Privacy-Preserving Random Kernel Classification of Checkerboard Partitioned Data

Olvi L. Mangasarian[*] Edward W. Wild[†]

Abstract

We propose a privacy-preserving support vector machine (SVM) classifier for a data matrix A whose input feature columns as well as individual data point rows are divided into groups belonging to different entities. Each entity is unwilling to make public its group of columns and rows. Our classifier utilizes the entire data matrix A while maintaining the privacy of each block. This classifier is based on the concept of a random kernel $K(A,B')$ where B' is the transpose of a random matrix B, as well as the reduction of a possibly complex pattern of data held by each entity into a checkerboard pattern. The proposed 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 based on the entire set of input features and data points all made public.

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

1 INTRODUCTION

Recently there has been wide interest in privacy-preserving support vector machine (SVM) classification. 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 [22], privacy-preserving SVM classifiers were obtained for vertically partitioned data by adding random perturbations to the data. In [20, 21], *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 [16], wavelet-based distortion [10] and rotation perturbation [2]. More recently [14, 13] a random kernel $K(A,B')$ where B' is the transpose of a random matrix B was used to handle vertically partitioned data [14] as well as horizontally partitioned data [13].

In this work we propose a highly efficient privacy-preserving SVM (PPSVM) classifier for vertically *and* horizontally partitioned data that employs a random kernel $K(A,B')$. Thus the $m \times n$ data matrix A with n features and m data points, each of which in R^n, is partitioned in a possibly complex way among p entities as depicted, for example, among $p = 4$ entities as shown in Figure 1. Our task is to construct an SVM classifier based on the entire data matrix A without requiring the contents of each entity's matrix block be made public.

Our approach will be to first subdivide a given data matrix A that is owned by p entities into a checkerboard pattern of q cells, with $q \geq p$, as depicted, for example in Figure 2. Secondly, each cell block A_{ij} of the checkerboard will be utilized to generate the random kernel block $K(A_{ij}, B_{\cdot j}{}')$, where $B_{\cdot j}$ is random matrix of appropriate dimension. It will be shown in Section 2 that under mild assumptions, the random kernel $K(A_{ij}, B_{\cdot j}{}')$

[*] 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*.

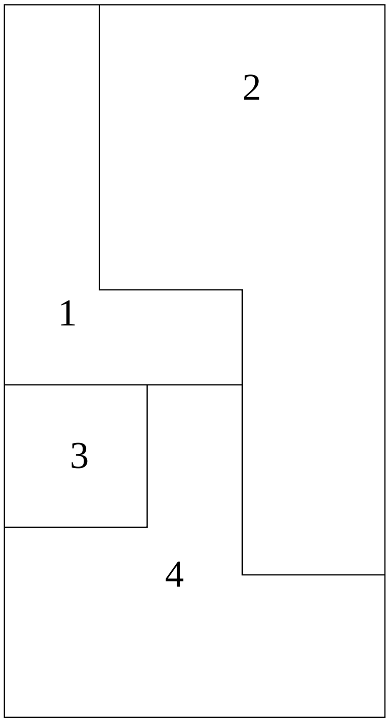

Figure 1: A data matrix A partitioned into $p = 4$ blocks with each block owned by a distinct entity.

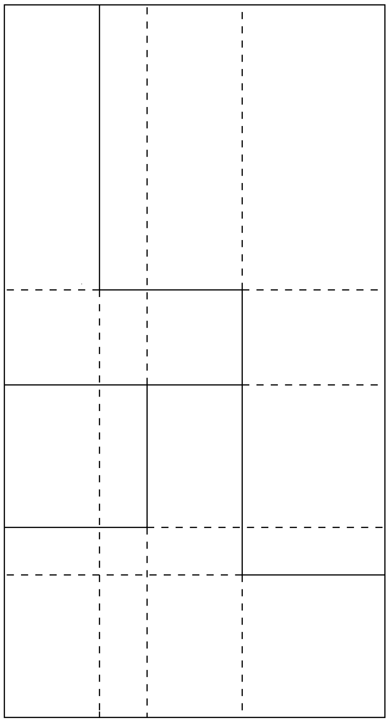

Figure 2: The checkerboard pattern containing $q = 20$ cell blocks generated from the data matrix A of Figure 1.

will safely protect the data block A_{ij} from discovery by entities that do not own it, while allowing the computation of a classifier based on the entire data matrix A.

We now briefly describe the contents of the paper. In Section 2 we present our method for a privacy-protecting linear SVM classifier for checkerboard 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 that use the entire dataset. 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: $(\sum_{i=1}^{n}|x_i|)$. 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 [18, 17, 11] 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 [18, 17, 3], 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 [5], $E \in R^{m\times n_1}$, $F \in R^{m\times n_2}$, $G \in R^{k\times n_1}$ 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 Checkerboard 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 columns of A are partitioned into s vertical blocks of $n_1, n_2, \ldots\ldots$ and n_s columns in each block such that $n_1 + n_2 + \ldots + n_s = n$. Furthermore, all of the column blocks are identically partitioned into r horizontal blocks of $m_1, m_2, \ldots\ldots$ and m_r rows in each block such that $m_1 + m_2 + \ldots + m_r = m$. This checkerboard pattern of data similar to that of Figure 2 may result from a more complex data pattern such that of Figure 1. We note that each cell block of the checkerboard is owned by a separate entity but with the possibility of a single entity owning more than one checkerboard cell. No entity is willing to make its cell block(s) public. Furthermore, each individual 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 all the 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 [11]. 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 [9, 7] $B = \bar{A}$, where $\bar{A}$ is a submatrix of A whose rows are a small subset of the rows of A. However, 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 [12]. Our computational results of Section 4 will show that there is no substantial difference between using a random B or a random submatrix of $\bar{A}$ of the rows of A as in reduced SVMs [9, 8]. One justification for these similar results can be given for the case when $\bar{m} \geq n$ and the rank of the $\bar{m} \times n$ 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 n columns of the random matrix $B \in R^{\bar{m}\times n}$ into s column blocks with column block $B_{\cdot j}$ containing n_j columns for $j = 1,\ldots,s$. Furthermore, each column block $B_{\cdot j}$ will be generated by entities owning the $m-$by-n_j column block of $A_{\cdot j}$ and is never made public. Thus, we have:

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

We will show that under the assumption that:

$$n_j > \bar{m}, \ \ j = 1,\ldots,s, \tag{2.5}$$

the privacy of each checkerboard block privacy is protected.

We are ready to state our algorithm which will provide a linear classifier for the data without revealing privately held checkerboard cell blocks A_{ij}, $i = 1,\ldots,r$, $j = 1,\ldots,s$. The accuracy of this algorithm will, in general, be comparable to that of a linear SVM using a publicly available A instead of merely $A_{\cdot 1}B_{\cdot 1}{}', A_{\cdot 2}B_{\cdot 2}{}' \ldots\ldots A_{\cdot s}B_{\cdot s}{}'$, as will be the case in the following algorithm.

ALGORITHM 2.1. **Linear PPSVM Algorithm**

(I) All 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 $B \in R^{\bar{m}\times n}$ which must satisfy (2.5).*

(II) All entities $i = 1,\ldots,r$, sharing the same same column block j, $1 \leq j \leq s$, with n_j features, must agree on using the same $\bar{m} \times n_j$ random matrix $B_{\cdot j}$ which is privately held by themselves.

(III) Each entity $i = 1,\ldots,r$, owning cell block A_{ij} makes public its linear kernel $A_{ij}B_{\cdot j}{}'$, but not A_{ij}. This allows the public computation of the full linear kernel:

$$(AB')_i = A_{i1}B_{\cdot 1}{}' + \ldots\ldots + A_{is}B_{\cdot s}{}', \ i = 1,\ldots,r. \tag{2.6}$$

(IV) A publicly calculated linear classifier $x'Bu - \gamma = 0$ is computed by some standard method such as 1*-norm SVM [11, 1] for some positive parameter ν:*

$$\begin{array}{rrcl} \min\limits_{(u,\gamma,y)} & \nu\|y\|_1 + \|u\|_1 & & \\ s.t. & D(AB'u - e\gamma) + y & \geq & e, \\ & y & \geq & 0. \end{array} \tag{2.7}$$

(V) *For each* new $x \in R^n$*, the component blocks* ${x_j}'{B_{\cdot j}}'$*,* $j = 1,\ldots,s$*, are made public from which a public linear classifier is computed as follows:*

$$x'B'u - \gamma = ({x_1}'{B_{\cdot 1}}' + {x_2}'{B_{\cdot 2}}' + \ldots\ldots + {x_s}'{B_{\cdot s}}')u - \gamma = 0, \tag{2.8}$$

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

REMARK 2.2. *Note that in the above algorithm no entity* ij *which owns cell block* A_{ij} *reveals its dataset nor its components of a new data point* x_j*. This is so because it is impossible to compute the* $m_i n_j$ *numbers constituting* $A_{ij} \in R^{m_i \times n_j}$ *given only the* $m_i\bar{m}$ *numbers constituting* $(A_{ij}{B_{\cdot j}}') \in R^{m_i \times \bar{m}}$*, because* $m_i n_j > m_i\bar{m}$*. Similarly it is impossible to compute the* n_j *numbers constituting* $x_j \in R^{n_j}$ *from the* $\bar{m}$ *constituting* ${x_j}'{B_{\cdot j}}' \in R^{\bar{m}}$ *because* $n_j > \bar{m}$*. Hence, all entities share the publicly computed linear classifier (2.8) using* AB' *and* $x'B'$ *without revealing either the individual datasets or new point components.*

We turn now to nonlinear classification.

3 Privacy Preserving Nonlinear Classifier for Checkerboard 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* s *entities agree on the* same *labels for each data point, that* $D_{ii} = \pm 1$*,* $i = 1,\ldots\ldots,m$ *and on the magnitude of* $\bar{m}$*, the number of rows of the random matrix* $B \in R^{\bar{m} \times n}$ *which must satisfy (2.5).*

(II) *All entities* $i = 1,\ldots,r$*, sharing the same same column block* j*,* $1 \leq j \leq s$*, with* n_j *features, must agree on using the same* $\bar{m} \times n_j$ *random matrix* $B_{\cdot j}$ *which is privately held by themselves.*

(III) *Each entity* $i = 1,\ldots,r$*, owning cell block* A_{ij} *makes public its nonlinear kernel* $K(A_{ij}, {B_{\cdot j}}')$*, but not* A_{ij}*. This allows the public computation of the full nonlinear kernel:*

$$K(A,B')_i = K(A_{i1},{B_{\cdot 1}}') \odot \ldots\ldots \odot K(A_{is},{B_{\cdot s}}'),\ i = 1,\ldots,r. \tag{3.9}$$

(IV) *A publicly calculated linear classifier* $K(x',B')u - \gamma = 0$ *is computed by some standard method such as* 1*-norm SVM [11, 1] for some positive parameter* ν*:*

$$\begin{array}{rlcl} \displaystyle\min_{(u,\gamma,y)} & \nu\|y\|_1 + \|u\|_1 & & \\ s.t. & D(K(A,B')u - e\gamma) + y & \geq & e, \\ & y & \geq & 0. \end{array} \tag{3.10}$$

(V) *For each* new $x \in R^n$*, the component blocks* $K(x^{j'}, {B_{\cdot j}}')$*,* $j = 1,\ldots,s$*, are made public from which a public nonlinear classifier is computed as follows:*

$$K(x',B')u - \gamma = (K({x_1}',{B_{\cdot 1}}') \odot K({x_2}',{B_{\cdot 2}}') \odot \ldots\ldots \odot K({x_s}',{B_{\cdot s}}'))u - \gamma = 0, \tag{3.11}$$

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 ij which owns cell block A_{ij} reveals its dataset nor its components of a new data point x_j. This is so because it is impossible to compute the m_in_j numbers constituting $A_{ij} \in R^{m_i\times n_j}$ given only the $m_i\bar{m}$ numbers constituting $K(A_{ij},B_{\cdot j}{}') \in R^{m_i\times\bar{m}}$, because $m_in_j > m_i\bar{m}$. Similarly it is impossible to compute the n_j numbers constituting $x_j \in R^{n_j}$ from the $\bar{m}$ constituting $K(x_j{}',B_{\cdot j}{}') \in R^{\bar{m}}$ because $n_j > \bar{m}$. Hence, all entities share the publicly computed nonlinear classifier (3.11) 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 useful 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 [17], the proximal SVM [4], and logistic regression [19].

We turn now to our computational results.

4 Computational Results

To illustrate the effectiveness of our proposed privacy preserving SVM (PPSVM), we used seven datasets from the UCI Repository [15] to simulate a situation in which data is distributed among several different entities. We formed a checkerboard partition which divided the data into blocks, with each entity owning exactly one block. Each block had data for approximately 25 examples, and we carried out experiments in which there were one, two, four, and eight vertical partitions (for example, the checkerboard pattern in Figure 2 has four vertical partitions). Thus, the blocks in each experiment all contained all, one half, one fourth, or one eighth of the total number of features. With one vertical partition, our approach is the same as the technique for horizontally partitioned data described in [13], and these results provide a baseline for the experiments with more partitions. We note that the errors with no sharing represent a worst-case scenario in that a different entity owns each block of data. If entities owned multiple blocks, their errors without sharing might decrease. Nevertheless, it is unlikely that such entities would generally do better than our PPSVM approach, especially in cases in which the PPSVM is close to the ordinary 1-norm SVM.

We compare our PPSVM approach to a situation in which each entity forms a classifier only using its own data, with no sharing, and to a situation in which all entities share the reduced kernel $K(A,\bar{A}')$ without privacy, where $\bar{A}$ is a matrix whose rows are a random subset of the rows of A [9]. Results for one, two, four, and eight vertical partitions are reported in Table 1. All experiments were run using the commonly used Gaussian kernel described in Section 1. In every result, $\bar{A}$ consisted of ten percent of the rows of A randomly selected, while B was a completely random matrix with the same number of columns as A. The number of rows of B was set to the minimum of $n-1$ and the number of rows of $\bar{A}$, where n is the number of features in the vertical partition. Thus, we ensure that the condition (2.5) discussed in the previous sections holds in order to guarantee that the private data A_{ij} cannot be recovered from $K(A_{ij},B')$. Each entry of B was selected independently from a uniform distribution on the interval $[0,1]$. All datasets were normalized so that each feature was between zero and one. This normalization can be carried out if the entities disclose only the maximum and minimum of each feature in their datasets. When computing ten-fold cross validation, we first divided the data into folds and set up the training and testing sets in the usual way. Then each entity's dataset was formed from the training set of each fold. The accuracies of all classifiers were computed on the testing set of each fold.

To save time, we used the tuning strategy described in [6] to choose the parameters ν of (3.10) and μ of the Gaussian kernel. 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. [6] 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 [6]. Note that we set the initial range of $\log_{10}\mu$ independently for

Dataset Examples × Features	No. of Vertical Partitions	Rows of B	Ideal Error Using Entire Data without Privacy $K(A,\bar{A}')$	PPSVM Error Sharing Protected Data $K(A,B')$	Error Using Individual Data without Sharing $K(A_{is},A_{is}{}')$
Cleveland Heart (CH)	1	12	0.17	0.15	0.24
297 × 13	2	5	0.19	0.19	0.28
	4	2	0.17	0.24	0.30
Ionosphere (IO)	1	33	0.07	0.09	0.19
351 × 34	2	16	0.06	0.11	0.20
	4	7	0.05	0.17	0.21
	8	3	0.06	0.26	0.24
WDBC (WD)	1	29	0.03	0.03	0.11
569 × 30	2	14	0.02	0.04	0.10
	4	6	0.03	0.06	0.12
	8	2	0.03	0.11	0.16
Arrhythmia (AR)	1	45	0.21	0.27	0.38
452 × 279	2	45	0.22	0.28	0.36
	4	45	0.23	0.27	0.40
	8	33	0.24	0.29	0.40
Pima Indians (PI)	1	7	0.23	0.25	0.36
768 × 8	2	3	0.23	0.31	0.35
	4	1	0.23	0.34	0.38
Bupa Liver (BL)	1	5	0.30	0.40	0.42
345 × 6	2	2	0.30	0.42	0.42
German Credit (GC)	1	23	0.24	0.24	0.34
1000 × 24	2	11	0.24	0.29	0.34
	4	5	0.24	0.30	0.34
	8	2	0.24	0.30	0.33

Table 1: Comparison of error rates for entities sharing entire data without privacy through the reduced kernel $K(A,\bar{A}')$, sharing data using our PPSVM approach, and not sharing data. When there are enough features, results are given for situations with one, two, four, and eight vertical partitions using a Gaussian kernel.

each entity using only that entity's examples and features. We used a Uniform Design with thirty runs from `http://www.math.hkbu.edu.hk/UniformDesign` for both nestings, and used leave-one-out cross validation on the training set to evaluate each (ν,μ) pair when the entities did not share and five-fold cross validation on the training set they did. We used leave-one-out cross validation when not sharing because only about 25 examples were available to each entity in that situation.

To illustrate the improvement in error rate of PPSVM compared to an ordinary 1-norm SVM based only on the data for each entity with no sharing, we provide a graphical presentation of some results in Table 1. Figure 3 shows a scatterplot comparing the error rates of our data-sharing PPSVM versus the 1-norm no-sharing reduced SVM using Gaussian kernels. The diagonal line in both figures marks equal error rates. Note that points below the diagonal line represent datasets for which PPSVM has a lower error rate than the average error of the entities using only their own data. Figure 3 shows a situation in which there are two vertical partitions of the dataset, while Figure 4 shows a situation in which there are four vertical partitions. Note that in Figure 3, our PPSVM approach has a lower error rate for six of the seven datasets, while in Figure 4, PPSVM has a lower error rate on

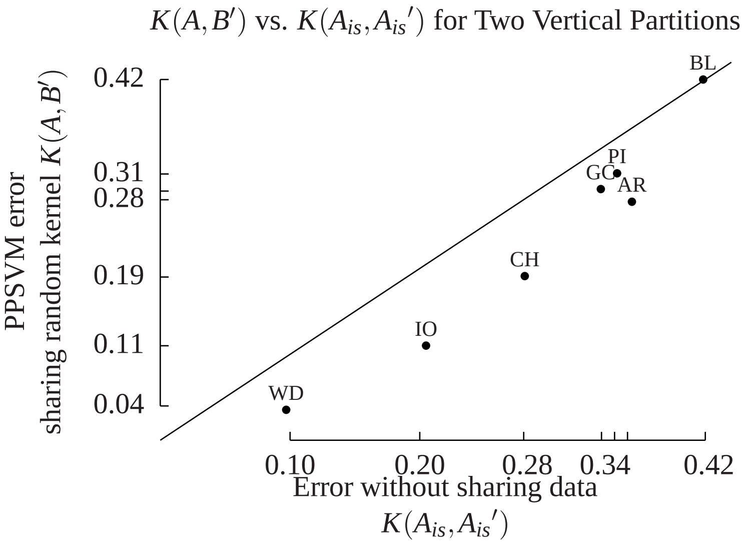

Figure 3: Error rate comparison of our PPSVM with a random kernel $K(A,B')$ vs 1-norm nonlinear SVMs sharing no data for checkerboard data with two vertical partitions. For points below the diagonal, PPSVM has a better error rate. The diagonal line in each plot marks equal error rates. Each point represents the result for the dataset in Table 1 corresponding to the letters attached to the point.

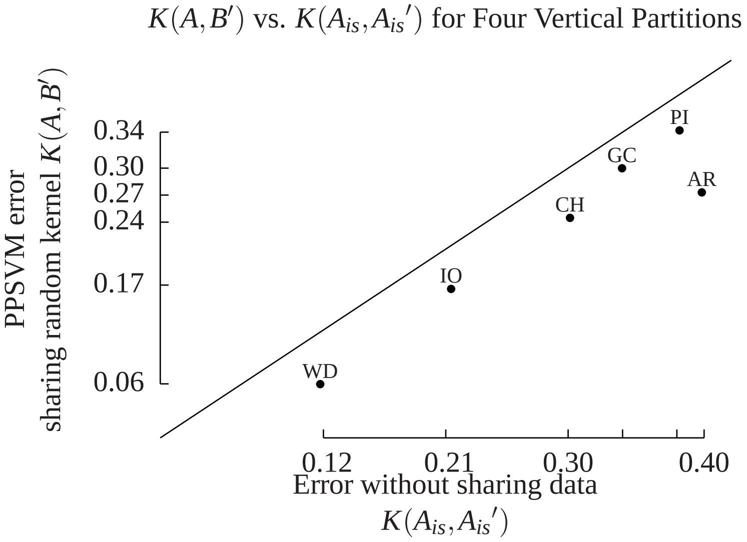

Figure 4: Error rate comparison of our PPSVM with a random kernel $K(A,B')$ vs 1-norm nonlinear SVMs sharing no data for checkerboard data with four vertical partitions. For points below the diagonal, PPSVM has a better error rate. The diagonal line in each plot marks equal error rates. Each point represents the result for the dataset in Table 1 corresponding to the letters attached to the point. Note that there are not enough features in the Bupa Liver dataset for four vertical partitions.

all six datasets.

5 Conclusion and Outlook

We have proposed a linear and nonlinear privacy-preserving SVM classifier for a data matrix, arbitrary blocks of which are held by various entities that are unwilling to make their blocks public. Our approach divides the data matrix into a checkerboard pattern and then creates a linear or nonlinear kernel matrix from each cell block of the checkerboard together with a suitable random matrix that preserves the privacy of the cell block data. 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 for each cell block, the proposed approach succeeds in generating an accurate classifier based on privately held data without revealing any of that data.

Future work will entail combining our approach with other ones such as those of rotation perturbation [2], cryptographic approach [16] and data distortion [10].

Acknowledgments The research described in this Data Mining Institute Report 08-02, September 2008, was supported by National Science Foundation Grant IIS-0511905.

References

[1] 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.

[2] 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.

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

[4] 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.

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

[6] C.-H. Huang, 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.

[7] 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.

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

[9] 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.

[10] L. Liu, J. Wang, Z. Lin, and J. Zhang. Wavelet-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.

[11] 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.

[12] 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.

[13] 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, November 2007. Proceedings of the 4th International Conference on Data Mining, DMIN'08, Las Vegas July 2008, to appear.

[14] O. L. Mangasarian, E. W. Wild, and G. M. Fung. Privacy-preserving classification of vertically partitioned data via random kernels. Technical Report 07-02, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, September 2007.

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

[16] 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.

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

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.