Data Selection for Support Vector Machine Classifiers

Glenn Fung and Olvi L. Mangasarian
Computer Sciences Department
University of Wisconsin
1210 West Dayton Street
Madison, WI 53706
(gfung,olvi)@cs.wisc.edu

ABSTRACT

The problem of extracting a minimal number of data points from a large dataset, in order to generate a support vector machine (SVM) classifier, is formulated as a concave minimization problem and solved by a finite number of linear programs. This minimal set of data points, which is the smallest number of support vectors that completely characterize a separating plane classifier, is considerably smaller than that required by a standard 1-norm support vector machine with or without feature selection. The proposed approach also incorporates a feature selection procedure that results in a minimal number of input features used by the classifier. Tenfold cross validation gives as good or better test results using the proposed minimal support vector machine (MSVM) classifier based on the smaller set of data points compared to a standard 1-norm support vector machine classifier. The reduction in data points used by an MSVM classifier over those used by a 1-norm SVM classifier averaged 66% on seven public datasets and was as high as 81%. This makes MSVM a useful incremental classification tool which maintains only a small fraction of a large dataset before merging and processing it with new incoming data.

Keywords

support vector machines, data classification, data selection, concave minimization, linear programming

1. INTRODUCTION

Support vector machines [20, 8, 3, 7, 14] are powerful tools for data classification. Classification is achieved by a linear or nonlinear separating surface in the input space of the dataset. The separating surface depends only on a subset of the original data. This subset of data, which is all that is needed to generate the separating surface, constitutes the set of support vectors. In this paper we give a method for selecting as small a set of support vectors as possible which completely determines a separating plane classifier. We term such a set of support vectors *minimal*, and the corresponding classifier, a *minimal support vector machine*. Such a classifier turns out to have improved testing set accuracy over one chosen by a standard support vector machine. Mathematically, support vectors are data points corresponding to constraints with positive multipliers in a constrained optimization formulation of a support vector machine. Computationally, the problem of determining a minimal set of support vectors does not appear to have been addressed before as proposed in this paper. This is an important problem in applications such as fraud detection where the dataset may contain millions of data points. To make support vector machines viable for such applications, it is important to identify a minimal set of support vectors, often an order of magnitude smaller than the original dataset, which determines the separating surface and allows the removal of redundant data. This dependence on a small subset of a given dataset, which often leads to an improved classifier, can be utilized in an incremental approach such as chunking [3, 15] where a small fraction of the data is maintained before merging and processing it with new incoming data. For the sake of simplicity and getting basic ideas across we shall confine ourselves here to linear separating surfaces.

We briefly summarize the contents of the paper now. In Section 2 we introduce the linear support vector machine as a separating plane classifier midway between and parallel to two bounding planes, with maximum margin (distance) between them. See Figures 1 and 2. The bounding planes attempt to place the two classes of a given dataset on opposite sides. The separating plane is obtained by solving a quadratic program (1) or a linear program (8), depending on the norm used in measuring the margin between the bounding planes. In order to incorporate a concave suppression term in the objective function that eliminates as many redundant data points as possible and still maintain concavity of the objective function for computational purposes, we utilize the linear programming formulation (8) and combine it with a step function in (9) to eliminate as many misclassified points as possible. This translates into a minimal set of support vectors that determine the separating plane. The Successive Linearization Algorithm (SLA) 3.1 obtains a very effective local solution to (9) by solving 4 to 7 linear programs. This leads to a classifier with as good or improved generalization and which depends on a substantially smaller number of data points when compared to other classifiers, as shown by the numerical tests of Section 4 on seven public datasets. These results indicate a reduction of support vectors, i.e. data points that define the separating surface, as high as 81% and a corresponding test set correctness increase of 5.6%.

We now describe our notation and give some background material. All vectors will be column vectors unless transposed to a row vector by a prime $'$. For a vector x in the n-dimensional real space 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 $(x_*)_i = 1$ 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 $1 \leq p < \infty$, the p-norm and the ∞-norm are defined as follows:

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

The notation $A \in R^{m\times n}$ will signify a real $m \times n$ matrix. For such a matrix A' will denote the transpose of A, and A_i will denote the i-th row of A. A column vector of ones in a real space of arbitrary dimension will be denoted by e. Thus, for the column vectors e and y in R^m, the scalar product $e'y$ denotes the sum $\sum_{j=1}^{m} y_i$. A vector of zeros in a real space of arbitrary dimension will be denoted by 0. 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 $\mathcal{B}$. A real valued function $f(x)$ on R^n is concave ("mountain-like") if linear interpolation between two function values never overestimates the function.

2. THE LINEAR SUPPORT VECTOR MACHINE

We consider the problem, depicted in Figures 1 and 2, of classifying m points in the n-dimensional real space R^n, represented by the $m\times n$ matrix A, according to membership of each point A_i in the class $A+$ or $A-$ as specified by a given $m\times m$ diagonal matrix D with plus ones or minus ones along its diagonal. For this problem the standard support vector machine with a linear kernel [20, 8] is given by the following quadratic program with parameter $\nu > 0$:

$$\begin{array}{rrcl} \displaystyle\min_{(w,\gamma,y)\in R^{n+1+m}} & \nu e'y + \frac{1}{2}w'w & & \\ \text{s.t.} & D(Aw - e\gamma) + y & \geq & e \\ & y & \geq & 0. \end{array} \tag{1}$$

Written in individual component notation, and taking into account that D is a diagonal matrix of $\pm$ 1, this problem becomes:

$$\begin{array}{rrcll} \displaystyle\min_{(w,\gamma,y)\in R^{n+1+m}} & \displaystyle\nu\sum_{i=1}^{m} y_i + \frac{1}{2}\sum_{j=1}^{n} w_j^2 & & & \\ \text{s.t.} & A_iw + y_i & \geq & \gamma + 1, & \text{for} \quad D_{ii} = 1 \\ & A_iw - y_i & \leq & \gamma - 1, & \text{for} \quad D_{ii} = -1 \\ & y_i & \geq & 0 & \\ & i & = & 1 \ldots = m. & \end{array} \tag{2}$$

Here, w is the normal to the bounding planes:

$$\begin{array}{rcl} x'w & = & \gamma + 1 \\ x'w & = & \gamma - 1, \end{array} \tag{3}$$

and γ determines their location relative to the origin. See Figure 1. When the two classes are strictly linearly separable, that is when the error variable $y = 0$ in (1)-(2), as in the case of Figure 1, the plane $x'w = \gamma + 1$ bounds the class $A+$ points, while the plane $x'w = \gamma - 1$ bounds the class $A-$ points as follows:

$$\begin{array}{rcll} A_iw & \geq & \gamma + 1, & \text{for} \quad D_{ii} = 1 \\ A_iw & \leq & \gamma - 1, & \text{for} \quad D_{ii} = -1. \end{array} \tag{4}$$

The linear separating surface is the plane:

$$x'w = \gamma, \tag{5}$$

midway between the bounding planes (3). The quadratic term in (1), which is twice the reciprocal of the square of the 2-norm distance $\frac{2}{\|w\|_2}$ between the two bounding planes of (3) (see Figure 1), maximizes this distance, often called the "margin". Maximizing the margin enhances the generalization capability of a support vector machine [20, 8].

If the classes are linearly inseparable then the two planes bound the two classes with a "soft margin" (i.e. bound approximately with some error) determined by the nonnegative error variable y, that is:

$$\begin{array}{rcrcll} A_iw & + & y_i & \geq & \gamma + 1, & \text{for } D_{ii} = 1 \\ A_iw & - & y_i & \leq & \gamma - 1, & \text{for } D_{ii} = -1. \end{array} \tag{6}$$

The 1-norm of the error variable y is minimized parametricly with weight ν in (1) resulting in an approximate separation as depicted in Figure 2, for example. Points of $A+$ that lie in the halfspace $\{x \mid x'w \leq \gamma + 1\}$ (i.e. on the plane $x'w = \gamma + 1$ and on the wrong side of the plane) as well as points of $A-$ that lie in the halfspace $\{x \mid x'w \geq \gamma - 1\}$ are called *support vectors*.

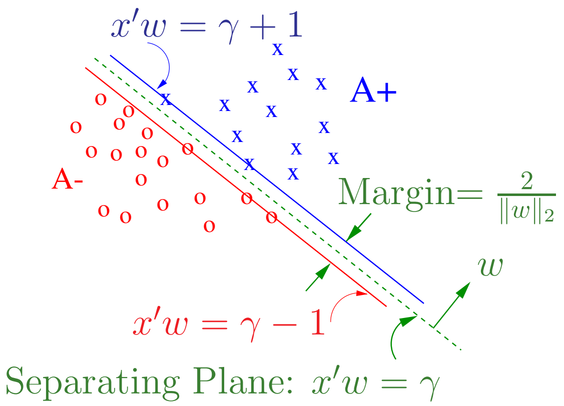

Figure 1: The Linearly Separable Case: The bounding planes of equation (3) with margin $\frac{2}{\|w\|_2}$, and the plane of equation (5) separating $A+$, the points represented by rows of A with $D_{ii} = +1$, from $A-$, the points represented by rows of A with $D_{ii} = -1$.

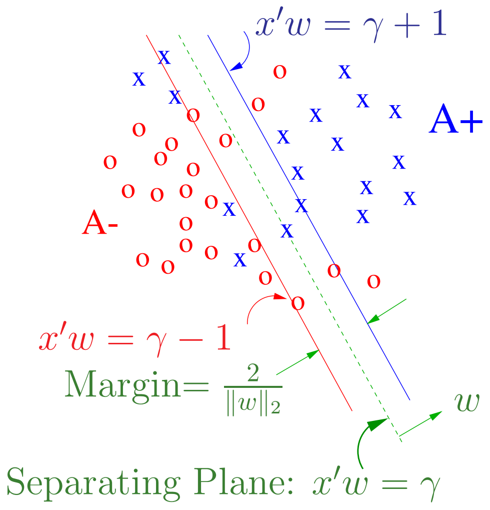

Figure 2: The Linearly Inseparable Case: The approximately bounding planes of equation (3) with a soft (i.e. with some error) margin $\frac{2}{\|w\|_2}$, and the plane of equation (5) approximately separating $A+$ from $A-$.

Support vectors, which constitute the complement of the strictly separated points by the bounding planes (3), completely determine the separating surface. Minimizing the number of such exceptional points can lead to a minimum length description model [17, p. 66],[1] that depends on much fewer data points. Computational results indicate that such lean models generalize as well or better than models that depend on many more data points. We give in the next section of the paper an algorithm that minimizes the number of support vectors that determine the separating plane as well as the number of input space features.

3. MSVM: A MINIMAL SUPPORT VECTOR MACHINE

In order to make use of a faster linear programming based approach, instead of the standard quadratic programming formulation (1), we reformulate (1) by replacing the 2-norm by a 1-norm as follows [14, 2]:

$$\begin{array}{rl} \displaystyle\min_{(w,\gamma,y)\in R^{n+1+m}} & \nu e'y + \|w\|_1 = \nu\sum_{i=1}^{m} y_i + \sum_{j=1}^{n}|w_j| \\ \text{s.t.} & D(Aw - e\gamma) + y \geq e \\ & y \geq 0. \end{array} \tag{7}$$

This SVM$\|\cdot\|_1$ reformulation in effect maximizes the margin, the distance between the two bounding planes of Figures 1 and 2, using a different norm, the ∞-norm, and results with a margin in terms of the 1-norm, $\frac{2}{\|w\|_1}$, instead of $\frac{2}{\|w\|_2}$ [13]. The mathematical program (7) is easily converted to a linear program as follows:

$$\begin{array}{rl} \displaystyle\min_{(w,\gamma,y,v)\in R^{n+1+m+n}} & \nu e'y + e'v = \nu\sum_{i=1}^{m} y_i + \sum_{j=1}^{n} v_j \\ \text{s.t.} & D(Aw - e\gamma) + y \geq e \\ & v \geq w \geq -v \\ & y \geq 0, \end{array} \tag{8}$$

where, at a solution, v is the absolute value $|w|$ of w. This fact follows from the constraints $v \geq w \geq -v$ which imply that $v_i \geq |w_i|,\ i = 1\ldots,n$. Hence at optimality, $v = |w|$, otherwise the objective function can be strictly decreased without changing any variable except v. We will modify this linear program so as to generate an SVM with as few support vectors as possible by adding an error term $e'y_*$ to the objective function, where $_*$ denotes the step function as defined in the Introduction. The term $e'y_*$ suppresses misclassified points and results in our minimal support vector machine MSVM:

$$\begin{array}{rl} \displaystyle\min_{(w,\gamma,y,v)\in R^{n+1+m+n}} & \nu e'y + e'v + \mu e'y_* \\ \text{s.t.} & D(Aw - e\gamma) + y \ \geq\ e \\ & v \geq\ w \geq\ -v \\ & y\ \geq\ 0. \end{array} \tag{9}$$

Note that when the error vector y is zero all the points have been strictly separated by the plane $x'w = \gamma$. Thus, the separation error term in the objective function of (9) results in:

$$e'y_* = \sum_{i=1}^{m} y_{i_*} = \bar{m}, \tag{10}$$

where $\bar{m}$ is the number of positive components of y_i, or equivalently the number of misclassified points by the bounding planes $x'w = \gamma \pm 1$. This number is directly related to the number of support vectors as shown below following equation (13). The positive parameter μ, chosen by a tuning set, multiplies the term $e'y_*$ which eliminates positive components of the error variable y. The justification for eliminating components of the error vector y, other than the intuitive idea of having a separating surface with as few misclassified points as possible, is as follows. If we define nonnegative multipliers $u \in R^m$ associated with the first set of constraints of the linear program (8), and multipliers $(r, s) \in R^{n+n}$ for the second set of constraints of (8), then the dual linear program [9] associated with the linear SVM formulation (8) is the following:

$$\begin{array}{rrcl} \displaystyle\max_{(u,r,s)\in R^{m+n+n}} & & e'u & \\ \text{s.t.} & A'Du - r + s & = & 0 \\ & -e'Du & = & 0 \\ & u & \leq & \nu e \\ & r + s & = & e \\ & u,\ r,\ s & \geq & 0. \end{array} \tag{11}$$

Equality of the primal objective function of (8) and the dual objective function of (11) imply the (equality) complementarity conditions of the following Karush-Kuhn-Tucker optimality conditions [10, p. 94] for (8):

$$\begin{array}{rcl} u'(D(Aw-e\gamma)+y-e) & = & 0 \\ u & \geq & 0 \\ D(Aw-e\gamma)+y-e & \geq & 0 \\ y'(\nu e-u) & = & 0 \\ y & \geq & 0 \\ \nu e-u & \geq & 0. \end{array} \qquad (12)$$

These optimality conditions lead to the following implications for $i=1,\dots,m$:

$$\begin{array}{rr} y_i>0 & \\ \implies & u_i=\nu>0 \\ \implies & D_{ii}(A_iw-\gamma)-1=-y_i<0. \end{array} \qquad (13)$$

Thus, a positive y_i implies a positive multiplier $u_i=\nu>0$ and a corresponding support vector A_i that violates (4). Consequently eliminating positive components of y tends to minimize the number of multipliers at the upper bound ν as well as data points A_i that violate (4), that is, points that lie on the wrong sides of the bounding planes (3). Minimizing $e'y_*$ works remarkably well computationally in eliminating positive components of the multiplier u and consequently the number of misclassified points.

Even though the discontinuity of the step function term $e'y_*$ in (9) can be handled directly by an algorithm such as that of [12, Algorithm 1 SLA], we prefer to approximate it here by a smooth concave exponential on the nonnegative real line [11] as was done in the feature selection approach of [2]. For $y\geq 0$, the approximation of the step vector y_* of (9) by the concave exponential, $y_{i*}\approx 1-\varepsilon^{-\alpha y_i}$, $i=1,\dots,m$, that is:

$$y_*\approx e-\varepsilon^{-\alpha y},\ \alpha>0, \qquad (14)$$

where ε is the base of natural logarithms, leads to the following smooth reformulation of problem (9), the smooth MSVM:

$$\begin{array}{rl} \displaystyle\min_{(w,\gamma,y,v)\in R^{n+1+m+n}} & \nu e'y+e'v+\mu e'(e-\varepsilon^{-\alpha y}) \\ \text{s.t.} & D(Aw-e\gamma)+y\geq e \\ & v\geq w\geq -v \\ & y\geq 0. \end{array} \qquad (15)$$

Note that:

$$e'(e-\varepsilon^{-\alpha y})=m-\sum_{i=1}^{m}\varepsilon^{-\alpha y_i}. \qquad (16)$$

It can be shown [4, Theorem 2.1] that, for a finite value of the parameter α (appearing in the concave exponential), the smooth problem (15) generates an *exact* solution of the nonsmooth problem (9). 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 4 to 7 steps) at a stationary point which satisfies the minimum principle necessary optimality condition for problem (15) [5, Theorem 2.2] and leads to a locally minimal number of support vectors, that is, a minimal number of data points A_i with positive multipliers u_i that completely determine the separating surface.

ALGORITHM 3.1. **Successive Linearization Algorithm (SLA) for (15).** *Choose $\nu,\mu,\alpha>0$. Start with some (w^0,γ^0,y^0,v^0). Having (w^i,γ^i,y^i,v^i) determine the next iterate by solving the linear program:*

$$\begin{array}{rl} \displaystyle\min_{(w,\gamma,y,v)\in R^{n+1+m+n}} & \nu e'y+e'v+\mu\alpha(\varepsilon^{-\alpha y^i})'(y-y^i) \\ s.t. & D(Aw-e\gamma)+y\geq e \\ & v\geq w\geq -v \\ & y\geq 0. \end{array} \qquad (17)$$

Stop when:

$$\nu e'(y-y^i)+e'(v-v^i)+\mu\alpha(\varepsilon^{-\alpha y^i})'(y-y^i)=0. \qquad (18)$$

Comment: The parameter α was set to 5. The parameters ν and μ were chosen with the help of a tuning set surrogate for a testing set to simultaneously minimize the number of support vectors, number of input space features and tuning set error.

We turn our attention now to numerical implementation and testing.

4. NUMERICAL IMPLEMENTATION AND COMPARISONS

Before applying Algorithm 3.1, which typically consists of solving 4 to 7 linear programs, the dimensionality of $w\in R^n$ was reduced by solving the 1-norm SVM (8), as a single linear program, with weight $\nu\in[0.01,0.1]$ and discarding components of w less than 10^{-8} in magnitude. The reason for this dimensionality reduction, described more fully in [2], is the presence of the term $\|w\|_1$ in (7), which suppresses components of w.

The remaining components of w with the corresponding values of γ, y and v were used as the initial starting point (w^0,γ^0,y^0,v^0) in Algorithm 3.1. After the termination of Algorithm 3.1, only support vectors were kept, that is A_i for which the multiplier $u_i>10^{-8}$. This small set of support vectors generated the same stationary point for problem (15) as that generated by the entire dataset. Such stationary points, which satisfy necessary optimality conditions, are typically good candidates to being a global solution to optimization problems of the type considered here.

The smooth minimal support vector machine (MSVM) (15) which generates a linear separating surface (5) by using a minimum number of data points was compared with the 1-norm support vector machine SVM $\|\cdot\|_1$ (7) as well as the the 1-norm support vector machine with feature selection FSV [2] which is problem (7) with the added feature-suppression term $\mu e'|w|_*$ in the objective function and smoothed to:

$$\mu e'(e-\varepsilon^{-\alpha|w|})=\mu\sum_{i=1}^{n}(1-\varepsilon^{-\alpha|w_i|}). \qquad (19)$$

This smoothing, similar to that of (15)-(16) except that it is applied here to $|w|$ instead of y, leads to a selection of $\bar{n}(< n)$ input space features. The three classifiers MSVM (15), SVM$\|\cdot\|_1$ (8) and FSV [2, Eqn. (8)] were tested on seven datasets, the first five of which, WPBC, Ionosphere, Cleveland Heart, Pima Indians, and BUPA Liver are from the Irvine Machine Learning Repository [18], while the Galaxy Dim dataset is from [19], and the Census dataset is a version of the US Census Bureau "Adult" dataset, which is publicly available from Silicon Graphics' website [6]. For WPBC(60), 110 breast cancer patients were classified into those who had a recurrence of the disease within 60 months and those who had not. For the Census dataset, ten features were used to predict whether the income of a person was greater or equal to the mean income or below it. Our computational results are summarized in Table 1 for the three classifiers. We make the following observations based on numerical results:

1. For all test problems MSVM had the least number of support vectors. This translates into the least number of data points selected for determining the separating surface and may be interpreted as a minimum description length model [17, p. 66],[1].

2. For the Ionosphere problem, the reduction in the number of support vectors of MSVM over SVM $\|\cdot\|_1$ is 81% with a corresponding increase of tenfold test set correctness of 5.6% with associated p value of 0.0003. (The p value measures the probability that two test results are the same, with sameness typically rejected if $p \leq 0.05$ [17]). For the seven datasets, the average reduction in the number of support vectors of MSVM over SVM $\|\cdot\|_1$ is 65.8%.

3. Tenfold testing set correctness of MSVM was as good or better on all seven datasets.

4. Computing times were higher for MSVM than those for SVM$\|\cdot\|_1$ and FSV. For example, one fold testing for the Galaxy Dim problem took 43.7 seconds on a 400 MHz Pentium II using MATLAB [16], while the correponding times were 11.2 seconds for SVM$\|\cdot\|_1$ and 16.0 seconds for FSV. One justification of the additional time taken by MSVM is that it trades support vector storage space needed to generate a separating surface, with a one-time additional computational time expense.

5. CONCLUSION AND FUTURE WORK

We have proposed a minimal support vector machine that extracts a minimum number of points from a given dataset in order to define a separating surface that classifies the dataset into two categories, based on this minimal subset of the data only. This minimality property which is in the spirit of Occam's Razor [1], not only is useful in classifying very large datasets by using only a fraction of the data, but also maintains or improves generalization over other classifiers that use a considerably higher number of data points in order to determine the separating surface. Elimination of a large portion of a dataset makes MSVM suitable as an incremental algorithm that maintains only a small portion of a large dataset before merging and processing it with new incoming data. Since MSVM requires the solution of a few linear programs to determine a separating surface, this makes it easier than a standard support vector machine that uses a quadratic programming formulation [20, 8].

Our future work includes the application of MSVM to massive datasets using chunking approaches [3, 15] that break a linear program into smaller ones, as well as extension to nonlinear separating surfaces generated by generalized nonlinear support vector machines [14] where the dependence on the training data size becomes more critical. The potential of MSVM as an incremental algorithm will be utilized in all these applications to solve massive data classification problems.

Acknowledgements
The research described in this Data Mining Institute Report 00-02, February 2000, was supported by National Science Foundation Grants CCR-9729842 and CDA-9623632, by Air Force Office of Scientific Research Grant F49620-00-1-0085 and by the Microsoft Corporation.

Table 1: Tenfold training and testing correctness, number of features (#Features) and number of support vectors (#SV) used in seven public datasets by an MSVM classifier, and by a 1-norm SVM classifier without feature selection SVM $\|\cdot\|_1$ and with feature selection (FSV).

Data Set	MSVM (Eqn. (15))	SVM $\|\cdot\|_1$ (Eqn. (8))	FSV [2, Eqn. (8)]
$m \times n$	Train	Train	Train
	Test	Test	Test
	#Features	#Features	#Features
	#SV	#SV	#SV
WPBC (60 mo.)	76.4%	69.5%	69.5%
110×32	68.3%	62.1%	62.1%
	5.0	4.3	2.6
	29.6	**69.4**	**69.1**
Ionosphere	91.4%	84.6%	91.2%
351×34	88.9%	84.2%	86.5%
	7.0	7.2	8.1
	34.2	**179.9**	**80.8**
Cleveland Heart	89.5%	86.8%	87.0%
297×13	86.9%	85.8%	85.2%
	9.2	8.8	8.9
	38.5	**109.8**	**92.4**
Pima Indians	76.6%	77.1%	76.9%
768×8	79.6%	76.5%	75.9%
	7.5	6.8	5.0
	150.1	**374.8**	**396.3**
BUPA Liver	72.7%	71.3%	70.0%
345×6	70.0%	69.9%	67.3%
	6.0	6.0	4.5
	91.9	**236.8**	**236.7**
Galaxy Dim	95.0%	94.4%	94.9%
4192×14	94.7%	94.4%	94.7%
	5.0	5.0	4.9
	193.0	**774.0**	**541.0**
Census	94.0%	93.9%	94.0%
$20,000 \times 10$	94.1%	94.0%	93.8%
	9.3	9.8	7.0
	1065.0	**2745.5**	**2783.2**

6. REFERENCES

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

[2] P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In J. Shavlik, editor, *Machine Learning Proceedings of the Fifteenth International Conference(ICML '98)*, pages 82–90, San Francisco, California, 1998. Morgan Kaufmann. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-03.ps.

[3] P. S. Bradley and O. L. Mangasarian. Massive data discrimination via linear support vector machines. *Optimization Methods and Software*, 13:1–10, 2000. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-03.ps.

[4] P. S. Bradley, O. L. Mangasarian, and J. B. Rosen. Parsimonious least norm approximation. *Computational Optimization and Applications*, 11(1):5–21, October 1998. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/97-03.ps.

[5] P. S. Bradley, O. L. Mangasarian, and W. N. Street. Feature selection via mathematical programming. *INFORMS Journal on Computing*, 10(2):209–217, 1998. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/95-21.ps.

[6] US Census Bureau. Adult dataset. Publicly available from: www.sgi.com/Technology/mlc/db/.

[7] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[8] V. Cherkassky and F. Mulier. *Learning from Data - Concepts, Theory and Methods*. John Wiley & Sons, New York, 1998.

[9] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, New Jersey, 1963.

[10] O. L. Mangasarian. *Nonlinear Programming*. SIAM, Philadelphia, PA, 1994.

[11] 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.

[12] O. L. Mangasarian. Solution of general linear complementarity problems via nondifferentiable concave minimization. *Acta Mathematica Vietnamica*, 22(1):199–205, 1997. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/96-10.ps.

[13] O. L. Mangasarian. Arbitrary-norm separating plane. *Operations Research Letters*, 24:15–23, 1999. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/97-07r.ps.

[14] O. L. Mangasarian. Generalized support vector machines. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 135–146, Cambridge, MA, 2000. MIT Press. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-14.ps.

[15] O. L. Mangasarian and David R. Musicant. Data discrimination via nonlinear generalized support vector machines. Technical Report 99-03, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, March 1999. To appear in: "Applications and Algorithms of Complementarity", M. C. Ferris, O. L. Mangasarian and J.-S. Pang, editors, Kluwer Academic Publishers,Boston 2000. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/99-03.ps.

[16] MATLAB. *User's Guide*. The MathWorks, Inc., Natick, MA 01760, 1992.

[17] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Boston, 1997.

[18] P. M. Murphy and D. W. Aha. UCI repository of machine learning databases, 1992. www.ics.uci.edu/∼mlearn/MLRepository.html.

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

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