RSVM: Reduced Support Vector Machines

Yuh-Jye Lee [*] ***and Olvi L. Mangasarian*** [†]

1 Introduction

Abstract An algorithm is proposed which generates a nonlinear kernel-based separating surface that requires as little as 1% of a large dataset for its explicit evaluation. To generate this nonlinear surface, the *entire* dataset is used as a constraint in an optimization problem with very few variables corresponding to the 1% of the data kept. The remainder of the data can be thrown away after solving the optimization problem. This is achieved by making use of a *rectangular* $m \times \bar{m}$ kernel $K(A, \bar{A}')$ that greatly reduces the size of the quadratic program to be solved and simplifies the characterization of the nonlinear separating surface. Here, the m rows of A represent the original m data points while the $\bar{m}$ rows of $\bar{A}$ represent a greatly reduced $\bar{m}$ data points. Computational results indicate that test set correctness for the reduced support vector machine (RSVM), with a nonlinear separating surface that depends on a small randomly selected portion of the dataset, is better than that of a conventional support vector machine (SVM) with a nonlinear surface that explicitly depends on the entire dataset, and much better than a conventional SVM using a small random sample of the data. Computational times, as well as memory usage, are much smaller for RSVM than that of a conventional SVM using the entire dataset.

Support vector machines have come to play a very dominant role in data classification using a kernel-based linear or nonlinear classifier [23, 6, 21, 22]. Two major problems that confront large data classification by a nonlinear kernel are:

1. The sheer size of the mathematical programming problem that needs to be solved and the time it takes to solve, even for moderately sized datasets.

[*] Computer Sciences Department, University of Wisconsin, Madison, WI 53706. *yuh-jye@cs.wisc.edu*.
[†] Computer Sciences Department, University of Wisconsin, Madison, WI 53706. *olvi@cs.wisc.edu*, corresponding author.

2. The dependence of the nonlinear separating surface on the entire dataset which creates unwieldy storage problems that prevents the use of nonlinear kernels for anything but a small dataset.

For example, even for a thousand point dataset, one is confronted by a fully dense quadratic program with 1001 variables and 1000 constraints resulting in constraint matrix with over a million entries. In contrast, our proposed approach would typically reduce the problem to one with a 101 variables and a 1000 constraints which is readily solved by a smoothing technique [10] as an unconstrained 101-dimensional minimization problem. This generates a nonlinear separating surface which depends on a hundred data points only, instead of the conventional nonlinear kernel surface which would depend on the entire 1000 points. In [24], an approximate kernel has been proposed which is based on an eigenvalue decomposition of a randomly selected subset of the training set. However, unlike our approach, the entire kernel matrix is generated within an iterative linear equation solution procedure. We note that our data-reduction approach should work equally well for 1-norm based support vector machines [1], chunking methods [2] as well as Platt's sequential minimization optimization (SMO) [19].

We briefly outline the contents of the paper now. In Section 2 we describe kernel-based classification for linear and nonlinear kernels. In Section 3 we outline our reduced SVM approach. Section 4 gives computational and graphical results that show the effectiveness and power of RSVM. Section 5 concludes the paper.

A word about our notation and background material. All vectors will be column vectors unless transposed to a row vector by a prime superscript $'$. For a vector x in the n-dimensional real space R^n, the plus function x_+ is defined as $(x_+)_i = \max\ \{0, x_i\}$, while the step function x_* is defined as $(x_*)_i = 1$ if $x_i > 0$ else $(x_*)_i = 0$, $i = 1, \ldots, n$. The scalar (inner) product of two vectors x and y in the n-dimensional real space R^n will be denoted by $x'y$ and the p-norm of x will be denoted by $\|x\|_p$. For a matrix $A \in R^{m\times n}$, A_i is the ith row of A which is a row vector in R^n. A column vector of ones of arbitrary dimension will be denoted by e. For $A \in R^{m\times n}$ and $B \in R^{n\times l}$, the kernel $K(A, B)$ maps $R^{m\times n} \times R^{n\times l}$ into $R^{m\times l}$. In particular, if x and y are column vectors in R^n then, $K(x', y)$ is a real number, $K(x', A')$ is a row vector in R^m and $K(A, A')$ is an $m \times m$ matrix. The base of the natural logarithm will be denoted by ε.

2 Linear and Nonlinear Kernel Classification

We consider the problem 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 classes +1 or -1 as specified by a given $m \times m$ diagonal matrix D with ones or minus ones along its diagonal. For this problem the standard support vector machine with a linear kernel AA' [23, 6] is given by the following quadratic program for some $\nu > 0$:

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

As depicted in Figure 1, w is the normal to the bounding planes:

$$\begin{array}{lcccl} x'w & - & \gamma & = & +1 \\ x'w & - & \gamma & = & -1, \end{array} \tag{2}$$

and γ determines their location relative to the origin. The first plane above bounds the class +1 points and the second plane bounds the class -1 points when the two classes are strictly linearly separable, that is when the slack variable $y = 0$. The linear separating surface is the plane

$$x'w = \gamma, \tag{3}$$

midway between the bounding planes (2). If the classes are linearly inseparable then the two planes bound the two classes with a "soft margin" determined by a nonnegative slack variable y, that is:

$$\begin{array}{lcccccl} x'w & - & \gamma & + & y_i & \geq & +1, \text{ for } x' = A_i \text{ and } D_{ii} = +1, \\ x'w & - & \gamma & - & y_i & \leq & -1, \text{ for } x' = A_i \text{ and } D_{ii} = -1. \end{array} \tag{4}$$

The 1-norm of the slack variable y is minimized with weight ν in (1). 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 (2) in the n-dimensional space of $w \in R^n$ for a *fixed* γ, maximizes that distance, often called the "margin". Figure 1 depicts the points represented by A, the bounding planes (2) with margin $\frac{2}{\|w\|_2}$, and the separating plane (3) which separates $A+$, the points represented by rows of A with $D_{ii} = +1$, from $A-$, the points represented by rows of A with $D_{ii} = -1$.

In our smooth approach, the square of 2-norm of the slack variable y is minimized with weight $\frac{\nu}{2}$ instead of the 1-norm of y as in (1). In addition the distance between the planes (2) is measured in the $(n+1)$-dimensional space of $(w, \gamma) \in R^{n+1}$, that is $\frac{2}{\|(w,\gamma)\|_2}$. Measuring the margin in this $(n+1)$-dimensional space instead of R^n induces strong convexity and has little or no effect on the problem as was shown in [14]. Thus using twice the reciprocal squared of the margin instead, yields our modified SVM problem as follows:

$$\begin{array}{rrcl} \min\limits_{(w,\gamma,y)\in R^{n+1+m}} & \frac{\nu}{2}y'y + \frac{1}{2}(w'w + \gamma^2) & & \\ \text{s.t.} & D(Aw - e\gamma) + y & \geq & e \\ & y & \geq & 0. \end{array} \tag{5}$$

It was shown computationally in [15] that this reformulation (5) of the conventional support vector machine formulation (1) yields similar results to (1). At a solution of problem (5), y is given by

$$y = (e - D(Aw - e\gamma))_+, \tag{6}$$

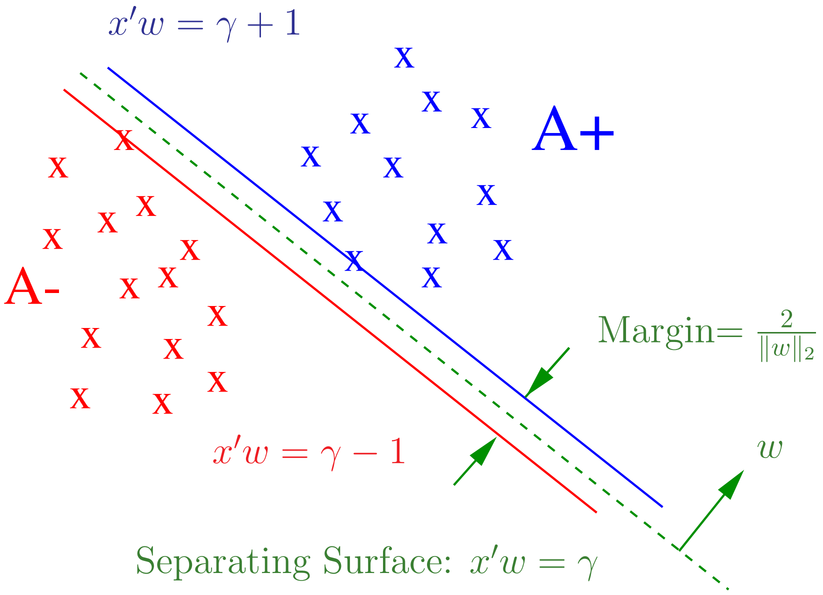

Figure 1. **The bounding planes (2) with margin $\frac{2}{\|w\|_2}$, and the plane (3) separating $A+$, the points represented by rows of A with $D_{ii} = +1$, from $A-$, the points represented by rows of A with $D_{ii} = -1$.**

where, as defined in the Introduction, $(\cdot)_+$ replaces negative components of a vector by zeros. Thus, we can replace y in (5) by $(e - D(Aw - e\gamma))_+$ and convert the SVM problem (5) into an equivalent SVM which is an unconstrained optimization problem as follows:

$$\min_{(w,\gamma)\in R^{n+1}} \quad \frac{\nu}{2}\|(e - D(Aw - e\gamma))_+\|_2^2 + \frac{1}{2}(w'w + \gamma^2). \tag{7}$$

This problem is a strongly convex minimization problem without any constraints. It is easy to show that it has a unique solution. However, the objective function in (7) is not twice differentiable which precludes the use of a fast Newton method. In [10] we smoothed this problem and applied a fast Newton method to solve it as well as the nonlinear kernel problem which we describe now.

We first describe how the generalized support vector machine (GSVM) [12] generates a nonlinear separating surface by using a completely arbitrary kernel. The GSVM solves the following mathematical program for a general kernel $K(A, A')$:

$$\begin{array}{lrcl} \displaystyle\min_{(u,\gamma,y)\in R^{2m+1}} & \nu e'y + f(u) & & \\ \text{s.t.} & D(K(A,A')Du - e\gamma) + y & \geq & e \\ & y & \geq & 0. \end{array} \tag{8}$$

Here $f(u)$ is some convex function on R^m which suppresses the parameter u and ν is some positive number that weights the classification error $e'y$ versus the suppression of u. A solution of this mathematical program for u and γ leads to the nonlinear separating surface

$$K(x', A')Du = \gamma. \tag{9}$$

The linear formulation (1) of Section 2 is obtained if we let $K(A, A') = AA'$, $w = A'Du$ and $f(u) = \frac{1}{2}u'DAA'Du$. We now use a different classification objective which not only suppresses the parameter u but also suppresses γ in our nonlinear formulation:

$$\begin{array}{rrcl} \displaystyle\min_{(u,\gamma,y)\in R^{2m+1}} & \frac{\nu}{2}y'y + \frac{1}{2}(u'u + \gamma^2) & & \\ \text{s.t.} & D(K(A, A')Du - e\gamma) + y & \geq & e \\ & y & \geq & 0. \end{array} \tag{10}$$

At a solution of (10), y is given by

$$y = (e - D(K(A, A')Du - e\gamma))_+, \tag{11}$$

where, as defined earlier, $(\cdot)_+$ replaces negative components of a vector by zeros. Thus, we can replace y in (10) by $(e - D(K(A, A')Du - e\gamma))_+$ and convert the SVM problem (10) into an equivalent SVM which is an unconstrained optimization problem as follows:

$$\min_{(u,\gamma)\in R^{m+1}} \quad \frac{\nu}{2}\|(e - D(K(A, A')Du - e\gamma))_+\|_2^2 + \frac{1}{2}(u'u + \gamma^2). \tag{12}$$

Again, as in (7), this problem is a strongly convex minimization problem without any constraints, has a unique solution but its objective function is not twice differentiable. To apply a fast Newton method we use the smoothing techniques of [4, 5] and replace x_+ by a very accurate smooth approximation as was done in [10]. Thus we replace x_+ by $p(x, \alpha)$, the integral of the sigmoid function $\frac{1}{1+\varepsilon^{-\alpha x}}$ of neural networks [11, 4] for some $\alpha > 0$. That is:

$$p(x, \alpha) = x + \frac{1}{\alpha}\log(1 + \varepsilon^{-\alpha x}),\ \alpha > 0. \tag{13}$$

This p function with a smoothing parameter α is used here to replace the plus function of (12) to obtain a smooth support vector machine **(SSVM)** :

$$\min_{(u,\gamma)\in R^{m+1}} \frac{\nu}{2}\|p(e - D(K(A, A')Du - e\gamma), \alpha)\|_2^2 + \frac{1}{2}(u'u + \gamma^2). \tag{14}$$

It was shown in [10] that the solution of problem (10) is obtained by solving problem (14) with α approaching infinity. Computationally, we used the limit values of the sigmoid function $\frac{1}{1+\varepsilon^{-\alpha x}}$ and the p function (13) as the smoothing parameter α approaches infinity, that is the unit step function with value $\frac{1}{2}$ at zero and the plus function $(\cdot)_+$ respectively. This gave extremely good results both here and in [10]. The twice differentiable property of the objective function of (14) enables us to utilize a globally quadratically convergent Newton algorithm for solving the smooth support vector machine (14) [10, Algorithm 3.1] which consists of solving successive linearizations of the gradient of the objective function set to zero. Problem (14) which is capable of generating a highly nonlinear separating surface (9), retains the strong convexity and differentiability properties for any arbitrary kernel. However, we still have to contend with two difficulties. Firstly, problem (14) is a problem in $m+1$ variables, where m could be of the order of millions for large datasets. Secondly, the resulting nonlinear separating surface (9) depends on the entire dataset represented by the matrix A. This creates an unwieldy storage difficulty for very large datasets and makes the use of nonlinear kernels impractical for such problems. To avoid these two difficulties we turn our attention to the reduced support vector machine.

3 RSVM: The Reduced Support Vector Machine

The motivation for RSVM comes from the practical objective of generating a nonlinear separating surface (9) for a large dataset which requires a small portion of the dataset for its characterization. The difficulty in using nonlinear kernels on large datasets is twofold. First is the computational difficulty in solving the the potentially huge unconstrained optimization problem (14) which involves the kernel function $K(A, A')$ that typically leads to the computer running out of memory even before beginning the solution process. For example for the Adult dataset with 32562 points, which is actually solved with RSVM in Section 4, this would mean a map into a space of over one billion dimensions for a conventional SVM. The second difficulty comes from utilizing the formula (9) for the separating surface on a new unseen point x. The formula dictates that we store and utilize the entire data set represented by the 32562×123 matrix A which may be prohibitively expensive storage-wise and computing-time-wise. For example for the Adult dataset just mentioned which has an input space of 123 dimensions, this would mean that the nonlinear surface (9) requires a storage capacity for 4,005,126 numbers. To avoid all these difficulties and based on experience with chunking methods [2, 13], we hit upon the idea of using a very small random subset of the dataset given by $\bar{m}$ points of the original m data points with $\bar{m} << m$, that we call $\bar{A}$ and use $\bar{A}'$ in place of A' in *both* the unconstrained optimization problem (14), to cut problem size and computation time, and for the same purposes in evaluating the nonlinear surface (9). Note that the matrix A is left intact in $K(A, \bar{A}')$. Computational testing results show a standard deviation of 0.002 or less of test set correctness over 50 random choices for $\bar{A}$. By contrast if *both* A and A' are replaced by $\bar{A}$ and $\bar{A}'$ respectively, then test set correctness declines substantially compared to RSVM, while the standard deviation of test set correctness over 50 cases increases more than tenfold over that of RSVM.

The justification for our proposed approach is this. We use a small random $\bar{A}$ sample of our dataset as a representative sample with respect to the *entire* dataset A both in solving the optimization problem (14) and in evaluating the the nonlinear separating surface (9). We interpret this as a possible instance-based learning [17, Chapter 8] where the small sample $\bar{A}$ is learning from the much larger training set A by forming the appropriate rectangular kernel relationship $K(A, \bar{A}')$ between the original and reduced sets. This formulation works extremely well computationally as evidenced by the computational results that we present in the next section of the paper.

By using the formulations described in Section 2 for the full dataset $A \in R^{m\times n}$ with a square kernel $K(A, A') \in R^{m\times m}$, and modifying these formulations for the reduced dataset $\bar{A} \in R^{\bar{m}\times n}$ with corresponding diagonal matrix $\bar{D}$ and rectangular kernel $K(A, \bar{A}') \in R^{m\times\bar{m}}$, we obtain our RSVM Algorithm below. This algorithm solves, by smoothing, the RSVM quadratic program obtained from (10) by replacing A' with $\bar{A}'$ as follows:

$$\begin{array}{rrcl} \displaystyle\min_{(\bar{u},\gamma,y)\in R^{\bar{m}+1+m}} & \frac{\nu}{2}y'y + \frac{1}{2}(\bar{u}'\bar{u} + \gamma^2) & & \\ \text{s.t.} & D(K(A,\bar{A}')\bar{D}\bar{u} - e\gamma) + y & \geq & e \\ & y & \geq & 0. \end{array} \tag{15}$$

Algorithm 3.1 RSVM Algorithm

(i) Choose a random subset matrix $\bar{A} \in R^{\bar{m}\times n}$ of the original data matrix $A \in R^{m\times n}$. Typically $\bar{m}$ is 1% to 10% of m. (The random matrix $\bar{A}$ choice was such that the distance between its rows exceeded a certain tolerance.)

(ii) Solve the following modified version of the SSVM (14) where A' **only** is replaced by $\bar{A}'$ with corresponding $\bar{D} \subset D$:

$$\min_{(\bar{u},\gamma)\in R^{\bar{m}+1}} \frac{\nu}{2}\|p(e - D(K(A,\bar{A}')\bar{D}\bar{u} - e\gamma), \alpha)\|_2^2 + \frac{1}{2}(\bar{u}'\bar{u} + \gamma^2), \tag{16}$$

which is equivalent to solving (10) with A' **only** replaced by $\bar{A}'$.

(iii) The separating surface is given by (9) with A' replaced by $\bar{A}'$ as follows:

$$K(x', \bar{A}')\bar{D}\bar{u} = \gamma, \tag{17}$$

where $(\bar{u}, \gamma) \in R^{\bar{m}+1}$ is the unique solution of (16), and $x \in R^n$ is a free input space variable of a new point.

(iv) A new input point $x \in R^n$ is classified into class $+1$ or -1 depending on whether the step function:

$$(K(x', \bar{A}')\bar{D}\bar{u} - \gamma)_*, \tag{18}$$

is $+1$ or zero, respectively.

As stated earlier, this algorithm is quite insensitive as to which submatrix $\bar{A}$ is chosen for (16)-(17), as far as tenfold cross-validation correctness is concerned. In fact, another choice for $\bar{A}$ is to choose it randomly but only keep rows that are more than a certain minimal distance apart. This leads to a slight improvement in testing correctness but increases computational time somewhat. Replacing *both* A and A' in a conventional SVM by randomly chosen reduced matrices $\bar{A}$ and $\bar{A}'$ gives poor testing set results that vary significantly with the choice of $\bar{A}$, as will be demonstrated in the numerical results given in the next section to which we turn now.

4 Computational Results

We applied RSVM to three groups of publicly available test problems: the checkerboard problem [8, 9], six test problems from the University of California (UC) Irvine repository [18] and the Adult data set from the same repository. We show that RSVM performs better than a conventional SVM using the *entire* training set and much better than a *conventional* SVM using only the *same* randomly chosen set by RSVM. We also show, using time comparisons, that RSVM performs better than sequential minimal optimization (SMO) [19] and projected conjugate gradient chunking (PCGC) [7, 3]. Computational time on the Adult datasets grows nearly linearly for RSVM, whereas SMO and PCGC times grow at a much faster nonlinear rate. All our experiments were solved by using the globally quadratically convergent smooth support vector machine (SSVM) algorithm [10] that merely solves a finite sequence of systems of linear equations defined by a positive definite Hessian matrix to get a Newton direction at each iteration. Typically 5 to 8 systems of linear equations are solved by SSVM and hence each data point $A_i,\ i = 1, \ldots, m$ is accessed 5 to 8 times by SSVM. Note that no special optimization packages such as linear or quadratic programming solvers are needed. We implemented SSVM using standard native MATLAB commands [16]. We used a Gaussian kernel [12]: $\varepsilon^{-\alpha\|A_i - A_j\|_2^2}$, $i, j = 1, \ldots, m$ for all our numerical tests. A polynomial kernel of degree 6 was also used on the checkerboard with similar results which are not reported here. All parameters in these tests were chosen for optimal performance on a tuning set, a surrogate for a test set. All our experiments were run on the University of Wisconsin Computer Sciences Department Ironsides cluster. This cluster of four Sun Enterprise E6000 machines, each machine consisting of 16 UltraSPARC II 250 MHz processors and 2 gigabytes of RAM, resulting in a total of 64 processors and 8 gigabytes of RAM.

The checkerboard dataset [8, 9] consists of 1000 points in R^2 of black and white points taken from sixteen black and white squares of a checkerboard. This dataset is chosen in order to depict graphically the effectiveness of RSVM using a random 5% or 10% of the given 1000-point training dataset compared to the very poor performance of a conventional SVM on the same 5% or 10% randomly chosen subset. Figures 2 and 4 show the poor pattern approximating a checkerboard obtained by a conventional SVM using a Gaussian kernel, that is solving (10) with *both* A and A' replaced by the randomly chosen $\bar{A}$ and $\bar{A}'$ respectively. Test set correctness of this conventional SVM using the reduced $\bar{A}$ and $\bar{A}'$ averaged, over 15 cases, 43.60% for the 50-point dataset and 67.91% for the 100-point dataset, on a test set of 39601 points. In contrast, using our RSVM Algorithm 3.1 on the *same* randomly chosen submatrices $\bar{A}'$, yields the much more accurate representations of the checkerboard depicted in Figures 3 and 5 with corresponding average test set correctness of 96.70% and 97.55% on the same test set.

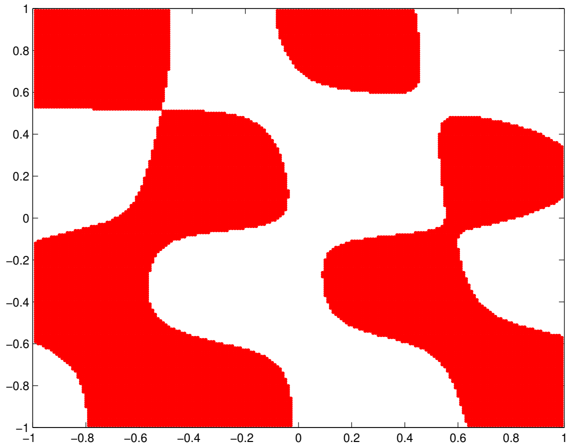

Figure 2. SVM: Checkerboard resulting from a randomly selected 50 points, out of a 1000-point dataset, and used in a conventional Gaussian kernel SVM (10). The resulting nonlinear surface, separating white and black areas, generated using the 50 random points only, depends explicitly on those points only. Correctness on a 39601-point test set averaged 43.60% on 15 randomly chosen 50-point sets, with a standard deviation of 0.0895 and best correctness of 61.03% depicted above.

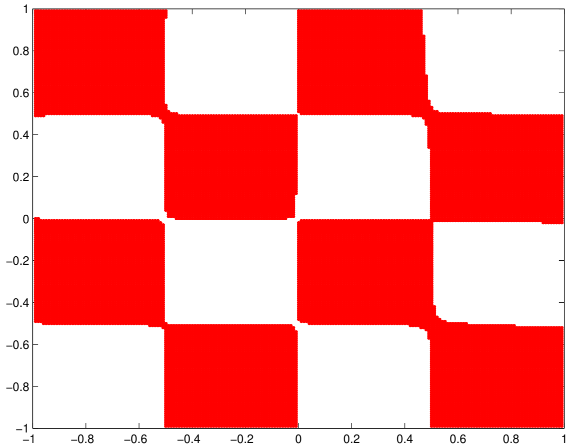

Figure 3. RSVM: Checkerboard resulting from randomly selected 50 points and used in a reduced Gaussian kernel SVM (15). The resulting nonlinear surface, separating white and black areas, generated using the entire 1000-point dataset, depends explicitly on the 50 points only. The remaining 950 points can be thrown away once the separating surface has been generated. Correctness on a 39601-point test set averaged 96.7% on 15 randomly chosen 50-point sets, with a standard deviation of 0.0082 and best correctness of 98.04% depicted above.

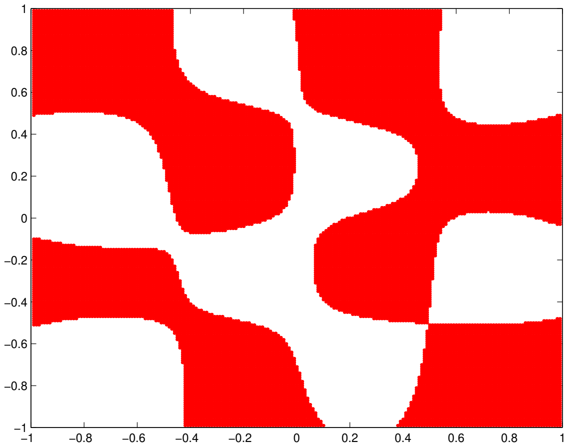

Figure 4. SVM: Checkerboard resulting from a randomly selected 100 points, out of a 1000-point dataset, and used in a conventional Gaussian kernel SVM (10). The resulting nonlinear surface, separating white and black areas, generated using the 100 random points only, depends explicitly on those points only. Correctness on a 39601-point test set averaged 67.91% on 15 randomly chosen 100-point sets, with a standard deviation of 0.0378 and best correctness of 76.09% depicted above.

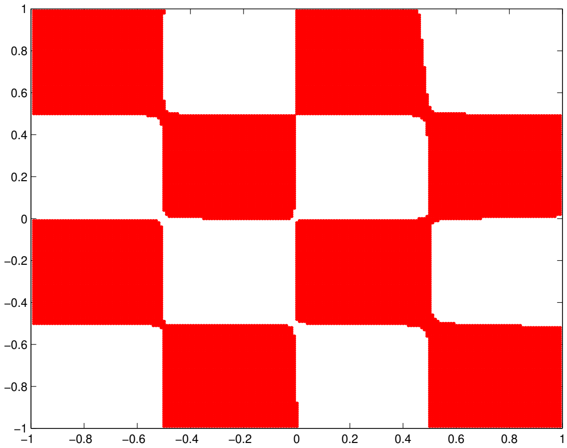

Figure 5. RSVM: Checkerboard resulting from randomly selected 100 points and used in a reduced Gaussian kernel SVM (15). The resulting nonlinear surface, separating white and black areas, generated using the entire 1000-point dataset, depends explicitly on the 100 points only. The remaining 900 points can be thrown away once the separating surface has been generated. Correctness on a 39601-point test set averaged 97.55% on 15 randomly chosen 100-point sets, with a standard deviation of 0.0034 and best correctness of 98.26% depicted above.

The next set of numerical results in Table 1 on the six UC Irvine test problems: Ionosphere, BUPA Liver, Pima Indians, Cleveland Heart, Tic-Tac-Toe and Mushroom, show that RSVM, with $\bar{m} \leq \frac{m}{10}$ on all these datasets, got better test set correctness than that of a conventional SVM (10) using the full data matrix A and much better than the conventional SVM (10) using the same reduced matrices $\bar{A}$ and $\bar{A}'$. RSVM was also better than the linear SVM using the full data matrix A. A possible reason for the improved test set correctness of RSVM is the avoidance of data overfitting by using a reduced data matrix $\bar{A}'$ instead of the full data matrix A'.

Tenfold Test Set Correctness % (**Best in Bold**) Tenfold Computational Time, *Seconds*				
	Gaussian Kernel Matrix Used in SSVM			
Dataset Size $m \times n, \quad \bar{m}$	$K(A, \bar{A}')$ $m \times \bar{m}$	$K(A, A')$ $m \times m$	$K(\bar{A}, \bar{A}')$ $\bar{m} \times \bar{m}$	AA' (Linear) $m \times n$
Cleveland Heart $297 \times 13, \quad 30$	**86.47** 3.04	85.92 32.42	76.88 1.58	86.13 1.63
BUPA Liver $345 \times 6, \quad 35$	**74.86** 2.68	73.62 32.61	68.95 2.04	70.33 1.05
Ionosphere $351 \times 34, \quad 35$	**95.19** 5.02	94.35 59.88	88.70 2.13	89.63 3.69
Pima Indians $768 \times 8, \quad 50$	**78.64** 5.72	76.59 328.3	57.32 4.64	78.12 1.54
Tic-Tac-Toe $958 \times 9, \quad 96$	**98.75** 14.56	98.43 1033.5	88.24 8.87	69.21 0.68
Mushroom $8124 \times 22, \quad 215$	**89.04** 466.20	N/A N/A	83.90 221.50	81.56 11.27

Table 1. Tenfold cross-validation correctness results on six UC Irvine datasets demonstrate that the RSVM Algorithm 3.1 can get test set correctness that is better than a conventional nonlinear SVM (10) using either the full data matrix A or the reduced matrix $\bar{A}'$, as well as a linear kernel SVM using the full data matrix A. The computer ran out of memory while generating the full nonlinear kernel for the Mushroom dataset. Average on these six datasets of the standard deviation of the tenfold test set correctness for $K(A, \bar{A}')$ was 0.034 and for $K(\bar{A}, \bar{A}')$ was 0.057. N/A denotes "not available" results because the kernel $K(A, A')$ was too large to store.

The third group of test problems, the UCI Adult dataset, uses an $\bar{m}$ that ranges between 1% to 5% of m in the RSVM Algorithm 3.1. We make the following observations on this set of results given in Table 2:

(i) Test set correctness of RSVM was better on average by 10.52% and by as much as 12.52% over a conventional SVM using the same reduced submatrices $\bar{A}$ and

$\bar{A}'$.

(ii) The standard deviation of test set correctness for 50 randomly chosen $\bar{A}'$ for RSVM was no greater than 0.002, while the corresponding standard deviation for a conventional SVM for the same 50 random $\bar{A}$ and $\bar{A}'$ was as large as 0.026. In fact, smallness of the standard deviation was used as a guide to determining $\bar{m}$, the size of the reduced data used in RSVM.

Adult Dataset Size	$K(A, \bar{A}')_{m\times\bar{m}}$		$K(\bar{A}, \bar{A}')_{\bar{m}\times\bar{m}}$		$\bar{A}_{\bar{m}\times 123}$	
(Training, Testing)	Testing %	Std. Dev.	Testing %	Std. Dev.	$\bar{m}$	$\bar{m}/m$
(1605, 30957)	84.29	0.001	77.93	0.016	81	5.0 %
(2265, 30297)	83.88	0.002	74.64	0.026	114	5.0 %
(3185, 29377)	84.56	0.001	77.74	0.016	160	5.0 %
(4781, 27781)	84.55	0.001	76.93	0.016	192	4.0 %
(6414, 26148)	84.47	0.001	77.03	0.014	210	3.2 %
(11221, 21341)	84.71	0.001	75.96	0.016	225	2.0 %
(16101, 16461)	84.90	0.001	75.45	0.017	242	1.5 %
(22697, 9865)	85.31	0.001	76.73	0.018	284	1.2 %
(32562, 16282)	85.07	0.001	76.95	0.013	326	1.0 %

Table 2. Computational results for 50 runs of RSVM on each of nine commonly used subsets of the Adult dataset [18]. Each run uses a randomly chosen $\bar{A}$ from A for use in an RSVM Gaussian kernel, with the number of rows $\bar{m}$ of $\bar{A}$ between 1% and 5% of the number of rows m of the full data matrix A. Test set correctness for the largest case is the same as that of SMO [20].

Finally, Table 3 and Figure 6 show the nearly linear time growth of RSVM on the Adult dataset as a function of the number of points m in the dataset, compared to the faster nonlinear time growth of SMO [19] and PCGC [7, 3].

5 Conclusion

We have proposed a Reduced Support Vector Machine (RSVM) Algorithm 3.1 that uses a randomly selected subset of the data that is typically 10% or less of the original dataset to obtain a nonlinear separating surface. Despite this reduced dataset, RSVM gets better test set results than that obtained by using the entire data. This may be attributable to a reduction in data overfitting. The reduced dataset is all that is needed in characterizing the final nonlinear separating surface. This is very important for massive datasets such as those used in fraud detection which number in the millions. We may think that all the information in the discarded data has

Adult Datasets - Training Set Size *vs.* CPU Time in *Seconds*									
Size	1605	2265	3185	4781	6414	11221	16101	22697	32562
RSVM	10.1	20.6	44.2	83.6	123.4	227.8	342.5	587.4	980.2
SMO	15.8	32.1	66.2	146.6	258. 8	781.4	1784.4	4126.4	7749.6
PCGC	34.8	114.7	380.5	1137.2	2530.6	11910.6	N/A	N/A	N/A

Table 3. CPU time comparisons of RSVM, SMO [19] and PCGC [7, 3] with a Gaussian kernel on the Adult datasets. SMO and PCGC were run on a 266 MHz Pentium II processor under Windows NT 4 and using Microsoft's Visual C++ 5.0 compiler. PCGC ran out of memory (128 Megabytes) while generating the kernel matrix when the training set size is bigger than 11221. We quote results from [19]. N/A denotes "not available" results because the kernel $K(A, A')$ was too large to store.

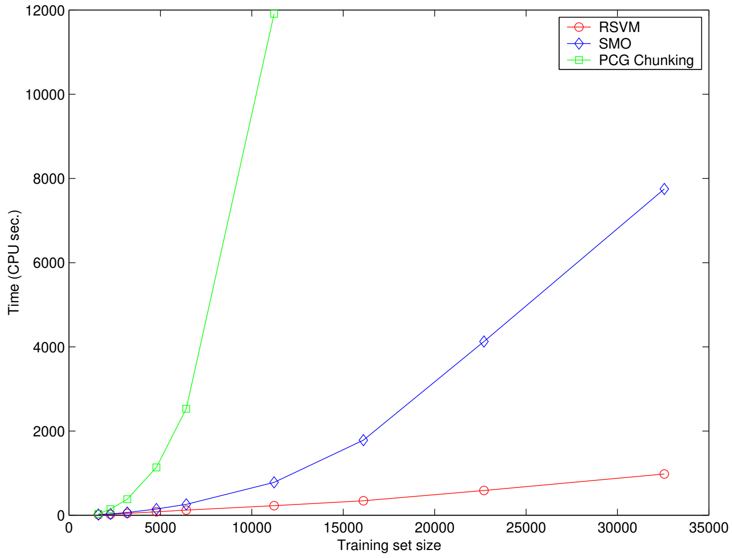

Figure 6. Indirect CPU time comparison of RSVM, SMO and PCGC for a Gaussian kernel SVM on the nine Adult data subsets.

been distilled into the parameters defining the nonlinear surface during the training process via the rectangular kernel $K(A, \bar{A}')$. Although the training process, which consists of the RSVM Algorithm 3.1, uses the entire dataset in an unconstrained optimization problem (14), it is a problem in $R^{\bar{m}+1}$ with $\bar{m} \leq \frac{m}{10}$, and hence much easier to solve than that for the full dataset which would be a problem in R^{m+1}. The choice of the random data submatrix $\bar{A}'$ to be used in RSVM does not affect test set correctness. In contrast, a random choice for a data submatrix for a conventional SVM has standard deviation of test set correctness which is more than ten times that of RSVM. With all these properties, RSVM appears to be a very promising method for handling large classification problems using a nonlinear separating surface.

Acknowledgements

The research described in this Data Mining Institute Report 00-07, July 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. We thank Paul S. Bradley for valuable comments and David R. Musicant for his Gaussian kernel generator.

Bibliography

[1] P. S. Bradley and O. L. Mangasarian, *Feature selection via concave minimization and support vector machines*, in Machine Learning Proceedings of the Fifteenth International Conference(ICML '98), J. Shavlik, editor, Morgan Kaufmann, San Francisco, California, 1998, pp. 82–90.

[2] P. S. Bradley and O. L. Mangasarian, *Massive data discrimination via linear support vector machines*, Optimization Methods and Software, 13 (2000), pp. 1–10.

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

[4] Chunhui Chen and O. L. Mangasarian, *Smoothing methods for convex inequalities and linear complementarity problems*, Mathematical Programming 71(1) (1995), pp. 51–69.

[5] ———, *A class of smoothing functions for nonlinear and mixed complementarity problems*, Computational Optimization and Applications, 5(2) (1996), pp. 97–138.

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

[7] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, 1981.

[8] T. K. Ho and E. M. Kleinberg, *Building projectable classifiers of arbitrary complexity*, in Proceedings of the 13th International Conference on Pattern Recognition, Vienna, Austria, 1996, pp. 880–885. http://cm.bell-labs.com/who/tkh/pubs.html. Checker dataset at: ftp://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/checker.

[9] L. Kaufman, *Solving the quadratic programming problem arising in support vector classification*, in Advances in Kernel Methods - Support Vector Learning, B. Schölkopf, C. J. C. Burges, and A. J. Smola, eds., MIT Press, 1999, pp. 147–167.

[10] Yuh-Jye Lee and O. L. Mangasarian, SSVM*: A smooth support vector machine*, Technical Report 99-03, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, September 1999. Computational Optimization and Applications, to appear. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/99-03.ps.

[11] O. L. Mangasarian, *Mathematical programming in neural networks*, ORSA Journal on Computing, 5(4) (1993), pp. 349–360.

[12] ——, *Generalized support vector machines*, in Advances in Large Margin Classifiers, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, eds., MIT Press,Cambridge, MA, 2000, pp. 135–146.

[13] O. L. Mangasarian and D. R. Musicant, *Massive support vector regression*, Technical Report 99-02, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, July 1999. Machine Learning, to appear. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/99-02.ps.

[14] ——, *Successive overrelaxation for support vector machines*, IEEE Transactions on Neural Networks, 10 (1999), pp. 1032–1037.

[15] ——, *Lagrangian support vector machines*, Technical Report 00-06, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, June 2000. Journal of Machine Learning Research, to appear. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/00-06.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] J. Platt, *Sequential minimal optimization: A fast algorithm for training support vector machines*, in Advances in Kernel Methods - Support Vector Learning, B. Schölkopf, C. J. C. Burges, and A. J. Smola, eds., MIT Press, 1999, pp. 185–208. http://www.research.microsoft.com/∼jplatt/smo.html.

[20] ——, *Personal communication*, May 2000.

[21] B. Schölkopf, C. Burges, and A. Smola (editors), *Advances in Kernel Methods: Support Vector Machines*, MIT Press, Cambridge, MA, 1998.

[22] A. Smola, P. L. Bartlett, B. Schölkopf, and J. Schürmann (editors), *Advances in Large Margin Classifiers*, MIT Press, Cambridge, MA, 2000.

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

[24] C. K. I. Williams and M. Seeger, *Using the Nyström method to speed up kernel machines*, in Advances in Neural Information Processing Systems (NIPS2000), 2000, to appear. http://www.kernel-machines.org.