Privacy-Preserving Linear and Nonlinear Approximation via Linear Programming

Glenn M. Fung* & Olvi L. Mangasarian†

Abstract

We propose a novel privacy-preserving random kernel approximation based on a data matrix $A \in R^{m\times n}$ whose rows are divided into privately owned blocks. Each block of rows belongs to a different entity that is unwilling to share its rows or make them public. We wish to obtain an accurate function approximation for a given $y \in R^m$ corresponding to each of the m rows of A. Our approximation of y is a real function on R^n evaluated at each row of A and is based on the concept of a reduced kernel $K(A, B')$ where B' is the transpose of a completely random matrix B. The proposed linear-programming-based approximation, which is public but does not reveal the privately-held data matrix A, has accuracy comparable to that of an ordinary kernel approximation based on a publicly disclosed data matrix A.

Keywords: privacy-preserving approximation, random kernels, support vector machines, linear programming

1 INTRODUCTION

The problem addressed in this work is that of obtaining an approximation to a given vector $y \in R^m$ of function values corresponding to the m rows of a data matrix $A \in R^{m\times n}$ that represents m points in the n-dimensional real space R^n. The matrix A is partitioned into q blocks of rows belonging to q entities that are unwilling to share their data or make them public. The motivation for this work arises from similar problems arising in classification theory where the data, corresponding to rows of a data matrix, is also held by various private entities and hence referred to as *horizontally partitioned* data. Thus in [19, 15] privacy-preserving support vector machine (SVM) classifiers were obtained for such data, while in [20] induction tree classifiers were generated for similar problems. Other privacy-preserving classifying techniques include cryptographically private SVMs [7], wavelet-based distortion [10] and rotation perturbation [3]. There is also a substantial body of research on privacy preservation in linear programming such as [1, 12, 13]. However, there does not appear to be any privacy-preserving applications to approximation problems in the literature. This is the problem we wish to address here as follows.

In this work we propose an efficient privacy-preserving approximation (PPA) for horizontally partitioned data that 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 $K(A, A') : R^{m\times n} \times R^{n\times m} \longrightarrow R^{m\times m}$ for constructing a linear or nonlinear approximation of a given $y \in R^m$ corresponding to the m rows of A, we use a *random* kernel [9, 8] $K(A, B') : R^{m\times n} \times R^{n\times \bar{m}} \longrightarrow R^{m\times \bar{m}}$, $\bar{m} < n$, where B is a completely random matrix that is publicly disclosed. Such a random kernel will be shown to completely hide the data matrix A. The second idea is that each entity $i \in \{1, \ldots, q\}$ makes public only the kernel function $K(A_i, B')$ of its

*R&D Clinical Systems, Siemens Medical Solutions, Inc., 51 Valley Stream Parkway. Malvern, PA 19355*glenn.fung@siemens.com.*

†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.*

privately held matrix block of data $A_i \in R^{m_i \times n}$. By employing these two ideas, we shall describe a linear-programming-based algorithm that protects the privacy of each horizontal partition block of the data matrix A, owned by a distinct entity, while generating a privacy-preserving approximation which has an accuracy comparable to that of an ordinary kernel approximation based on a totally disclosed A.

We now briefly describe the contents of the paper. In Section 2 we describe our method for a privacy-preserving linear kernel approximation for horizontally partitioned data. We show there that by using a random linear kernel each entity does not disclose its privately held data while enabling the generation of a global linear approximation to a given $y \in R^m$. Section 3 extends these results to nonlinear kernels. In Section 4 we give linear-programming-based computational results which show that the accuracy of our privacy-preserving nonlinear random krenel approximation is comparable not only to ordinary kernel approximation based on openly available data, but also accurately reproduces the underlying nonlinear function which generated the privately held data. Section 5 concludes the paper.

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: $(\sum_{i=1}^{n}|x_i|)$ while $\|x\|$ denotes the 2-norm: $(\sum_{i=1}^{n}(x_i)^2)^{\frac{1}{2}}$. 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, $A_{\cdot j}$ the j-th column or the j-th block of columns of A, and A_{ir} the ir-th element of A or the r-th row of i-th block A_i 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, 4, 11] whose ij-th element, $i = 1, \ldots, m,\ j = 1, \ldots, k$, 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, 16, 5]. However, we shall assume that they are associative in the following sense:

$$K\left(\begin{bmatrix} E \\ F \end{bmatrix}, G'\right) = \begin{pmatrix} K(E, G') \\ K(F, G') \end{pmatrix}, \tag{1.1}$$

where $E \in R^{m_1 \times n}$, $F \in R^{m_2 \times n}$, $G \in R^{k \times n}$. It is straightforward to check that both a linear kernel $K(A, B') = AB'$ and a Gaussian kernel satisfy (1.1). For the k matrices $G^1, \ldots, G^k$, of the same dimensions, their *affine hull* is defined as the set $\{H \mid H = \sum_{j=1}^{j=k}\lambda^j G^j$, with $\sum_{j=1}^{j=k}\lambda^j = 1\}$. The abbreviation "s.t." stands for "subject to".

2 Privacy-Preserving Linear Kernel Approximation

We wish to construct linear approximation of a given vector y in R^m associated with the m n-dimensional points represented by the m rows of the matrix $A \in R^{m \times n}$. The matrix A is divided into q blocks of $m_1, m_2, \ldots m_q$ rows with $m_1 + m_2 + \ldots + m_q = m$, and each block of rows "owned" by an entity that is unwilling to make it public or share it with others. The linear kernel approximation to be generated based on this data will be a plane defined on R^n as:

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

where $w = B'u$, $w \in R^n$ is the normal to the approximating plane $x'w - \gamma$, $\gamma \in R$ determines the distance of the plane from the origin and B is a completely 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 given data A and y satisfy, to the extent possible, the approximation:

$$AB'u - e\gamma \approx y. \tag{2.3}$$

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, 6] $B = \bar{A}$, 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 $\bar{m} < n$ for our application here. The random choice of B holds the key to our privacy-preserving approximation and has been used effectively in SVM classification problems [14]. Computational results have shown that there is no essential difference between using a random B or a random submatrix of $\bar{A}$ of the rows of A in reduced SVMs [9, 8].

We shall partition our data matrix A into q row blocks $A_1, A_2, \ldots, A_q$ with each row block belonging to one of the q entities and held privately by it and never made public. However, what is made public by each entity i is the matrix product A_iB' which allows the public calculation of the the linear kernel $AB' \in R^{m\times\bar{m}}$ as follows:

$$\begin{bmatrix} A_1B' \\ A_2B' \\ \vdots \\ A_qB' \end{bmatrix} \tag{2.4}$$

We are now ready to state our algorithm which will provide a linear classifier for the data without revealing the private blocks of the privately held data blocks $A_1, A_2, \ldots, A_q$. The accuracy of this algorithm will be comparable to that of a linear SVM using a publicly available linear kernel AA' instead of merely the blocks $A_1B', A_2B', \ldots, A_qB'$ of (2.4) as is the case here.

Algorithm 2.1. **Linear PPA Algorithm**

(I) *All q entities agree on the same random matrix $B \in R^{\bar{m}\times n}$ with $\bar{m} < n$ for security reasons as justified in the explanation immediately following this algorithm.*

(II) *All entities make public the function values y_ℓ, $\ell = 1, \ldots, m$, for all the rows A_i, $i = 1, \ldots, m$, of the data matrix A that they all hold collectively.*

(III) *Each entity i makes public its linear kernel $A_iB' \in R^{m_i\times\bar{m}}$. This does not reveal its matrix block A_i but allows the public computation of the full linear kernel:*

$$AB' = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_q \end{bmatrix} B' \tag{2.5}$$

(IV) A publicly calculated linear approximation $x'Bu - \gamma$ is computed by some standard method such as a 1-norm error minimization together with a Tikhonov regularization term $\nu\|u\|_1$ [17, 2] as follows:

$$\begin{array}{rrl} \displaystyle\min_{(u,\gamma,s)} & \|s\|_1 + \nu\|u\|_1 & \\ s.t. & s \geq AB'u - e\gamma - y & \geq -s. \end{array} \tag{2.6}$$

This problem is equivalent to the **linear program***:*

$$\begin{array}{rrl} \displaystyle\min_{(u,\gamma,s,t)} & e's + \nu e't & \\ s.t. & s \geq AB'u - e\gamma - y & \geq -s, \\ & t \geq u & \geq -t. \end{array} \tag{2.7}$$

(V) For each new *privately held $x \in R^n$ obtained by any entity, that entity privately computes $x'B'$ from which a linear approximation is computed as follows:*

$$x'B'u - \gamma. \tag{2.8}$$

Note that *no* entity i reveals its data matrix block A_i. This is so because each entity reveals only the $m_i\bar{m}$ numbers constituting the matrix $P_i = (A_iB') \in R^{m_i\times\bar{m}}$. When $\bar{m} < n$, there is an infinite number of matrices $A_i \in R^{m_i\times n}$ satisfying $A_iB' = P_i$, given B and P_i. We make this statement more precise by first showing that at least an exponential number of matrices A_i satisfy $A_iB' = P_i$ for a given B and P_i when $\bar{m} < n$. We then show that the infinite number of matrices that lie in the affine hull of these matrices also satisfy $A_iB' = P_i$. This obviously precludes the possibility of determining the matrix block A_i held by entity i given only A_iB'.

Proposition 2.2. *Given the matrix product $P_i' = A_iB' \in R^{m_i\times\bar{m}}$ where $A_i \in R^{m_i\times n}$ is unknown and B is a known matrix in $R^{\bar{m}\times n}$ with $\bar{m} < n$, there are an infinite number of solutions, including:*

$$\binom{n}{\bar{m}}^{m_i} = \left(\frac{n!}{(n-\bar{m})!\bar{m}!}\right)^{m_i}, \tag{2.9}$$

possible solutions $A_i \in R^{m_i\times n}$ to the equation $BA_i' = P_i$. Furthermore, the infinite number of matrices in the affine hull of these $\binom{n}{\bar{m}}^{m_i}$ matrices also satisfy $BA_i' = P_i$.

Proof See [15, Proposition 2.2].

For the specific case of $\bar{m} = n-1$, which is typically used in numerical computations, we have that:

$$\binom{n}{\bar{m}}^{m_i} = \left(\frac{n!}{(n-1)!}\right)^{m_i} = (n)^{m_i}. \tag{2.10}$$

This translates to $(2)^{10}$ for a typical case of $n = 2$, $\bar{m} = 1$ and $m_i = 10$.

We turn now to nonlinear approximation.

3 Privacy-Preserving Nonlinear Kernel Approximation

The approach to nonlinear approximation is similar to that for the linear one, except that we make use of the associative property (1.1) of a nonlinear kernel which is satisfied by a Gaussian kernel. This kernel is one of the most commonly used nonlinear kernels. Otherwise, the approach is very similar to that of a linear kernel. Thus, we wish to obtain the nonlinear kernel approximation:

$$y \approx K(x', B')u - \gamma, \tag{3.11}$$

where K is a Gaussian kernel, B is a random matrix in $R^{\bar{m}\times n}$ with $\bar{m} < n$, and $u \in R^{\bar{m}}$, $\gamma \in R$ are parameters to be determined as described in the following algorithm.

Algorithm 3.1. **Nonlinear PPSVM Algorithm**

(I) All q entities agree on the same random matrix $B \in R^{\bar{m}\times n}$ with $\bar{m} < n$ for security reasons as justified in the explanation immediately following this algorithm.

(II) All entities make public the function values y_ℓ, $\ell = 1,\ldots,m$ for the data matrix rows A_i, $i = 1,\ldots,m$, that they all hold.

(III) Each entity i makes public its nonlinear kernel $K(A_i, B')$. This does not reveal its matrix block A_i but allows the public computation of the full nonlinear kernel:

$$K(A, B') = K\left(\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_q \end{bmatrix}, B'\right) = \begin{bmatrix} K(A_1, B') \\ K(A_2, B') \\ \vdots \\ K(A_q, B') \end{bmatrix} \tag{3.12}$$

(IV) A publicly calculated nonlinear approximation $K(x', B)u - \gamma$ is computed by some standard method such as a 1-norm using Tikhonov regularization [17, 2]:

$$\begin{array}{rcl} \min\limits_{(u,\gamma,s)} & \|s\|_1 + \nu\|u\|_1 & \\ s.t. & s \geq K(A, B')u - e\gamma - y & \geq \ -s. \end{array} \tag{3.13}$$

This problem is equivalent to the following **linear program***:*

$$\begin{array}{rcl} \min\limits_{(u,\gamma,a,ts)} & e's + \nu e't & \\ s.t. & s \geq K(A, B')u - e\gamma - y & \geq \ -s, \\ & t \geq t & \geq \ -s. \end{array} \tag{3.14}$$

(V) For each new *$x \in R^n$ obtained by an entity, that entity privately computes $K(x', B')$ from which a nonlinear approximation is computed as follows:*

$$K(x', B')u - \gamma. \tag{3.15}$$

Note that in the above algorithm no entity i reveals its data matrix block A_i. This is so because it is impossible to compute unique $m_i n$ numbers constituting the matrix $A_i \in R^{m_i\times n}$ given only the $m_i\bar{m}$ numbers constituting the revealed kernel matrix $K(A_i, B') \in R^{m_i\times\bar{m}}$ with $\bar{m} < n$. However, all entities share the publicly computed nonlinear approximation (3.15) without revealing their individual matrix blocks A_i, $i = 1,\ldots,q$, or any new x that they obtain. Thus, for example, if we wish to compute the r-th row A_{ir} of entity i's data matrix A_i from the given matrix $P_i = K(A_i, B') \in R^{m_i\times\bar{m}}$, we need to solve the $\bar{m}$ *nonlinear* equations $K(B, {A_{ir}}') = {P_{ir}}'$ for the n components of $A_{ir} \in R^n$. Because $n > \bar{m}$, this would in general generate a nonlinear surface in R^n containing an infinite number of solutions which makes it impossible to determine A_{ir} uniquely.

We turn now to our linear-programming-based computational results.

4 Computational Results

We illustrate the effectiveness of our proposed privacy-preserving approximation (PPA) in three ways. First, we demonstrate that by using our approach, entities can obtain better approximations than approximations obtained using only the data of each entity alone. Second, we show that a random kernel $K(A, B')$ achieves comparable accuracy to that of the usual kernel $K(A, A')$ utilizing all the privately held data made public. Third, we show that PPA gives an accurate approximation to the underlying function itself that generated the privately held data.

The number of rows $\bar{m}$ of B was set to 1000. Thus, we ensure that the conditions discussed in the previous sections hold in order to guarantee the private data A_i cannot be recovered from $K(A_i, B')$. 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 variable in their datasets. In all the algorithms the Gaussian kernel parameter μ was chosen by cross-validation from the set $\{0.01, 0.2, 0.4, 0.6, 0.8, 1, 2\}$ and the regularization parameter ν was chosen by cross-validation from the set $\{2^{-5}, 2^{-4}, \ldots, 2^4, 2^5\}$.

4.1 Comparison of our privacy-preserving approach with approximations obtained using only each entity's data and also with all data revealed We investigate the benefit of using our PPA approach, instead of using only the data available to each entity alone, by using the two-dimensional sinc function defined on the square region $D = [-3, 3] \times [-3, 3]$:

$$sinc(x_1, x_2) = \frac{sin\pi x_1}{\pi x_1} \cdot \frac{sin\pi x_2}{\pi x_2}, \tag{4.16}$$

Figure 2 shows a graph of the sinc function $sinc(x_1, x_2)$ as defined above in (4.16).

For this experiment, we suppose that we have data owned by two private entities, Entity 1 and Entity 2, and that the data cannot be shared. Entity 1 has 1000 data points on a grid E_1 of equally spaced point that lies in the subset: $S_1 = \{[-1.5, 1.5] \times [-1.5, 1.5]\} \subset D$, while Entity 2 has 1000 data points on a grid E_2 of equally spaced points in the subset: $S_2 = D - S_1$ as shown in Figure 1.

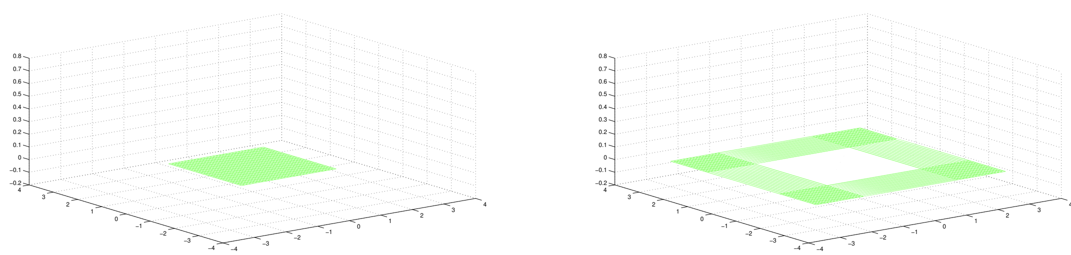

Figure 1: **(Left): Entity 1 domain** $S_1 = \{[-1.5, 1.5] \times [-1.5, 1.5]\}$ **, (Right): Entity 2 domain** $S_2 = D - S_1$

Our computational results are depicted in Figures 3 through 6 and can be summarized as as follows:

- Figure 3 depicts a poor approximation of the sinc function defined above based only on the privately held data of Entity 1. In order to measure the quality of the approximation with respect to the original sinc function, we calculate the relative error defined as:

$$Error_{E_1} = \Big(\sum_{x \in E_1 \cup E_2} \frac{(app_{E_1}(x) - sinc(x))^2}{sinc(x)^2}\Big)^{\frac{1}{2}}, \tag{4.17}$$

  where app_{E_1} is the approximation learned only using data from the set E_1. The value of $Error_{E_1}$ was 5.01

- Similarly Figure 4 shows a poor approximation of the sinc function based only on the privately held data of Entity 2. For this experiment the value of $Error_{E_2}$ was 1.05.
- Figure 5 shows an approximation of the sinc function based on publicly **revealed** data of both entities. The relative error in this case was 0.01.
- Figure 6 depicts an approximation of the sinc function based on **unrevealed** privately held data as described in our proposed Algorithm 3.1. The relative error in this case was also 0.01.

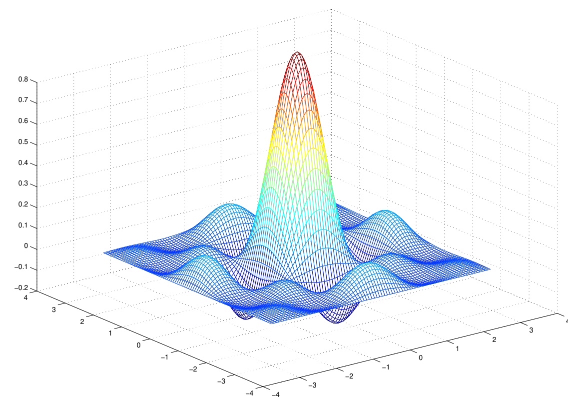

Figure 2: **The exact sinc function** $sinc(x_1, x_2) = \frac{sin\pi x_1}{\pi x_1} \cdot \frac{sin\pi x_2}{\pi x_2}$

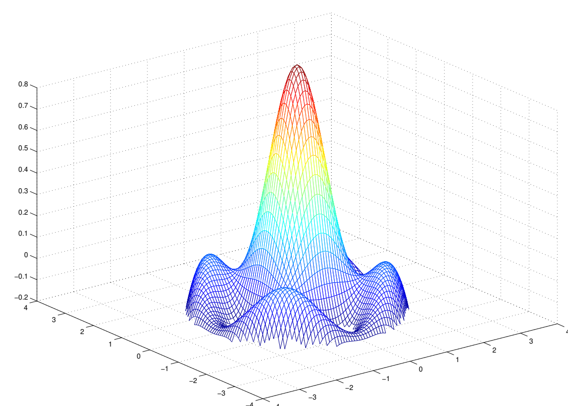

Figure 3: **The approximation of** $sinc(x_1, x_2)$ **by the data of Entity 1 only**

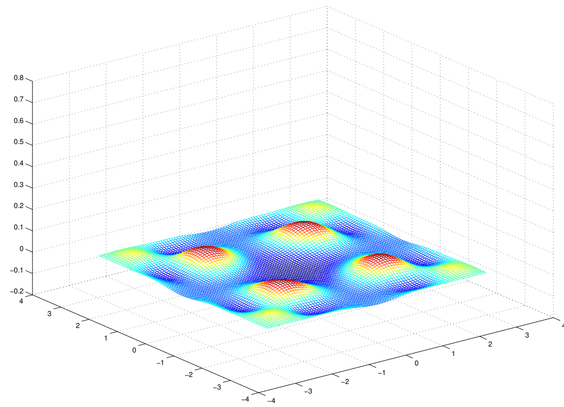

Figure 4: **The approximation of $sinc(x_1, x_2)$ by the data of Entity 2 only**

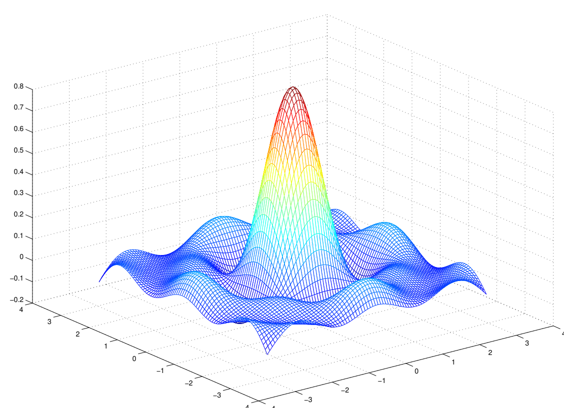

Figure 5: **The approximation of $sinc(x_1, x_2)$ by the publicly revealed data of both Entities 1 and 2**

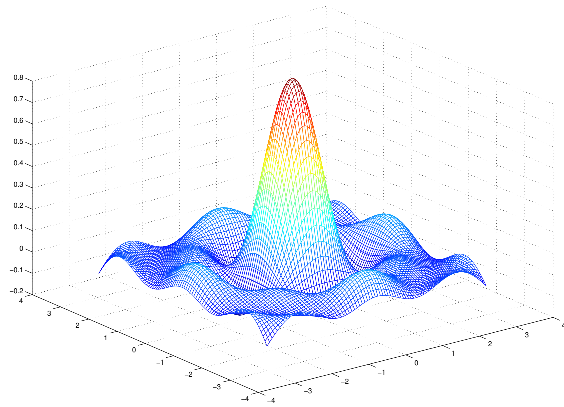

Figure 6: **The approximation of $sinc(x_1, x_2)$ by the privately held data of Entities 1 and 2 utilizing Algorithm 3.1**

5 Conclusion

We have proposed a linear and nonlinear privacy-preserving support vector machine (SVM) approximation based on a privately generated and privately held random matrix by each entity. Each entity possesses a different set of data used collectively to generate the SVM approximation. The proposed approach uses all the privately held data, yet does not reveal that data. Computational comparisons indicate that the accuracy of our proposed approach is comparable to fully revealed data approximation. Furthermore, the accuracy obtained by the privacy-preserving approximation is markedly better than the accuracy of approximation generated by each entity using only its own data.

Acknowledgments The research described in this Data Mining Institute Report 11-04, October 2011, was supported by the Microsoft Corporation and ExxonMobil.

References

[1] A. Bednarz, N. Bean, and M. Roughan. Hiccups on the road to privacy-preserving linear programming. In *Proceedings of the 8th ACM Workshopon on Privacy in the Electronic Society*, pages 117–120, 2009.

[2] C. M. Bishop. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7:108–116, 1995.

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

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

[6] 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.

[7] Sven Laur, Helger Lipmaa, and Taneli Mielikäinen. Cryptographically private support vector machines. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 618–624, New York, NY, USA, 2006. ACM.

[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. Privacy-preserving horizontally partitioned linear programs. Technical Report 10-02, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, April 2010. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/10-02.pdf, Optimization Letters, to appear.

[13] O. L. Mangasarian. Privacy-preserving linear programming. Technical Report 10-01, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, March 2010. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/10-01.pdf, Optimization Letters 5(01), 2011, 165-172.

[14] 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.

[15] 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 2008 International Conference on Data Mining, DMIN08, Las Vegas July 2008, Volume II, 473-479, R. Stahlbock, S.V. Crone and S. Lessman, Editors.

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

[17] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill–Posed Problems*. John Wiley & Sons, New York, 1977.

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

[19] 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.

[20] 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.