Feature Selection via Concave Minimization and Support Vector Machines

P. S. Bradley
Computer Sciences Department
University of Wisconsin
Madison, WI 53706
paulb@cs.wisc.edu

O. L. Mangasarian
Computer Sciences Department
University of Wisconsin
Madison, WI 53706
olvi@cs.wisc.edu

Abstract

Computational comparison is made between two feature selection approaches for finding a separating plane that discriminates between two point sets in an n-dimensional feature space that utilizes as few of the n features (dimensions) as possible. In the concave minimization approach [19, 5] a separating plane is generated by minimizing a weighted sum of distances of misclassified points to two parallel planes that bound the sets and which determine the separating plane midway between them. Furthermore, the number of dimensions of the space used to determine the plane is minimized. In the support vector machine approach [27, 7, 1, 10, 24, 28], in addition to minimizing the weighted sum of distances of misclassified points to the bounding planes, we also *maximize* the distance between the two bounding planes that generate the separating plane. Computational results show that feature suppression is an indirect consequence of the support vector machine approach when an appropriate norm is used. Numerical tests on 6 public data sets show that classifiers trained by the concave minimization approach and those trained by a support vector machine have comparable 10-fold cross-validation correctness. However, in all data sets tested, the classifiers obtained by the concave minimization approach selected fewer problem features than those trained by a support vector machine.

1 INTRODUCTION

The feature selection problem addressed here is that of discriminating between two finite point sets in n-dimensional feature space R^n by a separating plane that utilizes as few of the features as possible.

Classification performance is determined by the inherent class information available in the features provided. It seems logical to conclude that a large number of features would provide more discriminating ability. But, with a finite training sample, a high-dimensional feature space is almost empty [12] and many separators may perform well on the training data, but few may generalize well. Hence the importance of the feature selection problem in classification [15]. The optimization formulations in Section 2 exploit one realization of the Occam's Razor bias [3]: compute a separating plane with a small number of predictive features, discarding irrelevant or redundant features. These formulations can be considered *wrapper models* as defined in [14].

The first approach [19, 5], described in Section 2, involves the minimization of a concave function on a polyhedral set. A plane is constructed such that a weighted sum of distances of misclassified points to the plane is minimized and as few dimensions of the original feature space R^n are used. This is achieved by constructing two parallel bounding planes, in as small dimensional space as possible, that bound each of the two sets to the extent possible by placing the two sets on two opposite halfspaces determined by the two planes. The two planes are determined such that the sum of weighted distances of points in the wrong halfspace to the bounding plane is minimized. This leads to the minimization of a concave function on a polyhedral set (problems (6) and (8) below) for which a stationary point can be obtained a successive linearization algorithm (Algorithm 2.1 below). The final separating plane is taken midway between the two bounding parallel planes.

The second approach, that of a support vector machine [27, 7, 1, 10, 24, 28], described in Section 3, constructs two parallel bounding planes in n-dimensional space R^n as in the first approach outlined above, but in addition attempts to push these planes as far apart as possible. The justification for this, apart from reducing the VC dimension [27] which in turn improves generalization, is that for the linearly separable case, the further apart the planes, the smaller the halfspace assigned to each of the two sets, reducing the possibility that new unseen points from the wrong set lie in that halfspace. Although improved generalization is the primary purpose of the support vector machine formulation, it turns out that the linear program (13) resulting from employing the ∞-norm to measure the distance between the two bounding planes, leads also to a feature selection method, whereas the linear program resulting from the use of the 1-norm (12) and the quadratic program resulting from the 2-norm (14) do not lead to feature selection methods.

In Section 4 we describe our computational experiments on 6 publicly available data sets using the approaches described in Sections 2 and 3. The goal is to evaluate the generalization ability of classifiers trained by solving: the concave optimization problem (8), three versions of the support vector machine problem with different norms (12), (13), (14) as well as the robust linear program RLP (4). RLP, which underlies the proposed feature selection methods here, has no feature suppression capability built in. We measure generalization ability by 10-fold cross-validation [26]. Numerical tests on 6 public data sets show that classifiers trained by the concave minimization approach and those trained by a support vector machine have comparable 10-fold cross-validation correctness. However, in all data sets tested, the classifiers obtained by the concave minimization approach selected fewer problem features than those trained by a support vector machine. Further, computational time for the normally used quadratic programming approach for SVMs, was orders of magnitude larger than the proposed linear programming approaches.

We now describe our notation and give some background material. All vectors will be column vectors unless transposed to a row vector by a superscript T. For a vector x in R^n, $|x|$ will denote a vector in R^n of absolute values of the components of x. For a vector $x \in R^n$, x_+ denotes the vector in R^n with components $\max\{0, x_i\}$. For a vector $x \in R^n$, x_* denotes the vector in R^n with components $(x_*)_i = 1$ if $x_i > 0$ and 0 otherwise (i.e. x_* is the result of applying the step function component-wise to x). The base of the natural logarithm will be denoted by ε, and for a vector $y \in R^m$, ε^{-y} will denote a vector in R^m with components ε^{-y_i}, $i = 1, \ldots, m$. For $x \in R^n$ and $1 \le p < \infty$:

$$\|x\|_p = \left(\sum_{j=1}^{n} |x_j|^p\right)^{\frac{1}{p}}, \ \|x\|_\infty = \max_{1 \le j \le n} |x_j|.$$

For a general norm $\|\cdot\|$ on R^n, the *dual norm* $\|\cdot\|'$ on R^n is defined as

$$\|x\|' = \max_{\|y\|=1} x'y.$$

The 1-norm and ∞-norm are dual norms, and so are a p-norm and a q-norm for which $1 \le p,\ q \le \infty$ and $\frac{1}{p} + \frac{1}{q} = 1$. The notation $A \in R^{m \times n}$ will signify a real $m \times n$ matrix. For such a matrix A^T will denote the transpose of A and A_i will denote the i-th row of A. A vector of ones in a real space of arbitrary dimension will be denoted by e. A vector of zeros in a real space of arbitrary dimension will be denoted by 0. The notation $\arg\min_{x \in S} f(x)$ will denote the set of minimizers of $f(x)$ on the set S. A separating plane, with respect to two given point sets $\mathcal{A}$ and $\mathcal{B}$ in R^n, is a plane that attempts to separate R^n into two halfspaces such that each open halfspace contains points mostly of $\mathcal{A}$ or $\mathcal{B}$.

2 FSV: FEATURE SELECTION VIA CONCAVE MINIMIZATION

In this part of the paper we describe a feature selection procedure that has been effective in medical and other applications [5, 19].

Given two point sets $\mathcal{A}$ and $\mathcal{B}$ in R^n represented by the matrices $A \in R^{m \times n}$ and $B \in R^{k \times n}$ respectively, we wish to discriminate between them by a separating plane:

$$P = \{x \mid x \in R^n, x^T w = \gamma\}, \tag{1}$$

with normal $w \in R^n$ and 1-norm distance to the origin of $\dfrac{|\gamma|}{\|w\|_\infty}$ [20]. We shall attempt to determine w and γ so that the separating plane P defines two open halfspaces $\{x \mid x \in R^n, x^T w > \gamma\}$ containing mostly points of $\mathcal{A}$, and $\{x \mid x \in R^n, x^T w < \gamma\}$ containing mostly points of $\mathcal{B}$. Hence, upon normalization, we wish to satisfy

$$Aw \geq e\gamma + e, \; Bw \leq e\gamma - e. \tag{2}$$

to the extent possible. Conditions (2) can be satisfied if and only if, the convex hulls of $\mathcal{A}$ and $\mathcal{B}$ are disjoint. This is not the case in many real-world applications. Hence, we attempt to satisfy (2) in some "best" sense by minimizing some norm of the average violations of (2) such as

$$\begin{aligned} \min_{w,\gamma} \; f(w,\gamma) = \min_{w,\gamma} \; & \frac{1}{m}\|(-Aw + e\gamma + e)_+\|_1 \\ & + \frac{1}{k}\|(Bw - e\gamma + e)_+\|_1. \end{aligned} \tag{3}$$

Recall that for a vector x, x_+ denotes the vector with components $\max\{0, x_i\}$. Two principal reasons for choosing the 1-norm in (3) are: (1) problem (3) is then reducible to a linear program (4) with many important theoretical properties making it an effective computational tool [2], (2) the 1-norm is less sensitive to outliers such as those occurring when the underlying data distributions have pronounced tails, hence (3) has a similar effect to that of robust regression [13],[11, pp 82-87].

The formulation (3) is equivalent to the following robust linear programming formulation (RLP) proposed in [2] and effectively used to solve problems from real-world domains [21]:

$$\begin{array}{ll} \underset{w,\gamma,y,z}{\text{minimize}} & \frac{e^T y}{m} + \frac{e^T z}{k} \\ \text{subject to} & \begin{array}{c} -Aw + e\gamma + e \leq y, \\ Bw - e\gamma + e \leq z, \\ y \geq 0, z \geq 0. \end{array} \end{array} \tag{4}$$

The linear program (4) or, equivalently, the formulation (3), define a separating plane P that approximately satisfies the conditions (2) in the following sense. Each positive value of y_i determines the distance $\frac{y_i}{\|w\|'}$ [20, Theorem 2.2] between a point A_i of $\mathcal{A}$ lying on the wrong side of the bounding plane $x^T w = \gamma + 1$ for $\mathcal{A}$, that is A_i lying in the open halfspace

$$\{x \mid x^T w < \gamma + 1\},$$

and the bounding plane $x^T w = \gamma + 1$. Similarly for $\mathcal{B}$ and $x^T w = \gamma - 1$. Thus the objective function of the linear program (4) minimizes the average sum of distances, weighted by $\|w\|'$, of misclassified points to the bounding planes. The separating plane P (1) is midway between the two bounding planes and parallel to them.

Feature selection [19, 5] is imposed by attempting to suppress as many components of the normal vector w to the separating plane P that is consistent with obtaining an acceptable separation between the sets $\mathcal{A}$ and $\mathcal{B}$. We achieve this by introducing an extra term with parameter $\lambda \in [0, 1)$ into the objective of (4) while weighting the original objective by $(1 - \lambda)$ as follows:

$$\begin{array}{ll} \underset{w,\gamma,y,z}{\text{minimize}} & (1-\lambda)\left(\frac{e^T y}{m} + \frac{e^T z}{k}\right) + \lambda e^T |w|_* \\ \text{subject to} & \begin{array}{c} -Aw + e\gamma + e \leq y, \\ Bw - e\gamma + e \leq z, \\ y \geq 0, z \geq 0. \end{array} \end{array} \tag{5}$$

Note that the vector $|w|_* \in R^n$ has components which are equal to 1 if the corresponding components of w are nonzero and components equal to zero if the corresponding components of w are zero. Recall that e is a vector of ones and $e^T|w|_*$ is simply a count of the nonzero elements in the vector w. Problem (5) balances the error in separating the sets $\mathcal{A}$ and $\mathcal{B}$, $\left(\frac{e^T y}{m} + \frac{e^T z}{k}\right)$, and the number of nonzero elements of w, $(e^T|w|_*)$. Further, if an element of w is zero, the corresponding feature is removed from the problem.

By introducing the variable v we are able to eliminate the absolute value from problem (5) which leads to the following equivalent parametric program (for $\lambda \in [0, 1)$):

$$\begin{array}{ll} \underset{w,\gamma,y,z,v}{\text{minimize}} & (1-\lambda)\left(\frac{e^T y}{m} + \frac{e^T z}{k}\right) + \lambda e^T v_* \\ \text{subject to} & \begin{array}{c} -Aw + e\gamma + e \leq y, \\ Bw - e\gamma + e \leq z, \\ y \geq 0, z \geq 0, \\ -v \leq w \leq v. \end{array} \end{array} \tag{6}$$

Since v appears positively weighted in the objective and is constrained by $-v \leq w \leq v$, it effectively models the vector $|w|$. This feature selection problem will be solved for a value of $\lambda \in [0, 1)$ for which the resulting classification obtained by the separating plane (1) midway between the bounding planes $x^T w = \gamma \pm 1$, generalizes best, estimated by a cross-validation tuning procedure. Typically this will be achieved in a feature space of reduced dimensionality, that is $e^T v_* < n$ (i.e. the number of features used is less than n).

Because of the discontinuity of the step function term $e^T v_*$, we approximate it by a concave exponential on the nonnegative real line [19]. The approximation of the step vector v_* of (6) by the concave exponential :

$$v_* \approx t(v, \alpha) = e - \varepsilon^{-\alpha v}, \alpha > 0, \tag{7}$$

leads to the smooth problem (**FSV**:Feature Selection Concave):

$$\begin{array}{ll} \underset{w,\gamma,y,z,v}{\text{minimize}} & (1-\lambda)\left(\frac{e^T y}{m} + \frac{e^T z}{k}\right) + \lambda e^T(e - \varepsilon^{-\alpha v}) \\ \text{subject to} & \begin{array}{c} -Aw + e\gamma + e \le y, \\ Bw - e\gamma + e \le z, \\ y \ge 0, z \ge 0, \\ -v \le w \le v. \end{array} \end{array} \tag{8}$$

It can be shown [4, Theorem 2.1] that for a finite value of α (appearing in the concave exponential) the smooth problem (8) generates an exact solution of the nonsmooth problem (6). We note that this problem is the minimization of a concave objective function over a polyhedral set. Even though it is difficult to find a global solution to this problem, a fast successive linear approximation (SLA) algorithm [5, Algorithm 2.1] terminates finitely (usually in 5 to 7 steps) at a stationary point which satisfies the minimum principle necessary optimality condition for problem (8) [5, Theorem 2.2] and leads to a sparse w with good generalization properties. For convenience we state the SLA algorithm below.

Algorithm 2.1
Successive Linearization Algorithm (SLA) for FSV (8). *Choose $\lambda \in [0, 1)$. Start with a random $(w^0, \gamma^0, y^0, z^0, v^0)$. Having $(w^i, \gamma^i, y^i, z^i, v^i)$ determine $(w^{i+1}, \gamma^{i+1}, y^{i+1}, z^{i+1}, v^{i+1})$ by solving the linear program:*

$$\begin{array}{ll} \underset{w,\gamma,y,z,v}{\text{minimize}} & (1-\lambda)(\frac{e^T y}{m} + \frac{e^T z}{k}) + \lambda\alpha\left(\varepsilon^{-\alpha v^i}\right)^T (v - v^i) \\ \text{subject to} & \begin{array}{c} -Aw + e\gamma + e \le y, \\ Bw - e\gamma + e \le z, \\ y \ge 0, z \ge 0, \\ -v \le w \le v. \end{array} \end{array} \tag{9}$$

Stop when

$$(1-\lambda)(\frac{e^T(y^{i+1} - y^i)}{m} + \frac{e^T(z^{i+1} - z^i)}{k}) + \lambda\alpha\left(\varepsilon^{-\alpha v^i}\right)^T (v^{i+1} - v^i) = 0. \tag{10}$$

Comment: The parameter α was set to 5. The parameter λ was chosen to "maximize" generalization performance.

We have found useful solutions to (8) for the fixed value $\alpha = 5$ [5, 4]. Another approach, involving more computation, is to solve (8) for an increasing sequence of α values.

3 SVM: FEATURE SELECTION VIA SUPPORT VECTOR MACHINES

The support vector machine idea [27, 1, 10, 24, 28], although not originally intended as a feature selection tool, does in fact indirectly suppress components of the normal vector w to the separating plane P (1) when an appropriate norm is used for measuring the distance between the two parallel bounding planes for the sets being separated. The SVM approach consists of adding another term, $\frac{\|w\|'}{2}$, to the objective function of the RLP (4) in a similar manner to the appended term $e^T|w|_*$ of problem (5). Here, $\|\cdot\|'$ is the dual of some norm on R^n used to measure the distance between the two bounding planes. The justification for this term is as follows. The separating plane P (1) generated by the RLP linear program (4) lies midway between the two parallel planes $w^T x = \gamma + 1$ and $w^T x = \gamma - 1$. The distance, measured by some norm $\|\cdot\|$ on R^n, between these planes is precisely $\frac{2}{\|w\|'}$ [20, Theorem 2.2]. The appended term to the objective function of the RLP (4), $\frac{\|w\|'}{2}$, is the reciprocal of this distance, thus driving the distance between these two planes up to obtain better separation. This results then in the following mathematical programming formulation for the SVM formulation:

$$\begin{array}{ll} \underset{w,\gamma,y,z,\nu}{\text{minimize}} & (1-\lambda)(e^T y + e^T z) + \frac{\lambda}{2}\|w\|' \\ \text{subject to} & \begin{array}{c} -Aw + e\gamma + e \le y, \\ Bw - e\gamma + e \le z, \\ y \ge 0, z \ge 0. \end{array} \end{array} \tag{11}$$

Points $A_i \in \mathcal{A}$ and $B_i \in \mathcal{B}$ appearing in active constraints of the linear program (11) with positive dual variables constitute the *support vectors* of the problem. These points are the *only* data points that are relevant for determining the optimal separating plane. Their number is usually small and it is proportional to the generalization error of the classifier [24].

If we use the 1-norm to measure the distance between the planes, then the dual to this norm is the ∞-norm and accordingly $\|w\|' = \|w\|_\infty$ in (11) which leads to the following linear programming formulation:

$$\begin{array}{ll} \underset{w,\gamma,y,z,\nu}{\text{minimize}} & (1-\lambda)(e^Ty + e^Tz) + \frac{\lambda}{2}\nu \\ \text{subject to} & \begin{array}{c} -Aw + e\gamma + e \leq y, \\ Bw - e\gamma + e \leq z, \\ -e\nu \leq w \leq e\nu, \\ y \geq 0, z \geq 0. \end{array} \end{array} \tag{12}$$

Similarly if we use the ∞-norm to measure the distance between the planes, then the dual to this norm is the 1-norm and accordingly $\|w\|' = \|w\|_1$ in (11) which leads to the following linear programming formulation:

$$\begin{array}{ll} \underset{w,\gamma,y,z,s}{\text{minimize}} & (1-\lambda)(e^Ty + e^Tz) + \frac{\lambda}{2}e^Ts \\ \text{subject to} & \begin{array}{c} -Aw + e\gamma + e \leq y, \\ Bw - e\gamma + e \leq z, \\ -s \leq w \leq s, \\ y \geq 0, z \geq 0. \end{array} \end{array} \tag{13}$$

We note that the first paper on the multisurface method on pattern separation [17] also proposed and implemented, just as does the support vector machine approach, forcing the two parallel planes that bound the sets to be separated to be as far apart as possible.

Usually the support vector machine problem is formulated using the 2-norm in the objective [27, 1]. Since the 2-norm is dual to itself, it follows that the distance between the parallel planes defining the separating surface is also measured in the 2-norm when this formulation is used. In this case $\|w\|' = \|w\|_2$, and one usually appends the term $\frac{\lambda}{2}\|w\|_2^2$ to the objective of (11) resulting in the following quadratic program:

$$\begin{array}{ll} \underset{w,\gamma,y,z}{\text{minimize}} & (1-\lambda)(e^Ty + e^Tz) + \frac{\lambda}{2}w^Tw \\ \text{subject to} & \begin{array}{c} -Aw + e\gamma + e \leq y, \\ Bw - e\gamma + e \leq z, \\ y \geq 0, z \geq 0. \end{array} \end{array} \tag{14}$$

Nonlinear separating surfaces, which are linear in their parameters, can also easily be handled by the formulations (8), (12) and (13) [16]. If the data are mapped nonlinearly via $\Phi : R^n \to R^\ell$, a nonlinear separating surface in R^n is easily computed as a linear separator in R^ℓ. In practice, one usually solves (14) by way of its dual [18]. In this formulation, the data enter only as inner products which are computed in the *transformed* space via a kernel function $K(x,y) = \Phi(x) \cdot \Phi(y)$ [6, 27, 28].

We note that separation errors in (12) - (14) are weighted equally conforming to the SVM formulations in [6, 27]. In contrast, the formulations (4) and (8) measure *average* separation error. Minimizing average separation error in (4) ensures that the solution $w = 0$ occurs iff $\frac{e^TA}{m} = \frac{e^TB}{k}$, in which case it is not unique [2, Theorem 2.5].

We turn our attention now to computational testing and comparison.

4 COMPUTATIONAL RESULTS

4.1 DATA SETS

The Wisconsin Prognostic Breast Cancer Database consists of 198 instances with 35 features representing follow-up data for one breast cancer case [23].

We used 2 variants of this data set. The first data set was created where the elements of the set $\mathcal{A}$ were 30 nuclear features plus diameter of excised tumor and number of positive lymph nodes of instances corresponding to patients in which cancer had recurred in less than 24 months (28 points). The set $\mathcal{B}$ consisted of the same features for patients in which cancer had not recurred in less than 24 months (127 points). The second variant of the data set consisted of the same 32 features, but but splits the data into $\mathcal{A}$ and $\mathcal{B}$ differently. Elements of $\mathcal{A}$ corresponds to patients with a cancer recurrence in less than 60 months (41 points) and $\mathcal{B}$ corresponds to patients which cancer had not recurred in less than 60 months (69 points).

The Johns Hopkins University Ionosphere data set consists of 34 continuous features of 351 instances [23]. Each instance represents a radar return from the ionosphere. The set $\mathcal{A}$ consists of 225 radar returns termed "good" or showing some type of structure in the ionosphere. The set $\mathcal{B}$ consists of 126 radar returns termed "bad"; their signals pass through the ionosphere.

The Cleveland Heart Disease data set consists of 297 instance with 13 features (see documentation [23]). Set $\mathcal{A}$ consist of 214 instance. The set $\mathcal{B}$ consists of 83 instances.

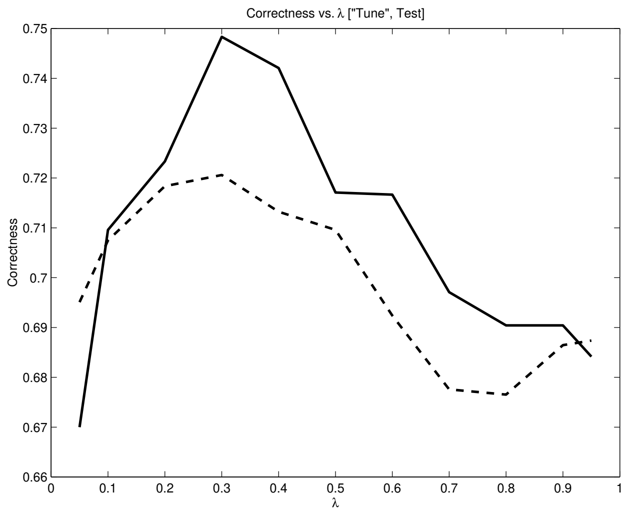

Figure 1: Tuning and testing sets correctness for a support vector machine (13) versus the sparsity-inducing parameter λ on the WPBC (24 months) data set. **Dashed** = "tuning" correctness, **Solid** = test correctness.

The Pima Indians Diabetes data set consists of 768 instances with 8 features plus a class label (see documentation [23]). The 500 instances with class label "0" were place in $\mathcal{A}$, the 268 instances with class label "1" were placed in $\mathcal{B}$.

The BUPA Liver Disorders data set consists of 345 instances with 6 features plus a selector field used to split the data into 2 sets (see documentation [23]). Set $\mathcal{A}$ consists of 145 instances, the set $\mathcal{B}$ consists of 200 instances.

4.2 EXPERIMENTAL METHODOLOGY

Our goal was to evaluate the generalization ability of the classifiers obtained by solving: the concave minimization problem FSV (8), SVM 1-norm problem (13), the SVM ∞-norm problem (12), the SVM 2-norm problem (14) and the robust linear program (RLP) (4). We estimate the generalization ability of a classifier via 10-fold cross-validation [26].

We note that the objective function parameter λ, which can induce sparsity, must be chosen carefully to maximize the generalization ability of the resulting classifier. Choosing $\lambda = 0$ will maximize the *training* correctness of the resulting classifier, but often this classifier performs poorly on data not in the training set [25]. We employ the following "tuning set" procedure for choosing λ at each fold of 10-fold cross-validation: For each λ in a candidate set Λ, we perform the following: (i) set aside 10% of the training data as a "tuning" set, (ii) obtain a classifier for the given value of λ, (iii) determine correctness on the "tuning" set, (iv) repeat steps (i)-(iii) ten times, each time setting aside a different 10% portion of the training data. The "score" for this value of λ is the average of the 10 correctness values determined in (iii).

We fix the value of λ as that with the best "score" determined from the tuning procedure (ties are broken by choosing the smallest λ-value). This is the value used for the given fold of 10-fold cross-validation. The set Λ is a set of candidate values and for these experiments was set at: $\Lambda = \{0.05, 0.10, 0.20, \ldots, 0.90, 0.95\}$. The curves in Figure 1 indicate that the value of λ that maximizes the "tuning" score (dashed curve in Figure 1) is a good estimate of the value of λ that maximizes the test set correctness (solid curve).

4.3 EXPERIMENTAL RESULTS

Table 1 summarizes the average number of original problem features selected by the classifiers trained by each of the methods.

Table 2 summarizes the results of the 10-fold cross-validation experiments on 6 real-world data sets. All "Train" and "Test" numbers presented are average correctnesses over 10-folds. The p-value is an indicator of significance difference in "Test" correctness between the classifiers obtained by solving FSV (8) and the classifiers obtained by solving the SVM 1-norm problem (13) [1]. Recall that a high p-value indicates that the difference is not significant. We note that p-values were not calculated for the other pairwise comparisons because the solutions obtained by solving the SVM ∞-norm, SVM 2-norm and the RLP did not suppress problem features (see Table 1).

4.4 DISCUSSION

The FSV (8) and the SVM 1-norm (13) problems where the only ones exhibiting feature selection (Table 1). On the 6 data sets tested, the SVM 1-norm classifiers performed slightly better on 3 data sets and FSV classifiers performed slightly better on 3 data sets. The minimum p-value is 0.1246 indicates that classifiers obtained by the FSV (8) and the SVM 1-norm (13) methods have similar generalization properties. Applying the paired t-test to 10-fold cross validation results may indicate a difference in the average test

[1] Specifically, this is the p-value of a two-tailed paired t-test testing the hypothesis that the difference in "Test" correctnesses for the FSV and SVM 1-norm classifiers is zero

set correctness when one is not present [9]. Thus the results of these experiments may be more similar than indicated by the p-values.

We note that the classifiers obtained by solving the SVM ∞-norm (12) suppressed none of the original problem features for all but the largest values of λ (near 1.0), which in general is of little use because it is often accompanied by poor set separation. Similar behavior was observed by solving the SVM 2-norm (14) problem. Note that the ∞-norm is sensitive to outliers, as is the 2-norm *squared*.

The classifiers obtained by solving the FSV problem (8) selected fewer problem features than the any of the SVM formulations (12), (13), (14) and the RLP (4) FSV classifiers reduced the number of features used over SVM 1-norm by as much as 39.5% (WPBC 60 month), while maintaining comparable generalization performance.

On the WPBC 24 month dataset, both the FSV classifiers (8) and the SVM 1-norm classifiers (13) most often selected a nuclear area feature and number of lymph nodes removed from the patient. These features are deemed relevant to the prognosis problem.

All linear programs formulations were solved using the CPLEX package [8] called from within MATLAB [22]. The quadratic programming problem (14) was solved using MATLAB's quadratic optimization solver, which encountered difficulty on conditioning the QP constraint matrix, which may affect the interpretation of the results for this approach. See Table 3 for average solve times.

5 SUMMARY AND FUTURE WORK

Computational comparisons of classifiers obtained by solving four mathematical optimization problems are presented. The optimization formulations are either linear (4), (12) and (13), or quadratic (14), or can be solved by a finite sequence of linear programs (solving (8) via Algorithm 2.1). **Classifiers obtained by solving the FSV problem (8) and the SVM 1-norm problem (13) exhibit feature suppression and have comparable generalization performance on six publicly available real world data sets tested. The classifiers obtained by solving the FSV problem (8) suppressed more features than the corresponding SVM 1-norm classifiers (13). The quadratic SVM (14) took orders of magnitude more time than the linear-programming-based SVMs (12) and (13).**

When the distance between the 2 parallel planes defining the separating surface in the SVM problem is chosen to be the 1-norm, the resulting SVM optimization problem has the ∞-norm (dual norm to the 1-norm) appearing in the objective. The classifiers obtained by solving this problem (SVM ∞-norm (12)) did not exhibit feature selection. Similar behavior was observed for classifiers obtained by solving the SVM 2-norm (14) problem. The generalization ability of these classifiers in comparison with the others presented needs to be further investigated.

Future work includes further analysis of the benefits of measuring the distance between the bounding parallel planes defining the separating plane and the resulting optimization problem utilizing the dual norm (11). A characterization of classes of data sets which lend themselves to better separation with the choice of one norm over another will allow practitioners to choose *a priori* an optimization formulation believed to be "best" suited to the separation problem at hand.

Acknowledgements

This work was supported by National Science Foundation Grants CCR-9322479, CCR-9729842 and Air Force Office of Scientific Research Grant F49620-97-1-0326 as Mathematical Programming Technical Report 98-03, February 1998.

References

[1] K. P. Bennett and J. A. Blue. A support vector machine approach to decision trees. Department of Mathematical Sciences Math Report No. 97-100, Rensselaer Polytechnic Institute, Troy, NY 12180, 1997. http://www.math.rpi.edu/ bennek/.

[2] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.

[3] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Information Processing Letters*, 24:377–380, 1987.

[4] P. S. Bradley, O. L. Mangasarian, and J. B. Rosen. Parsimonious least norm approximation. Technical Report 97-03, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, March

Table 1: Average number of features selected in the classifiers. (Asterisk * indicates that the full experiment had not been carried out because of excessive time (see Table 3),hence results are averaged over folds completed.)

Data Set	FSV (8)	SVM $\|\cdot\|_1$ (13)	SVM $\|\cdot\|_\infty$ (12)	SVM $\|\cdot\|_2^2$ (14)	RLP (4)
WPBC (24 mo.)	3.9	5.4	32	32	32
WPBC (60 mo.)	2.6	4.3	32	32	32
Ionosphere	10.4	11.1	34	33*	33
Cleveland	6.4	9.3	13	13*	13
Pima Indians	5.3	6.0	8	*	8
BUPA Liver	4.5	5.8	6	6*	6

Table 2: Ten-fold cross-validation correctness results on 6 publicly available data sets. (Asterisk * indicates that the full experiment had not been carried out because of excessive time (see Table 3), hence results are averaged over folds completed.)

Data Set	FSV (8) Train Test	SVM $\|\cdot\|_1$ (13) Train Test	p-Value	SVM $\|\cdot\|_\infty$ (12) Train Test	SVM $\|\cdot\|_2^2$ (14) Train Test	RLP (4) Train Test
WPBC (24 mo.)	73.97	74.40		73.69	82.86	85.23
	66.42	71.08	0.1246	71.04	75.46	67.12
WPBC (60 mo.)	70.70	71.21		73.34	75.54	87.58
	67.05	66.23	0.6408	66.38	66.21	63.50
Ionosphere	90.47	88.92		89.65	94.56*	94.78
	84.07	86.10	0.1254	84.06	85.75*	86.04
Cleveland	83.57	85.30		85.82	84.70*	86.31
	80.94	84.55	0.1819	82.52	75.86*	83.87
Pima Indians	75.22	75.52		76.01	*	76.48
	74.60	74.47	0.8889	74.99	*	76.16
BUPA Liver	68.18	67.83		68.73	60.22*	68.98
	65.20	64.03	0.1696	64.63	60.95*	64.34

1997. *Computational Optimization and Applications*, to appear. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/97-03.ps.Z.

[5] P. S. Bradley, O. L. Mangasarian, and W. N. Street. Feature selection via mathematical programming. INFORMS Journal on Computing, 1998. To appear. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/95-21.ps.Z.

[6] C. J. C. Burges. A tutorial on support vector machines. *Data Mining and Knowledge Discovery*, 2, 1998. To appear.

[7] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–279, 1995.

[8] CPLEX Optimization Inc., Incline Village, Nevada. *Using the CPLEX(TM) Linear Optimizer and CPLEX(TM) Mixed Integer Optimizer (Version 2.0)*, 1992.

[9] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 1997. To appear. http://www.cs.orst.edu/ tgd/cv/pubs.html.

[10] F. Girosi. An equivalence between sparse approximation and support vector machines. A.I. Memo 1606, Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetts, 1997. http://www.ai.mit.edu/people/girosi/home-page/svm.html.

[11] M. H. Hassoun. *Fundamentals of Artificial Neural Networks*. MIT Press, Cambridge, MA, 1995.

Table 3: Average running: Ionosphere data set.

Method	Time (Seconds)
Algorithm 2.1	30.94
$\|\cdot\|_1$ (13)	3.09
$\|\cdot\|_\infty$ (12)	1.42
$\|\cdot\|_2^2$ (14)	2956.8
RLP (4)	1.28

[12] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, California, 1991.

[13] P. J. Huber. *Robust Statistics*. John Wiley, New York, 1981.

[14] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the 11th International Conference on Machine Learning*, San Mateo, CA, 1994. Morgan Kaufmann.

[15] D. Koller and M. Sahami. Toward optimal feature selection. In L. Saitta, editor, *Machine Learning–Proceedings of the Thirteenth International Conference (ICML '96)–Bari, Italy July 3-6, 1996*, pages 284–292, San Francisco, CA, 1996. Morgan Kaufmann.

[16] O. L. Mangasarian. Linear and nonlinear separation of patterns by linear programming. *Operations Research*, 13:444–452, 1965.

[17] O. L. Mangasarian. Multi-surface method of pattern separation. *IEEE Transactions on Information Theory*, IT-14:801–807, 1968.

[18] O. L. Mangasarian. *Nonlinear Programming*. McGraw–Hill, New York, 1969. Reprint: SIAM Classic in Applied Mathematics 10, 1994, Philadelphia.

[19] O. L. Mangasarian. Machine learning via polyhedral concave minimization. In H. Fischer, B. Riedmueller, and S. Schaeffler, editors, *Applied Mathematics and Parallel Computing - Festschrift for Klaus Ritter*, pages 175–188. Physica-Verlag A Springer-Verlag Company, Heidelberg, 1996. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/95-20.ps.Z.

[20] O. L. Mangasarian. Arbitrary-norm separating plane. Technical Report 97-07, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, May 1997. *Operations Research Letters*, submitted. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/97-07.ps.Z.

[21] O. L. Mangasarian, W. N. Street, and W. H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4):570–577, July-August 1995.

[22] MATLAB. *User's Guide*. The MathWorks, Inc., 1992.

[23] P. M. Murphy and D. W. Aha. UCI repository of machine learning databases. Technical report, Department of Information and Computer Science, University of California, Irvine, 1992. www.ics.uci.edu/ mlearn/MLRepository.html.

[24] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *IEEE Confernece on Computer Vision and Pattern Recognition, Puerto Rico, June 1997, 130-136*, 1997. http://www.ai.mit.edu/people/girosi/home-page/svm.html.

[25] J. W. Shavlik and T. G. Dietterich (editors). *Readings in Machine Learning*. Morgan Kaufman, San Mateo, California, 1990.

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

[27] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[28] G. Wahba. Support vector machines, reproducing kernel Hilbert spaces and the randomized gacv. Technical report no. 984, Department of Statistics, University of Wisconsin, Madison, WI 53706, 1997. ftp://ftp.stat.wisc.edu/pub/wahba/index.html.