Privacy-Preserving Classification of Vertically Partitioned Data via Random Kernels

Olvi L. Mangasarian[*] Edward W. Wild[†] Glenn M. Fung[‡]

Abstract

We propose a novel privacy-preserving support vector machine (SVM) classifier for a data matrix A whose input feature columns are divided into groups belonging to different entities. Each entity is unwilling to share its group of columns or make it public. Our classifier is based on the concept of a reduced kernel $K(A,B')$ where B' is the transpose of a random matrix B. The column blocks of B corresponding to the different entities are privately generated by each entity and never made public. The proposed linear or nonlinear SVM classifier, which is public but does not reveal any of the privately-held data, has accuracy comparable to that of an ordinary SVM classifier that uses the entire set of input features directly.

Keywords: privacy preserving classification, support vector machines, vertically partitioned data

1 INTRODUCTION

Recently there has been wide interest in privacy-preserving support vector machine (SVM) classifiers. Basically the problem revolves around generating a classifier based on data, parts of which are held by private entities who for various reasons are unwilling to make it public. When each entity holds its own group of input feature values for all individuals while other entities hold other groups of feature values for the same individuals, the data is referred to as *vertically partitioned*. This is so because feature values are represented by columns of a data matrix while individuals are represented by rows of the data matrix. In [24], privacy-preserving SVM classifiers were obtained for vertically partitioned data by adding random perturbations to the data. In [22, 23], *horizontally partitioned* privacy-preserving SVMs and induction tree classifiers were obtained for data where different entities hold the same input features for different groups of individuals. Other privacy preserving classifying techniques include cryptographically private SVMs [18], wavelet-based distortion [13] and rotation perturbation [3].

In this work we propose a highly efficient privacy-preserving SVM (PPSVM) classifier for vertically partitioned data that is different from existing SVM classifiers and is based on the following two ideas. For a given data matrix $A \in R^{m\times n}$, instead of using the usual kernel function $K(A,A') : R^{m\times n} \times R^{n\times m} \longrightarrow R^{m\times m}$, we use a *reduced* kernel [12, 11] $K(A,B') : R^{m\times n} \times R^{n\times \bar{m}} \longrightarrow R^{m\times \bar{m}}$, $\bar{m} < m$, where B is a completely random matrix. The second idea is that the $\bar{m}$ columns of the random matrix $B \in R^{\bar{m}\times n}$ are privately generated in p blocks corresponding to the p entities holding the p blocks of input features. Each random column block of the matrix B is generated by only one of the p entities, and that block of the random B is known only to the entity that generated it and never made public. By employing these two ideas, we shall describe an algorithm that completely protects the privacy of each vertical partition of the data matrix A, owned by a distinct entity, while generating an SVM classifier with ten-fold cross validation accuracy comparable to that of an ordinary SVM classifier that utilizes the whole data in one shot.

[*] Computer Sciences Department, University of Wisconsin, Madison, WI 53706 and Department of Mathematics, University of California at San Diego, La Jolla, CA 92093.*olvi@cs.wisc.edu*.

[†] Computer Sciences Department, University of Wisconsin, Madison, WI 53706. *wildt@cs.wisc.edu*.

[‡] Siemens Medical Solutions, Inc., 51 Valley Stream Parkway, Malvern, PA 19355. *glenn.fung@siemens.com*.

We now briefly describe the contents of the paper. In Section 2 we describe our method for a privacy-protecting linear SVM classifier for vertically partitioned data, and in Section 3 do the same for a nonlinear SVM classifier. In Section 4 we give computational results that show the effectiveness of our approach, including correctness that is comparable to ordinary SVMs. Section 5 concludes the paper with a summary and some ideas for future work.

We describe our notation now. All vectors will be column vectors unless transposed to a row vector by a prime $'$. For a vector $x \in R^n$ the notation x_j will signify either the j-th component or j-th block of components. The scalar (inner) product of two vectors x and y in the n-dimensional real space R^n will be denoted by $x'y$. For $x \in R^n$, $\|x\|_1$ denotes the 1-norm: $(\sum\limits_{i=1}^{n}|x_i|)$ while $\|x\|$ denotes the 2-norm: $(\sum\limits_{i=1}^{n}(x_i)^2)^{\frac{1}{2}}$. 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, A_i will denote the i-th row or i-th block of rows of A and $A_{\cdot j}$ the j-th column or the j-th block of columns of A. A vector of ones in a real space of arbitrary dimension will be denoted by e. Thus for $e \in R^m$ and $y \in R^m$ the notation $e'y$ will denote the sum of the components of y. A vector of zeros in a real space of arbitrary dimension will be denoted by 0. For $A \in R^{m\times n}$ and $B \in R^{k\times n}$, a *kernel* $K(A,B')$ maps $R^{m\times n} \times R^{n\times k}$ into $R^{m\times k}$. In particular, if x and y are column vectors in R^n then, $K(x',y)$ is a real number, $K(x',B')$ is a row vector in R^k and $K(A,B')$ is an $m \times k$ matrix. The base of the natural logarithm will be denoted by ε. A frequently used kernel in nonlinear classification is the Gaussian kernel [20, 19, 14] whose ij-th element, $i = 1,\ldots\ldots,m,\ j = 1,\ldots\ldots,k$, is given by: $(K(A,B'))_{ij} = \varepsilon^{-\mu\|A_i-B_{\cdot j}'\|^2}$, where $A \in R^{m\times n}$, $B \in R^{k\times n}$ and μ is a positive constant. We shall not assume that our kernels satisfy Mercer's positive definiteness condition [20, 19, 4], however we shall assume that they are separable in the following sense:

$$K([E\ \ F],[G\ \ H]') = K(E,G') + K(F,H') \ \text{ or } \ K([E\ \ F],[G\ \ H]') = K(E,G') \odot K(F,H'), \tag{1.1}$$

where the symbol $\odot$ denotes the Hadamard component-wise product of two matrices of the same dimensions [8], $E \in R^{m\times n_1}$, $F \in R^{m\times n_2}$, $G \in R^{k\times n_1}$ and $H \in R^{k\times n_2}$. It is straightforward to show that a linear kernel $K(A,B') = AB'$ satisfies (1.1) with the $+$ sign, and a Gaussian kernel satisfies (1.1) with the $\odot$ sign. The abbreviation "s.t." stands for "subject to".

2 Privacy-Preserving Linear Classifier for Vertically Partitioned Data

The dataset that we wish to obtain a classifier for consists of m points in R^n represented by the m rows of the matrix $A \in R^{m\times n}$. The matrix A is divided into p vertical blocks of $n_1, n_2, \ldots\ldots$ and n_p columns with $n_1 + n_2 + \ldots + n_p = n$. Each block of columns is "owned" by an entity that is unwilling to make it public or share it with the other entities. Furthermore, each row of A is labeled as belonging to the class $+1$ or -1 by a corresponding diagonal matrix $D \in R^{m\times m}$ of ±1's. The linear kernel classifier to be generated based on this data will be a separating plane in R^n:

$$x'w - \gamma = x'B'u - \gamma = 0, \tag{2.2}$$

which classifies a given point x according to the sign of $x'w - \gamma$. Here, $w = B'u$, $w \in R^n$ is the normal to the plane $x'w - \gamma = 0$, $\gamma \in R$ determines the distance of the plane from the origin in R^n and B is a random matrix in $R^{k\times n}$. The change of variables $w = B'u$ is employed in order to kernelize the data and is motivated by the fact that when $B = A$ and hence $w = A'u$, the variable u is the dual variable for a 2-norm SVM [14]. The variables $u \in R^k$ and $\gamma \in R$ are to be determined by an optimization problem such that the labeled data A satisfy, to the extent possible, the separation condition:

$$D(AB'u - e\gamma) \geq 0. \tag{2.3}$$

This condition (2.3) places the $+1$ and -1 points represented by A on opposite sides of the separating plane (2.2). In general, the matrix B which determines a transformation of variables $w = B'u$, is set equal to A. However, in reduced support vector machines [12, 9] $B = \bar{A}$, where $\bar{A}$ is a submatrix of A whose rows are a small subset of the rows of A. In fact B can be a random matrix in $R^{\bar{m}\times n}$ with $n \leq \bar{m} \leq m$ if $m \geq n$ and $\bar{m} = m$ if $m \leq n$. This random choice of B holds the key to our privacy-preserving classifier and has been used effectively in SVM classification problems [15]. Our computational results of Section 4 will show that there is no essential difference between using a random B or a random submatrix of $\bar{A}$ of the rows of A as in reduced SVMs [12, 11]. One justification for these similar results can be given for the case when $\bar{m} \geq n$ and the rank of the $m \times n$ matrix B is n. For such a case, when B is replaced by A in (2.3), this results in a regular linear SVM formulation with a solution, say $v \in R^m$. In this case , the reduced SVM formulation (2.3) can match the regular SVM term $AA'v$ by the term $AB'u$, since $B'u = A'v$ has a solution u for any v because B' has rank n.

We shall now partition the random matrix B into p column blocks with each column block belonging to one of the p entities held privately by it and never made public. Thus, we have:

$$B = [B_{\cdot 1}\ \ B_{\cdot 2} \ldots\ldots B_{\cdot p}]. \tag{2.4}$$

We are ready to state our algorithm which will provide a linear classifier for the data without revealing privately held data blocks $[A_{\cdot 1}\ \ A_{\cdot 2} \ldots\ldots A_{\cdot p}]$. The accuracy of this algorithm will be comparable to that of a linear SVM using a publicly available A instead of merely $A_{\cdot 1}B_{\cdot 1}{}', A_{\cdot 2}B_{\cdot 2}{}' \ldots\ldots A_{\cdot p}B_{\cdot p}{}'$, as is the case here.

ALGORITHM 2.1. **Linear PPSVM Algorithm**

(I) *All p entities agree on the* same *labels for each data point, that is $D_{ii} = \pm 1$, $i = 1, \ldots\ldots, m$ and on the magnitude of $\bar{m}$, the number of rows of the random matrix B. (If an agreement on D is not possible, we can use semisupervised learning to handle such data points [1, 7]. We leave this to future work.)*

(II) *Each entity generates its own privately held random matrix $B_{\cdot j} \in R^{\bar{m} \times n_j}$, $j = 1, \ldots\ldots, p$, where n_j is the number of features held by entity j.*

(III) *Each entity j makes public its linear kernel $A_{\cdot j}B_{\cdot j}{}'$. This does not reveal $A_{\cdot j}$ but allows the public computation of the full linear kernel:*

$$AB' = A_{\cdot 1}B_{\cdot 1}{}' + A_{\cdot 2}B_{\cdot 2}{}' + \ldots\ldots + A_{\cdot p}B_{\cdot p}{}' \tag{2.5}$$

(IV) *A publicly calculated linear classifier $x'Bu - \gamma = 0$ is computed by some standard method such as* 1*-norm SVM [14, 2]:*

$$\begin{array}{rrcl} \min\limits_{(u,\gamma,y)} & \nu\|y\|_1 + \|u\|_1 & & \\ s.t. & D(AB'u - e\gamma) + y & \geq & e, \\ & y & \geq & 0. \end{array} \tag{2.6}$$

(V) *For each* new *$x \in R^n$, each entity makes public $x_j{}'B_{\cdot j}{}'$ from which a public linear classifier is computed as follows:*

$$x'B'u - \gamma = (x_1{}'B_{\cdot 1}{}' + x_2{}'B_{\cdot 2}{}' + \ldots\ldots + x_p{}'B_{\cdot p}{}')u - \gamma = 0, \tag{2.7}$$

which classifies the given x according to the sign of $x'Bu - \gamma$.

REMARK 2.2. *Note that in the above algorithm no entity j reveals its dataset $A_{\cdot j}$ nor its components of a new data point x_j. This is so because it is impossible to compute the mn_j numbers constituting $A_{\cdot j} \in R^{m \times n_j}$ given only the $m\bar{m}$ numbers constituting $(A_{\cdot j}B_{\cdot j}{}') \in R^{m \times \bar{m}}$ and not even knowing $B_{\cdot j} \in R^{\bar{m} \times n_j}$. Similarly it is impossible to compute $x_j \in R^{n_j}$ from $x_j{}'B_{\cdot j}{}' \in R$ without even knowing $B_{\cdot j}$. Hence, all entities share the publicly computed linear classifier (2.7) using AB' and $x'B'$ without revealing either the individual datasets or new point components.*

We turn now to nonlinear classification.

3 Nonlinear SVM Classifier for Vertically Partitioned Data

The approach to nonlinear classification is similar to that for the linear one, except that we make use of the Hadamard separability of a nonlinear kernel (1.1) which is satisfied by a Gaussian kernel. Otherwise, the approach is very similar to that of a linear kernel. We state that approach explicitly now.

ALGORITHM 3.1. **Nonlinear PPSVM Algorithm**

(I) All p entities agree on the same *labels for each data point, that $D_{ii} = \pm 1$, $i = 1,\ldots\ldots,p$ and on the magnitude of $\bar{m}$, the number of rows of the random matrix B. (If an agreement on D is not possible, we can use semisupervised learning to handle such data points [1, 7].)*

(II) Each entity generates its own privately held random matrix $B_{\cdot j} \in R^{\bar{m}\times n_j}$, $j = 1,\ldots\ldots,p$, where n_j is the number of input features held by entity j.

(III) Each entity j makes public its nonlinear kernel $K(A_{\cdot j}, B_{\cdot j}{}')$. This does not reveal $A_{\cdot j}$ but allows the public computation of the full nonlinear kernel:

$$K(A,B') = K(A_{\cdot 1},B_{\cdot 1}{}') \odot K(A_{\cdot 2},B_{\cdot 2}{}') \odot \ldots\ldots \odot K(A_{\cdot p},B_{\cdot p}{}') \tag{3.8}$$

(IV) A publicly calculated linear classifier $K(x',B)u - \gamma = 0$ is computed by some standard method such as 1*-norm SVM [14, 2]:*

$$\begin{array}{rrcl} \min\limits_{(u,\gamma,y)} & \nu\|y\|_1 + \|u\|_1 & & \\ s.t. & D(K(A,B')u - e\gamma) + y & \geq & e, \\ & y & \geq & 0. \end{array} \tag{3.9}$$

(V) For each new *$x \in R^n$, each entity makes public $K(x_j{}', B_{\cdot j}{}')$ from which a public nonlinear classifier is computed as follows:*

$$K(x',B')u - \gamma = (K(x_1{}',B_{\cdot 1}{}') \odot K(x_2{}',B_{\cdot 2}{}') \odot \ldots\ldots \odot K(x_p{}',B_{\cdot p}{}'))u - \gamma = 0, \tag{3.10}$$

which classifies the given x according to the sign of $K(x',B')u - \gamma$.

REMARK 3.2. *Note that in the above algorithm no entity j reveals its dataset $A_{\cdot j}$ nor its components of a new data point x_j. This is so because it is impossible to compute the mn_j numbers constituting $A_{\cdot j} \in R^{m\times n_j}$ given only the $m\bar{m}$ numbers constituting $K(A_{\cdot j}, B_{\cdot j}{}') \in R^{m\times\bar{m}}$ and not even knowing $B_{\cdot j} \in R^{\bar{m}\times n_j}$. Similarly it is impossible to compute $x_j \in R^{n_j}$ from $K(x_j{}', B_{\cdot j}{}') \in R$ without even knowing $B_{\cdot j}$. Hence, all entities share the publicly computed nonlinear classifier (3.10) using $K(A,B')$ and $K(x',B')$ without revealing either the individual datasets or new point components.*

Before turning to our computational results, it is important to note that Algorithms 2.1 and 3.1 can be used easily with other kernel classification algorithms instead of the 1-norm SVM, including the ordinary 2-norm SVM [19], the proximal SVM [6], and logistic regression [21].

It is instructive to compare our proposed privacy preserving SVM (PPSVM) given in Algorithms 2.1 and 3.1 to other recent work on privacy preserving classification for vertically-partitioned data which also makes use of random matrices. Du et al. [5] propose a method by which two parties can compute a privacy-preserving linear kernel classifier by securely computing the matrix $A'A$ with the use of random matrices. Yu et al. [24] use random matrices to securely compute the full kernel matrix, $K(A,A')$. Our approach is motivated by the observation that the accuracy of an SVM using a *random kernel*, $K(A,B')$, where B is a completely random matrix, is comparable to the accuracy of an SVM using the full kernel $K(A,A')$. We provide experimental support for this observation in Section 4. By using $K(A,B')$ instead of $K(A,A')$ we are able to obtain an accurate classifier with only very simple, asynchronous communication required among the entities. That is, each entity j need only broadcast $A_{\cdot j}B'_{\cdot j}$ to and receive the corresponding message from each of the other entities. In [5, 24] more communication steps are needed to securely compute $A'A$ or $K(A,A')$. For example, in [24], p rounds of data communication are needed among the p entities to compute $K(A,A')$.

We turn now to our computational results.

4 Computational Results

We illustrate the effectiveness of our proposed privacy preserving SVM (PPSVM) in two ways. First, we demonstrate that by using our approach entities can obtain classifiers with lower misclassification error than classifiers obtained using only the input features of each entity alone. Second, we show that a random kernel $K(A,B')$ achieves comparable accuracy to the usual kernel $K(A,A')$ or the reduced kernel $K(A,\bar{A}')$. All experiments were run using both a linear kernel and the commonly used Gaussian kernel described in Section 1. In all of our results, $\bar{A}$ consisted of ten percent of the rows of A randomly selected, while B was a completely random matrix of the same size as $\bar{A}$. Each entry of B was selected from a normal distribution with mean zero and standard deviation one. All datasets were normalized so that each feature had mean zero and standard deviation one. Note that this normalization is carried out for each feature independently, and does not require cooperation among the entities.

4.1 Comparison of our approach to classifiers obtained using only each entity's features We investigate the benefit of using our PPSVM approach instead of using only the input features available to each entity using seven datasets from the UCI repository [17]. To simulate a situation in which each entity has only a subset of the features for each data point, we randomly distribute the features among the entities such that each entity receives about the same number of features. We chose arbitrarily to perform experiments using five entities for each dataset, and also to perform experiments using whatever number of entities was needed so that each entity received about three features. We also investigate our approach as the number of entities increases on the Ionosphere dataset described below and in Figures 3 and 4.

Figure 1 shows results comparing the ten-fold cross validation misclassification error of our linear kernel PPSVM with the average misclassification error of the 1-norm SVM classifiers learned using only the input features available to each entity. Points below the 45 degree line represent experiments in which our PPSVM has lower error rate than the average error rate of the classifiers learned with only each entity's subset of the features. This indicates that the entities can expect improved performance using PPSVM instead of going it alone. Note that each dataset is represented by two points: one for the experiment using five entities, and one for the experiment using a sufficient number of entities so that each entity receives about three features. The results shown in Figure 1 are detailed in Table 1. We note that PPSVM obtains classifiers with lower error than the average of the classifiers using only each entity's features in eleven of the fourteen experiments. The parameter ν was selected from $\{10^i | i = -7,\dots,7\}$ for each dataset was selected using a random ten percent of each training set as a tuning set.

Figure 2 shows results for similar experiments using Gaussian kernels, with details in Table 2. We used the same datasets as for the experiments described above. To save time, we used the tuning strategy described in [10]. In this Nested Uniform Design approach, rather than evaluating a classifier at each point of a grid in the parameter space, the classifier is evaluated only at a set of points which is designed to "cover" the original grid to the extent possible. The point from this smaller set on which the classifier does best is then made the center of a grid which covers a smaller range of parameter space, and the process is repeated. Huang et al. [10] demonstrate empirically that this approach finds classifiers with similar misclassification error as a brute-force search through the entire grid. We set the initial range of $\log_{10}\nu$ to $[-7,7]$, and the initial range of $\log_{10}\mu$ as described in [10]. We used a Uniform Design with thirty runs from `http://www.math.hkbu.edu.hk/UniformDesign` for both

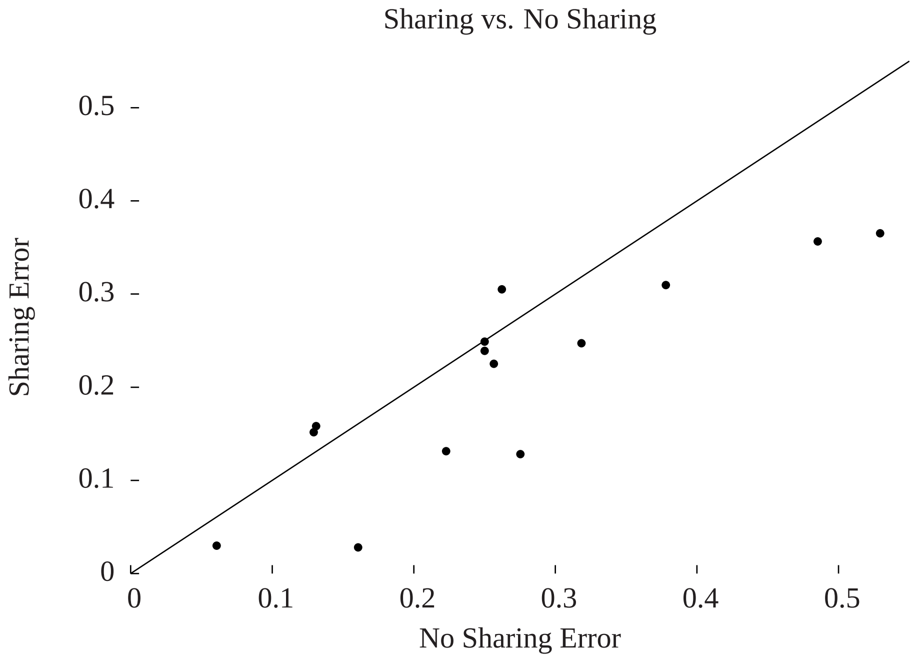

Figure 1: Error rate comparison of a 1-norm linear SVM sharing $A_{\cdot j}B'_{\cdot j}$ data for each entity versus a 1-norm linear SVM using just the input features $A_{\cdot j}$ of each entity. Points below the diagonal represent situations in which the error rate for sharing is lower than the error rate for not sharing. Results are given for each dataset with features randomly distributed evenly among 5 entities, and with features randomly distributed so that each entity has about 3 features. Seven datasets given in Table 1 were used to generate two points each.

Dataset Examples × Input Features	No. of Entities	No Sharing Error	Sharing Error
Cleveland Heart 297 × 13	5 4	0.1310 0.1293	0.1582 0.1516
Ionosphere 351 × 34	5 11	0.2229 0.2753	0.1312 0.1282
WDBC 569 × 30	5 10	0.0607 0.1607	0.0299 0.0281
Arrhythmia 452 × 279	5 93	0.2622 0.3780	0.3051 0.3096
German Credit 1000 × 24	5 8	0.2500 0.2500	0.2390 0.2490
Pima Indians 768 × 8	5 2	0.3184 0.2566	0.2472 0.2251
Bupa Liver 345 × 6	5 2	0.5294 0.4853	0.3652 0.3565

Table 1: Comparison of error rates for entities not sharing and sharing their datasets using a 1-norm linear SVM.

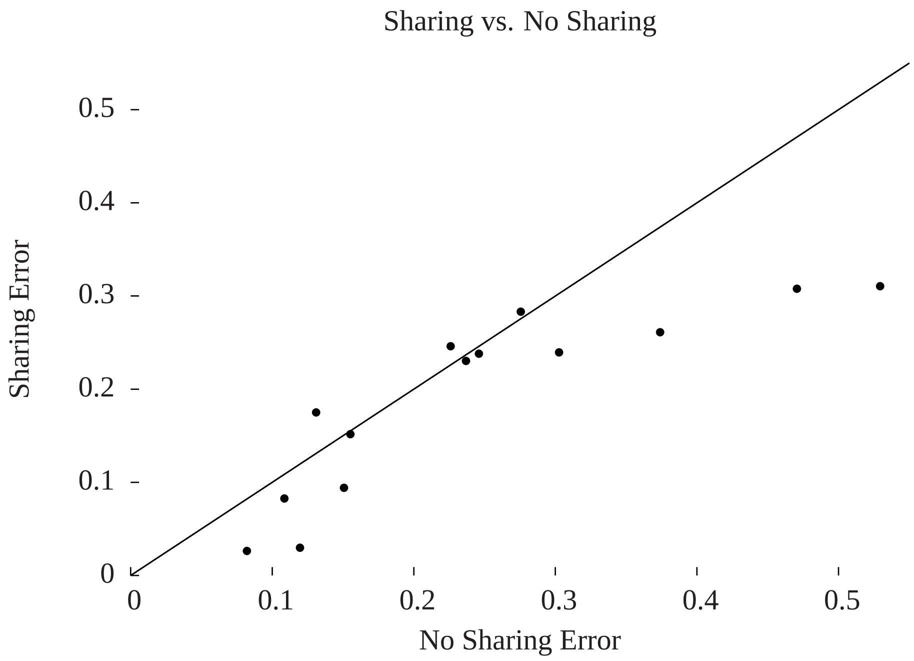

Figure 2: Error rate comparison of a 1-norm nonlinear SVM sharing $K(A_{\cdot j}, B'_{\cdot j})$ data for each entity versus a 1-norm nonlinear SVM using just the input features $A_{\cdot j}$ of each entity. Points below the diagonal represent situations in which the error rate for sharing is lower than the error rate for not sharing. Results are given for each dataset with features randomly distributed evenly among 5 entities, and with features randomly distributed so that each entity has about 3 features. Seven datasets given in Table 2 were used to generate two points each.

Dataset	No. of Entities	No Sharing Error	Sharing Error
Examples × Input Features			
Cleveland Heart	5	0.1310	0.1751
297 × 13	4	0.1552	0.1516
Ionosphere	5	0.1086	0.0826
351 × 34	11	0.1506	0.0941
WDBC	5	0.0821	0.0263
569 × 30	10	0.1196	0.0299
Arrhythmia	5	0.2756	0.2831
452 × 279	93	0.3740	0.2611
German Credit	5	0.2460	0.2380
1000 × 24	8	0.2400	0.2460
Pima Indians	5	0.3026	0.2394
768 × 8	2	0.2368	0.2303
Bupa Liver	5	0.5294	0.3105
345 × 6	2	0.4706	0.3077

Table 2: Comparison of error rates for entities not sharing and sharing their datasets using a 1-norm nonlinear Gaussian SVM.

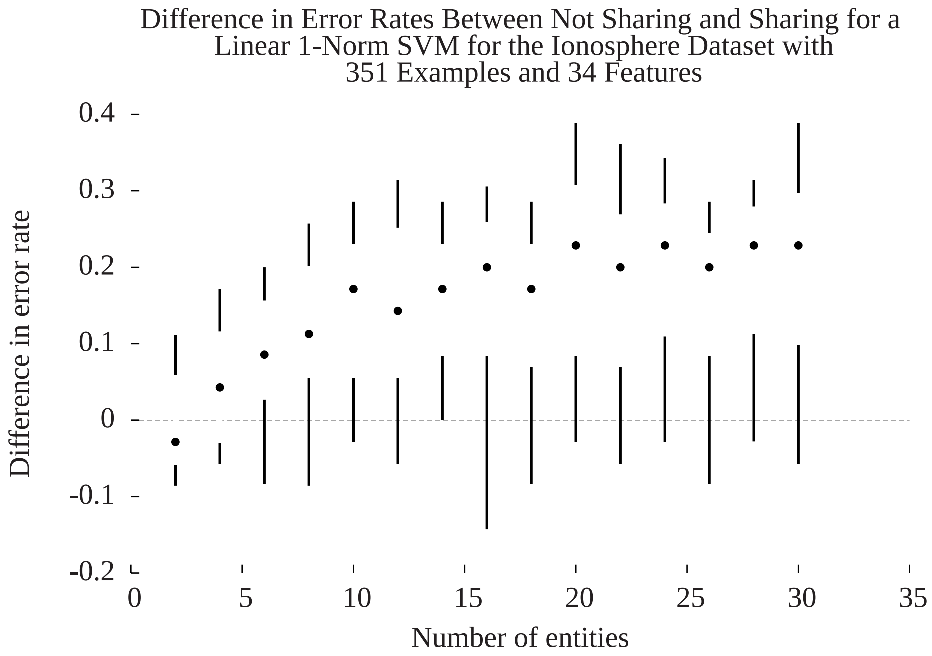

Figure 3: Box-and-whisker (median and interquartile) plot showing the improvement in error rate of linear kernel PPSVM as the number of entities increases from 2 to 30.

nestings, and used five-fold cross validation on the training set to evaluate each (ν,μ) pair.

We further explore the behavior of our approach as the number of entities changes on the Ionosphere dataset. Figure 3 shows the difference in misclassification error rates as the number of entities varies according to $\{2,4,\ldots,30\}$. For each number of entities, a box-and-whisker plot is given which shows the median (represented by a dot), interquartile range (the space between the dot and the vertical lines), and data range (the vertical lines) with outliers removed for each fold of ten-fold cross validation for all the entities. Note that as the number of entities increases, our PPSVM approach tends to have better median error rates, and that more of the observations favor PPSVM as more of the data lies above the $y = 0$ axis. The results shown in Figure 3 indicate that as each entity has fewer features, greater improvement due to using PPSVM would be expected, and also that some improvement is more likely to be observed. Figure 4 shows similar results using a Gaussian kernel. Each experiment was tuned according to the procedures for linear and nonlinear kernels described above.

4.2 Comparison of a random kernel to full and reduced kernels To justify the use of a random kernel we compare the performance of our PPSVM (Algorithms 2.1 and 3.1) with both an ordinary 1-norm SVM using a full kernel matrix and a 1-norm SVM using a reduced kernel matrix (RSVM) [12]. Figure 5 shows scatterplots comparing the error rates of our PPSVM with 1-norm SVM and PPSVM with RSVM, all using linear kernels. Note that points close to the 45 degree line represent datasets for which the classifiers being compared have similar error rates. All of the error rates were obtained using the procedure described above for linear kernels, and the datasets used are those in Table 1. Figure 6 shows similar results using the Gaussian kernel. All of the error rates were obtained using the same procedures and datasets as those in Table 2, described above. We note that the misclassification error for our PPSVM approach is comparable to that of 1-norm SVM and RSVM using both linear and Gaussian kernels.

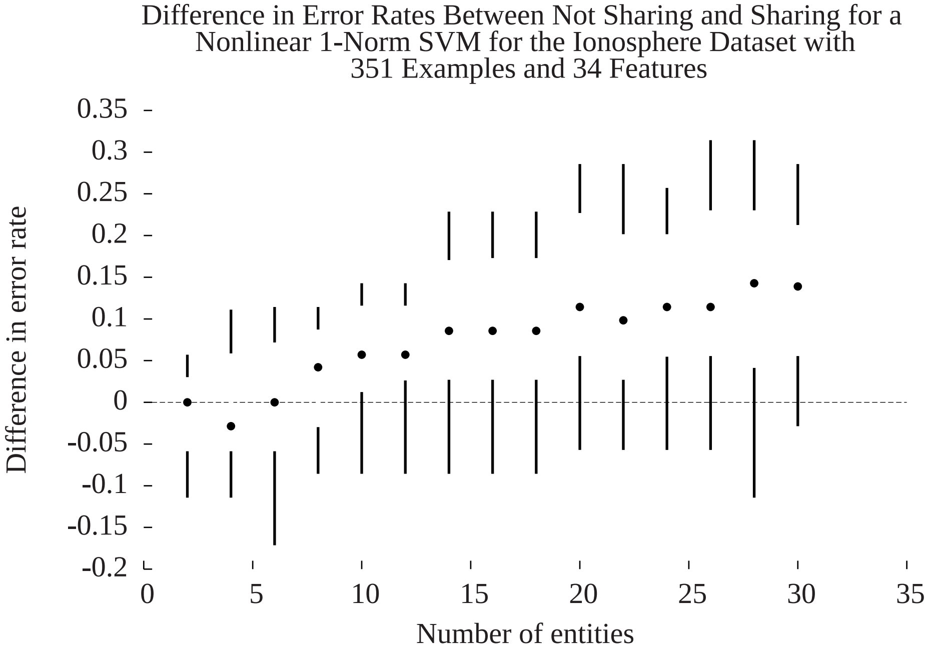

Figure 4: Box-and-whisker (median and interquartile) plot showing the improvement in error rate of Gaussian kernel PPSVM as the number of entities increases from 2 to 30.

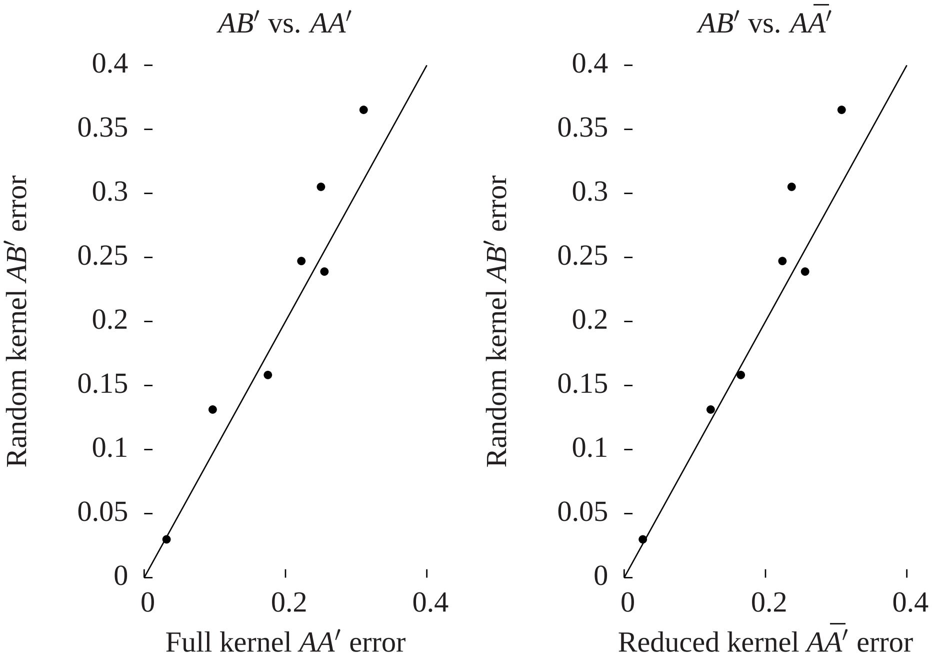

Figure 5: Error rate comparison of 1-norm linear SVMs for random kernel versus full and reduced kernels. For points below the diagonal, the random kernel has a lower error rate. The diagonal line in each plot marks equal error rates. One result is given for each dataset in Table 1.

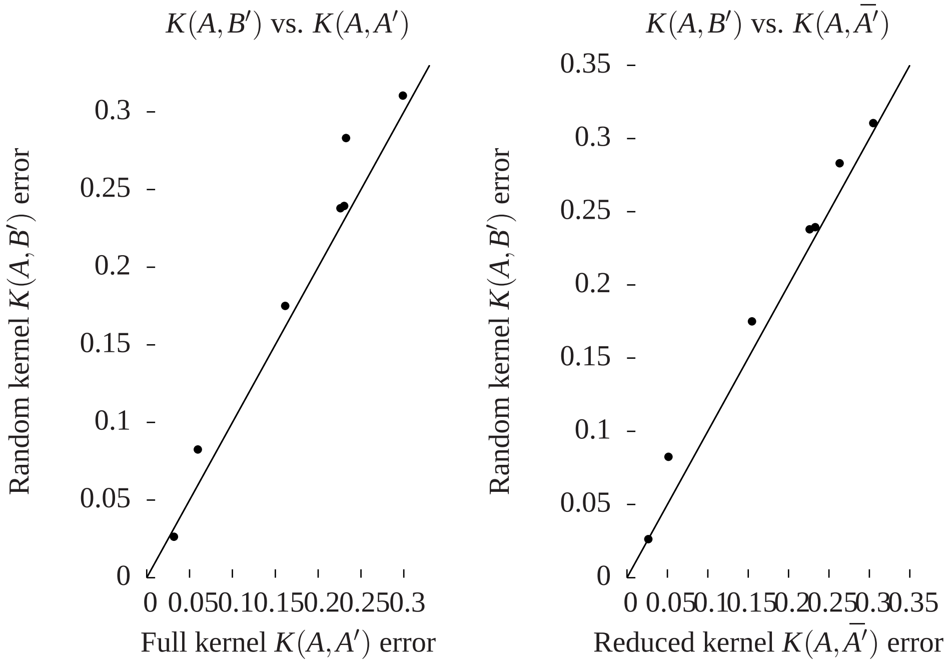

Figure 6: Error rate comparison of 1-norm nonlinear SVM for random kernel versus full and reduced kernels. For points below the diagonal, the random kernel has a lower error rate. The diagonal line in each plot marks equal error rates. One result is given for each dataset in Table 2.

5 Conclusion and Outlook

We have proposed a linear and nonlinear privacy-preserving SVM classifier based on a privately generated and privately held random matrix by each entity. Each entity possesses a different set of input features used collectively to generate the SVM classifier. The proposed approach uses all the privately held data in a form that does not reveal what that data is. Computational comparisons indicate that the accuracy of our proposed approach is comparable to full and reduced data classifiers. Furthermore, a marked improvement of accuracy is obtained by the privacy-preserving SVM compared to classifiers generated by each entity using its own data alone. Hence, by making use of a random kernel, the proposed approach succeeds in generating an accurate classifier based on privately held data without revealing any of that data.

Future work for horizontally partitioned data, wherein each entity possesses all of the same input features, but for different individuals, can be treated in a similar manner and will be described in a forthcoming paper [16].

Acknowledgements The research described in this Data Mining Institute Report 07-02, September 2007, was supported by National Science Foundation Grants CCR-0138308 and IIS-0511905, the Microsoft Corporation and ExxonMobil.

References

[1] K. P. Bennett and A. Demiriz. Semi-supervised support vector machines. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems -10-*, pages 368–374, Cambridge, MA, 1998. MIT Press.

[2] P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In J. Shavlik, editor, *Proceedings 15th International Conference on Machine Learning*, pages 82–90, San Francisco, California, 1998. Morgan Kaufmann. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-03.ps.

[3] K. Chen and L. Liu. Privacy preserving data classification with rotation perturbation. In *Proceedings of the Fifth International Conference of Data Mining (ICDM'05)*, pages 589–592. IEEE, 2005.

[4] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, 2000.

[5] W. Du, Y. Han, and S. Chen. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, pages 222–233, 2004. http://citeseer.ist.psu.edu/du04privacypreserving.html.

[6] G. Fung and O. L. Mangasarian. Proximal support vector machine classifiers. In F. Provost and R. Srikant, editors, *Proceedings KDD-2001: Knowledge Discovery and Data Mining, August 26-29, 2001, San Francisco, CA*, pages 77–86, New York, 2001. Association for Computing Machinery. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-02.ps.

[7] G. Fung and O. L. Mangasarian. Semi-supervised support vector machines for unlabeled data classification. *Optimization Methods and Software*, 15:29–44, 2001. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/99-05.ps.

[8] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, England, 1985.

[9] S.Y. Huang and Y.-J. Lee. Theoretical study on reduced support vector machines. Technical report, National Taiwan University of Science and Technology, Taipei, Taiwan, 2004. yuh-jye@mail.ntust.edu.tw.

[10] C.-H. Hunag, Y.-J. Lee, D.K.J. Lin, and S.-Y. Huang. Model selection for support vector machines via uniform design. In *Machine Learning and Robust Data Mining of Computational Statistics and Data Analysis*, Amsterdam, 2007. Elsevier Publishing Company. http://dmlab1.csie.ntust.edu.tw/downloads/papers/UD4SVM013006.pdf.

[11] Y.-J. Lee and S.Y. Huang. Reduced support vector machines: A statistical theory. *IEEE Transactions on Neural Networks*, 18:1–13, 2007.

[12] Y.-J. Lee and O. L. Mangasarian. RSVM: Reduced support vector machines. In *Proceedings First SIAM International Conference on Data Mining, Chicago, April 5-7, 2001, CD-ROM*, 2001. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/00-07.pdf.

[13] L. Liu, J. Wang, Z. Lin, and J. Zhang. Avelet-based data distortion for privacy-preserving collaborative analysis. Technical Report 482-07, Department of Computer Science, University of Kentucky, Lexington, KY 40506, 2007. http://www.cs.uky.edu/ jzhang/pub/MINING/lianliu1.pdf.

[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 M. E. Thompson. Massive data classification via unconstrained support vector machines. *Journal of Optimization Theory and Applications*, 131:315–325, 2006. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/06-01.pdf.

[16] O. L. Mangasarian and E. W. Wild. Privacy-preserving classification of horizontally partitioned data via random kernels. Technical Report 07-03, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, October 2007. In preparation.

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

[18] H. Lipmaa S. Laur and T. Mielikäinen. Cryptographically private support vector machines. In D. Gunopulos L. Ungar, M. Craven and T. Eliassi-Rad, editors, *Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2006, Philadelphia, August 20–23, 2006. ACM*, pages 618–624, 2006. http://eprints.pascal-network.org/archive/00002133/01/cpsvm.pdf.

[19] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[20] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, second edition, 2000.

[21] G. Wahba. Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 69–88, Cambridge, MA, 1999. MIT Press. ftp://ftp.stat.wisc.edu/pub/wahba/index.html.

[22] M.-J. Xiao, L.-S. Huang, H. Shen, and Y.-L. Luo. Privacy preserving id3 algorithm over horizontally partitioned data. In *Sixth International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT'05)*, pages 239–243. IEEE Computer Society, 2005.

[23] H. Yu, X. Jiang, and J. Vaidya. Privacy-preserving SVM using nonlinear kernels on horizontally partitioned data. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 603–610, New York, NY, USA,

2006. ACM Press.

[24] H. Yu, J. Vaidya, and X. Jiang. Privacy-preserving svm classification on vertically partitioned data. In *Proceedings of PAKDD '06*, volume 3918 of *Lecture Notes in Computer Science*, pages 647 – 656. Springer-Verlag, January 2006.