature Minimization results are compared to the linear programming results of RLP (4), RLP-P (10), and C4.5 as described below. Each linear programming method utilizes the CPLEX 3.0 [9] solver to optimize the linear subproblems. The simplex method was used to solve the initial linear programs and the dual simplex method solved the remaining subproblems. Different solvers could be used to achieve better results. In RLP-P and FM_RLP-P, we let $\epsilon = .02$. Better solutions may result with different choices for ϵ. To estimate generalization or accuracy on future data, 10-fold cross validation [26] was used to evaluate the testing set accuracies. The original data set is split into ten equal parts. Nine of these are used for training and the remaining one is saved for testing. This process is repeated ten times allowing each part to be the testing set. Feature Minimization requires that the training set be normalized. Thus, at each decision we normalize the training data and use the normalization information to transform (w, γ) for testing. The same training and testing sets were used for all methods.

Two experiments were performed. In both, results are given for the multivariate methods when obtaining a single linear separator. A greedy decision tree procedure could be applied to produce nonlinear separations. However in this paper we obtained excellent results using a single decision, so we only report results on this case. In the first experiment, we apply the mathematical programming methods to a Database Marketing data set. This data set

is divided into a training portion and a testing portion. Training and testing set accuracies are reported along with the number of features and decisions. In the second experiment, we compare generalization using the 10-fold cross validation procedure described previously. Note C4.5 produces a tree which consists of many univariate decisions. The other methods produce a single linear decision. For C4.5 the total number of decisions in the tree are reported. This is the same as the number of features used in the tree. We did not check for uniqueness. For the other methods, the total number of features used in the single decision is reported. For C4.5 all the default values were chosen, except windowing was disabled to allow comparisons with our methods.

The data sets used in the computational experiments are listed below. All of these data sets except the Star/Galaxy Database and the Database Marketing data set are available via anonymous file transfer protocol (ftp) from the University of California Irvine UCI Repository of Machine Learning Databases [21].

Database Marketing The Database Marketing data set is divided into a training portion and a testing portion. The training set contains information on 1979 customers. The testing set has 1491 customers. Each customer is classified as either young or old. 147 attributes are used to represent each customer.

Cleveland Heart Disease Database The Cleveland Heart Disease Database has 297 patients listed with 13 numeric attributes. Each patient is classified as to whether there is presence or absence of heart disease. There are 137 patients who have a presence of heart disease.

Wisconsin Breast Cancer Database This data set is used to classify 682 patients

with breast cancer. Each patient is represented by nine integral attributes ranging in value from 1 to 10. The two classes represented are benign and malignant: 442 of the patients are benign while 240 are malignant.

Star/Galaxy Database The Star/Galaxy Database consists of two data sets: dim and bright. The dim data set has 4192 examples and the bright data set has 2462 examples. Each example represents a star or a galaxy and is described by 14 numeric attributes. The bright data set is nearly linearly separable. These two data sets are generated from a large set of star and galaxy images collected by Odewahn [22] at the University of Minnesota.

Sonar, Mines vs. Rocks The Sonar data set [13] contains sixty real-valued attributes between 0.0 and 1.0 used to define 208 mines and rocks. Attributes are obtained by bouncing sonar signals off a metal cylinder (or rock) at various angles and rises in frequency. The value of the attribute represents the amount of energy within a particular frequency band, integrated over a certain period of time. This data set is linearly separable.

1984 United States Congressional Voting Records Database This data set includes votes for each of the 435 U.S. House of Representatives Congressmen. There are 267 democrats and 168 republicans. The chosen attributes represent 16 key votes. Possible values for the attributes are y, n, and ?. A value of ? indicates that the person did not make a position known. Our program requires numeric valued attributes, thus we let y, n, and ? be 2, -2, and 0 respectively.

	C4.5	RLP	FM_RLP	RLP-P	FM_RLP-P
Number of Features	62	133	23	41	2
Training Accuracy (%)	95.5	88.2	87.7	87.0	83.9
Testing Accuracy (%)	86.0	84.6	85.6	85.4	83.6

Table 1: Database Marketing Results

5.2 Computational Results

Table 1 lists results for the five methods when applied to the Database Marketing data set described above. The original data set contains 147 dimensions. The RLP method uses 133 of these features. The FM_RLP method reduces the number of features to 23 while slightly improving the testing set accuracy. The RLP-P method uses only 41 variables, yet this number is decreased significantly by the Feature Minimization method to only two features. This single linear separator with only two attributes has a testing accuracy comparable to those separators with more features. The Feature Minimization method was able to substantially improve the solutions of the two methods RLP and RLP-P. C4.5 produced the highest training accuracy, but achieved a testing accuracy similar to the FM_RLP method. FM_RLP used only 23 variables while the C4.5 tree contained 62 decisions. The Feature Minimization methods produced smaller decision trees while maintaining a similar level of accuracy as the other methods.

The results obtained by constructing a single decision for the multivariate methods RLP, FM_RLP, RLP-P, FM_RLP-P are contained in Table 2. The average number of variables used in each decision is reported. C4.5 results are reported for the same 10-fold cross validation

sets used on the multivariate methods. The average number of decisions or internal nodes on the univariate tree are given.

Table 2 shows that the Feature Minimization procedures, FM_RLP and FM_RLP-P, significantly reduced the average number of features used by the original RLP and RLP-P solutions. The testing set errors remained small. Thus, the Feature Minimization solutions are simpler and maintain accuracy. For most cases, the two Feature Minimization methods obtained much simpler decision trees than the univariate procedure C4.5. The testing set errors of FM_RLP and FM_RLP-P are generally smaller than those obtained by C4.5. For the House data set, FM_RLP-P reported errors comparable to the other methods, yet only one feature was used. Total computational times for the five methods are significant. The RLP and RLP-P methods used only 74.8 and 112.7 total seconds to complete the 10-fold cross validation procedure on all six data sets. The FM_RLP and FM_RLP-P methods used 1161.8 seconds and 1505.7 seconds respectively. The C4.5 method used a total of 679.8 seconds to complete its calculations. All problems were solved on a SUN Sparc10 workstation. The results for the Feature Minimization methods are an improvement over the RLP and RLP-P solutions. The accuracies of the solutions are similar to those of the RLP and RLP-P, yet the understandability has improved. Better solutions may be found by choosing different values for the parameters ϵ, δ, ν.

6 Conclusions

We have proposed two parametric bilinear programming methods for feature minimization. The first method, feature minimization with bounded accuracy, is the problem of finding

		C4.5	RLP	FM_RLP	RLP-P	FM_RLP-P
Heart	testing	27.3	16.5	16.8	16.8	19.5
	features	12.0	13.0	6.2	12.4	4.9
Cancer	testing	3.9	2.8	3.4	2.8	3.5
	features	6.0	9.0	5.6	8.7	4.9
Bright	testing	1.6	.6	.6	.7	1.0
	features	8.9	12.0	9.7	7.0	4.0
Dim	testing	5.9	4.5	5.1	5.0	6.0
	features	53.0	12.0	5.5	8.0	2.3
Sonar	testing	35.6	26.4	27.4	26.4	27.9
	features	15.2	55.5	29.7	38.8	18.1
House	testing	5.5	4.8	4.4	5.1	5.3
	features	3.4	15.8	9.3	11.9	1.0

Table 2: Testing set errors (%) and average number of features. Single decisions were formed for the multivariate methods: RLP, FM_RLP, RLP-P, and FM_RLP-P. C4.5 constructs a univariate decision tree.

a linear separator within a specific accuracy using as few features as possible. The second method, limited feature minimization, finds the best linear discriminant using at most ν features. The feature minimization with bounded accuracy problem was shown to be NP-complete. The feature minimization approach can be applied to many different error functions to produce accurate decision trees using a minimal number of features. We formulated bilinear programming problems for feature minimization with bounded accuracy with respect to two error functions. Feature minimization is an important aspect of data mining because we are interested in both the accuracy of the trees and the interpretability of the trees. A Frank-Wolfe algorithm was used to transform the bilinear program into a series of linear programs, half of which have closed form solutions. Computational results indicate that the Feature Minimization methods performed as accurately as the robust linear programming methods, RLP and RLP-P, and better than C4.5. Feature Minimization,

FM_RLP and FM_RLP-P, found planes with substantially fewer features than RLP and RLP-P, respectively. Therefore, Feature Minimization provides improvement over the linear programming methods with an additional computational time expense. As expected, the results are data set dependent and no single method always performs best. Overall Feature Minimization is a very promising approach. Further work is needed to explore the application of Feature Minimization to other types of discriminant functions and misclassification error metrics.

A Removal of the Step Function

The following equivalence relation will be used to transform the step function from program (5):

Proposition A.1 (Characterization of the Step Function) *[18, 19]*

$$r = (a)_* \;\; s = (a)_+ \iff (r,s) \in arg\min_{r,s} er \;\; subject\ to: \begin{array}{lll} r \geq 0 & s - a \geq 0 & r(s-a) = 0 \\ s \geq 0 & e - r \geq 0 & s(e-r) = 0 \end{array}$$

To apply this property to problem (5) we let r and s be as follows:

$$r = (w_+ + w_-)_* \tag{15}$$

$$s = (w_+ + w_-)_+ = (w_+ + w_-) = a \tag{16}$$

Resulting from above are the new constraints:

$$r \geq 0 \tag{17}$$

$$e - r \geq 0 \tag{18}$$

$$(w_+ + w_-)(e - r) = 0 \tag{19}$$

References

[1] K. P. Bennett. Decision tree construction via linear programming. In M. Evans, editor, *Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society Conference*, pages 97–101, Utica, Illinois, 1992.

[2] K. P. Bennett and E. J. Bredensteiner. A parametric optimization method for machine learning. R.P.I. Math Report No. 217, Rensselaer Polytechnic Institute, Troy, New York, 1995. To appear in *INFORMS Journal on Computing*.

[3] K. P. Bennett and E. J. Bredensteiner. Geometry in learning. In C. Gorini, E. Hart, W. Meyer, and T. Phillips, editors, *Geometry at Work*, Washington, D.C., 1997. Mathematical Association of America. To appear.

[4] K. P. Bennett and O. L. Mangasarian. Neural network training via linear programming. In P. M. Pardalos, editor, *Advances in Optimization and Parallel Computing*, pages 56–67, Amsterdam, 1992. North Holland.

[5] K. P. Bennett and O. L. Mangasarian. Bilinear separation of two sets in n-space. *Computational Optimization and Applications*, 2:207–227, 1993.

[6] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International, California, 1984.

[7] C. E. Brodley and P. E. Utgoff. Multivariate decision trees. *Machine Learning*, 19(1):45–77, 1995.

[8] C. Cortes and V. N. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[9] CPLEX Optimization Incorporated, Incline Village, Nevada. *Using the CPLEX Callable Library*, 1994.

[10] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.

[11] M.R. Garey and D.S. Johnson. *Computers and Intractability, A Guide to the Theory of NP–Completeness*. W.H. Freeman and Company, San Francisco, 1979.

[12] F. Glover. Improved linear programming models for discriminant analysis. *Decision Sciences*, 21:771–785, 1990.

[13] R.P. Gorman and T.J. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1:75–89, 1988.

[14] J. D. Jobson. *Applied Multivariate Data Analysis Volume II: Categorical and Multivariate Methods*. Springer Verlag, New York, 1992.

[15] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Journal of Artificial Intelligence*, 1996. to appear.

[16] R. Kohavi and D. Sommerfield. Feature subset selection using the wrapper method: Overfitting and dynamic search space topology. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, 1995.

[17] Z. Q. Luo, J. S. Pang, and D. Ralph. *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, Cambridge, England, 1996.

[18] O. L. Mangasarian. Misclassification minimization. *Journal of Global Optimization*, 5:309–332, 1994.

[19] O. L. Mangasarian. Mathematical programming in machine learning. Technical Report 95-06, University of Wisconsin, Madison, Wisconsin, 1995. To appear in *Proceedings of Nonlinear Optimization and Applications Workshop*, June 1995, Plenum Press.

[20] O. L. Mangasarian. Machine learning via polyhedral concave minimization. In S. Schaeffler H. Fischer, B. Riedmueller, editor, *Applied Mathematics and Parallel Computing - Festschrift for Klaus Ritter*, pages 175–188, Germany, 1996. Physica-Verlag. Technical Report 95-20, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, November 1995.

[21] P.M. Murphy and D.W. Aha. UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, California, 1992.

[22] S. Odewahn, E. Stockwell, R. Pennington, R Humphreys, and W Zumach. Automated star/galaxy discrimination with neural networks. *Astronomical Journal*, 103(1):318–331, 1992.

[23] O. L. Mangasarian P. S. Bradley and W. N. Street. Feature selection via mathematical programming. Technical Report 95-21, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, 1995. Submitted to *INFORMS Journal on Computing*.

[24] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[25] A. Roy, L. S. Kim, and S. Mukhopadhyay. A polynomial time algorithm for the construction and training of a class of multilayer perceptrons. *Neural Networks*, 6:535–545, 1993.

[26] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, 36:111–147, 1974.

[27] W.N. Street. Cancer diagnosis and prognosis via linear-programming-based machine learning. Technical Report 94-14, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, August 1994. Ph.D. thesis.

[28] W.H. Wolberg, W. N. Street, and O.L. Mangasarian. Image analysis and machine learning applied to breast cancer diagnosis and prognosis. *Quantitative Cytology and Histology*, 17(2):77–87, 1995.