Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: 0005-1144 (Print) 1848-3380 (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

Collaborative classification mechanism for privacy-Preserving on horizontally partitioned data

Zhancheng Zhang, Fu-Lai Chung & Shitong Wang

To cite this article: Zhancheng Zhang, Fu-Lai Chung & Shitong Wang (2019) Collaborative classification mechanism for privacy-Preserving on horizontally partitioned data, Automatika, 60:1, 58-67, DOI: 10.1080/00051144.2019.1578039

To link to this article: https://doi.org/10.1080/00051144.2019.1578039

AUTOMATIKA
2019, VOL. 60, NO. 1, 58–67
https://doi.org/10.1080/00051144.2019.1578039

REGULAR PAPER

OPEN ACCESS Check for updates

Collaborative classification mechanism for privacy-Preserving on horizontally partitioned data

Zhancheng Zhang[a], Fu-Lai Chung[b] and Shitong Wang[c]

[a]School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou, Jiangsu, People's Republic of China; [b]Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong, People's Republic of China; [c]School of Digital Media, Jiangnan University, Wuxi, Jiangsu, People's Republic of China

ABSTRACT

We propose a novel two-party privacy-preserving classification solution called Collaborative Classification Mechanism for Privacy-preserving (C^2MP^2) over horizontally partitioned data that is inspired from the fact, that global and local learning can be independently executed in two parties. This model collaboratively trains the decision boundary from two hyper-planes individually constructed by its own privacy data and global data. C^2MP^2 can hide true data entries and ensure the two-parties' privacy. We describe its definition and provide an algorithm to predict future data point based on Goethals's Private Scalar Product Protocol. Moreover, we show that C^2MP^2 can be transformed into existing Minimax Probability Machine (MPM), Support Vector Machine (SVM) and Maxi–Min Margin Machine (M^4) model when privacy data satisfy certain conditions. We also extend C^2MP^2 to a nonlinear classifier by exploiting kernel trick. Furthermore, we perform a series of evaluations on real-world benchmark data sets. Comparison with SVM from the point of protecting privacy demonstrates the advantages of our new model.

ARTICLE HISTORY
Received 23 September 2011
Accepted 22 December 2017

KEYWORDS
Classification; privacy-preserving; collaborative learning; support vector machine

1. Introduction

Collecting training and predicting data are two necessary steps in the pattern classification system. Those data instances are generally distributed in different parties. Traditional classifiers deal with the data under the assumption that all parties' data can be free accessed and centralized at the data centre. Currently, privacy concerns may prevent the parties from directly sharing the data and some confidential information about the data. It is well documented [1–4] that the unlimited exposing of privacy through the Internet and other media has reached a point where threats against privacy are very common and deserve serious concern.

Generally, there are mainly two kinks of approaches for privacy-preserving classification: the perturbation-based approach [5] and the cryptography-based approach [6]. The methods based on perturbation have been widely used for data mining, however, when being used for classification, those methods must have a trade-off between privacy and accuracy. The methods based on cryptography can safely preserve privacy without loss of accuracy, however, have high computing and communication costs.

Each privacy-preserving classifier may face the two scenarios: the vertically distributed data [7] and the horizontally distributed data [6,8]. In the first scenario, the features of one entry may be distributed in multi-parties. In the second scenario, each entity holds all the feature values for its own group of parties while other entities hold similar data for other groups of parties.

Global and local learning is a recently emerging field, to the best of our knowledge, the idea was firstly introduced by Lanckriet et al. in the Minimax Probability Machine (MPM) [9], this model utilizes a given mean and covariance matrix of each class to render individual global data, and tries to minimize the probability of misclassification of future data points in a worst-case setting, as a result, an optimal linear discriminant is obtained with an explicit upper bound on the probability of misclassification of future data. Following this idea, Huang et al. presented a unified theory of the Maxi–Min Margin Machine (M^4) [10,11] that collaboratively learns from local and global data and its connections with SVM, MPM and LDA are established. Furthermore, Deng et al. proposed a new classifier called Minimax-probability Based Fuzzy Hyperellipsoid Machine (MP-FHM) [12] where global data are represented by the centre and radius of hyperellipsoid. Also, in recent research [13], a bridge between the Minimum Enclosing Ball (MEB) [14] and the Fuzzy Inference Systems (FIS) was established. Those studies demonstrated that collaborating on classification with local data and global data gives advantages to a classifier.

From the viewpoint of privacy-preserving, local data can be considered as privacy data, and global

CONTACT Zhancheng Zhang cimszhang@163.com School of Electronic and Information Engineering, Suzhou University of Science and Technology, Kerui Road 1, CN-215009 Suzhou, Jiangsu, People's Republic of China

data can be estimated from local data. In turn, when some conditions are satisfied, local data cannot be derived from global data and privacy information will not be revealed from global data. Hence, global data can be appropriately used to hide privacy information in a classification scheme. A fact in the real world is that one's privacy should be shielded only from others, and should be freely accessed by oneself. Those findings motivate us to develop the proposed model.

In this paper, we focus on a two-category classification task in which Alice holds $X^{(A)} = \{x_i^{(A)}\}_{i=1}^{N_x^{(A)}}$ and $Y^{(A)} = \{y_j^{(A)}\}_{j=1}^{N_y^{(A)}}$, Bob holds $X^{(B)} = \{x_i^{(B)}\}_{i=1}^{N_x^{(B)}}$ and $Y^{(B)} = \{y_j^{(B)}\}_{j=1}^{N_y^{(B)}}$, where $x_i^{(A)}, x_i^{(B)} \in \mathbb{R}^n$ with positive label and $y_j^{(A)}, y_j^{(B)} \in \mathbb{R}^n$ with negative label. The two parties want to collaboratively learning a classifier from those samples. By the traditional method, those data may be centralized to train a classifier, however, for privacy concerns, $X^{(A)}$ and $Y^{(A)}$ owned by Alice are prohibited from being accessed by Bob, likewise $X^{(B)}$ and $Y^{(B)}$ are shielded from Alice.

Our proposed Collaborative Classification Mechanism for Privacy-Preserving (C^2MP^2) is different from existing privacy-preserving classifiers for its collaborative mechanism. Our approach bases on the following idea. Alice and Bob can individually get their local classifiers by training their local data, and the two local classifiers can be combined to get a jointed decision. From the view point of global and local learning, inaccuracy of local classifier can be compensated by introducing global information to the local classifiers.

As shown later, C^2MP^2 is closely related to the three models, namely M^4, MPM and SVM. Another important feature of the C^2MP^2 is that no any third party needs and its training and testing algorithm can be executed only within two parties.

The third feature of our proposed model is that the linear version of C^2MP^2 can be extended to more powerful nonlinear classification approach by using kernel trick.

The paper is organized as follows. In the next section, we overview the related preliminaries and definitions. In Section 3, we introduce the unkernelized linear version of C^2MP^2 model in detail, including its definition, collaborative mechanism, solving method, secure training and testing algorithms. In this section, we will also analyse its various connections with the existing M^4, MPM and SVM models. Following that, we demonstrate the kernelized nonlinear version of C^2MP^2. We then, in Section 5, evaluate the unkernelized linear version and kernelized nonlinear version of C^2MP^2 on real-world benchmark data sets. Finally, we summarize the main results of the paper, give concluding remarks and envision possible future work in Section 6.

2. Related preliminaries

Privacy Information. Most privacy concerns can be classified as either unwanted intrusions into an individual's private life, or the right to control the uses of personal information about oneself [15]. In this paper, we assume that all raw data except label attribute can be regarded as privacy, and should be shielded against other parties, thus, local data equal to private data.

Global Data are those data, which summarize the data and provide the practitioners with knowledge on the structure of data [10]. In this paper, the mean value and covariance matrix denoted as $\{\bar{x}, \Sigma_x\}$ and $\{\bar{y}, \Sigma_y\}$ in each category data represent global data respectively.

Secure Two-party Computation [16,17] deals with computing any function on any input in a distributed network. Each participant holds one of the inputs while ensuring that no more information is revealed to a participant in the computation that can be inferred from the participant's input and output.

Homomorphic Encryption Scheme (HES) [18,19] is a public-key cryptosystem represented by a triple (Gen, Enc, Dec), in which, Gen is the key generator, encryption algorithm Enc and its corresponding decryption algorithm satisfy that, given any two cipher-text $Enc(A)$ and $Enc(B)$, there exists a cipher-text $Enc(A * B)$ such that $Enc(A) * Enc(B) = Enc(A * B)$, hereafter $Dec(Enc(A * B)) = (A * B)$, where $*$ is an algebraic operation in Group G. This property can be used to construct a secure inner product when $*$ is addition operation.

3. Privacy-preserving classification scheme

In the following, we first present the definition and formulation of unkernelized linear version of C^2MP^2, then introduce its collaborative idea and discuss its connections with other models including M^4, MPM and SVM. In this section, we also present its training and testing algorithms and analyse their security.

3.1. *Linear version of* C^2MP^2

Following previously described data distribution, let $N^{(A)} = N_x^{(A)} + N_y^{(A)}$ and $N^{(B)} = N_x^{(B)} + N_y^{(B)}$ respectively denote the number of data entries held by Alice and Bob, let $N = N^{(A)} + N^{(B)}$ denote the total number of data entries. Hereafter, we denote the covariance matrix of positive class and that of negative class as Σ_x and Σ_y.

We wish to determine a hyperplane $f(z) = w^T z + b$, where $w \in \mathbb{R}^n \backslash \{0\}$ and $b \in \mathbb{R}$, which separates the above horizontally distributed two classes of data as robustly as possible. Future data points z for which $f(z) \geq 0$ will be then classified as positive class; otherwise, they will be classified as negative class. The procedure of training and testing should guarantee

that the private data entries in $X^{(A)}$ and $Y^{(A)}$ cannot be disclosed to Bob, and Bob's data entries should be also shielded against Alice. Moreover, when testing the future data z held by Tom, the privacy of z should not be disclosed to any other party.

We construct the first classifier with the privacy data of Alice and the global data, such that the privacy data of Alice is shielded. This classifier can be reasonably expressed as

$$\max_{w_a \neq 0, b_a} \quad \rho_a \tag{1}$$

$$\text{s.t.} \quad \frac{({w_a}^T x_i^{(A)} + b_a)}{\sqrt{{w_a}^T \Sigma_x w_a}} \geq \rho_a, \quad i = 1, 2, \ldots, N_x^{(A)}, \tag{2}$$

$$\frac{-({w_a}^T y_j^{(A)} + b_a)}{\sqrt{{w_a}^T \Sigma_y w_a}} \geq \rho_a, \quad j = 1, 2, \ldots, N_y^{(A)}. \tag{3}$$

The second classifier is constructed likewise for using only the privacy data of Bob, and can be also reasonably expressed as

$$\max_{w_b \neq 0, b_b} \quad \rho_b \tag{4}$$

$$\text{s.t.} \quad \frac{({w_b}^T x_i^{(B)} + b_b)}{\sqrt{{w_b}^T \Sigma_x w_b}} \geq \rho_b, \quad i = 1, 2, \ldots, N_x^{(B)}, \tag{5}$$

$$\frac{-({w_b}^T y_j^{(B)} + b_b)}{\sqrt{{w_b}^T \Sigma_y w_b}} \geq \rho_b, \quad j = 1, 2, \ldots, N_y^{(B)}, \tag{6}$$

where $\Sigma_x, \Sigma_y \in \mathbb{R}^{n \times n}$ respectively denote the covariance of positive class and negative class, both are symmetric and positive semi-definite.

The first classifier described by (1)–(3) tries to maximize the margin defined as the minimum Mahalanobis distance between the privacy training samples of Alice. This classifier uses only the local data held by Alice and the covariance matrices of the two classes, and can be executed only be Alice, without disclosure of Alice's privacy. The second classifier described by (4)–(6) works likewise and can protect the privacy of Bob. Compared to SVM and M^4, C^2MP^2 divides the whole classifier into two separated classifiers for protecting local private data. We concede that due to the absence of the opposite party's local private data, individual classifiers are biased and yield decision errors. However, we will show in the following that the injection of bias will be compensated by jointly combining the two decision hyperplanes.

For dealing with the nonseparable case, we introduce slack variables. Thus, the optimization of the first classifier is changed into

$$\max_{w_a \neq 0, b_a, \xi_k} \quad \rho_a - C_a \sum_{k=1}^{N^{(A)}} \xi_k \tag{7}$$

$$\text{s.t.} \quad \frac{({w_a}^T x_i^{(A)} + b_a)}{\sqrt{{w_a}^T \Sigma_x w_a}} \geq \rho_a - \xi_i, \quad i = 1, 2, \ldots, N_x^{(A)}, \tag{8}$$

$$\frac{-({w_a}^T y_j^{(A)} + b_a)}{\sqrt{{w_a}^T \Sigma_y w_a}} \geq \rho_a - \xi_{(N_x^{(B)}+j)}, \quad j = 1, 2, \ldots, N_y^{(A)}. \tag{9}$$

In a similar way, the second classifier described by (4)–(6) can be rewritten as

$$\max_{w_b \neq 0, b_b, \varepsilon_k} \quad \rho_b - C_a \sum_{k=1}^{N^{(A)}} \varepsilon_k \tag{10}$$

$$\text{s.t.} \quad \frac{({w_b}^T x_i^{(A)} + b_b)}{\sqrt{{w_b}^T \Sigma_x w_b}} \geq \rho_b - \varepsilon_i, \quad i = 1, 2, \ldots, N_x^{(B)}, \tag{11}$$

$$\frac{-({w_b}^T y_j^{(A)} + b_b)}{\sqrt{{w_b}^T \Sigma_y w_b}} \geq \rho_b - \varepsilon_{(N_x^{(B)}+j)}, \tag{12}$$

$$j = 1, 2, \ldots, N_y^{(B)}, \tag{13}$$

where ξ_k and ε_k are nonnegative slack variables, which can be considered as the degree how the local training data disobey the margin (ρ_a and ρ_b). Functionally, C_a and C_b are positive penalty parameters, thus, $C_a \sum_{k=1}^{N^{(A)}} \xi_k$ and $C_b \sum_{k=1}^{N^{(B)}} \varepsilon_k$ can be conceptually regarded as training errors or the empirical errors. In other words, the two optimizations (7)–(9) and (10)–(12) successfully maximize the minimum margin while minimizing the total training errors respectively.

The above two classifiers in (7)–(9) and (10)–(12) constitute the unkernelized linear version of C^2MP^2. As can be clearly observed, the optimization (7)–(9) is similar to M^4 [11], i.e. this optimization can be cast as a sequential Second Order Cone Programming (SOCP) problem with the $O(N^{(A)}n^3)$ time complexity. The problem defined by (10)–(12) can be solved likewise. Thus the total time complexity for solving the unkernelized linear version of C^2MP^2 is $O(Nn^3)$ which is equal to M^4.

By solving the above two sequential SOCP problems, we can obtain their corresponding solutions $\{{w_a}^*, {b_a}^*\}$ and $\{{w_b}^*, {b_b}^*\}$, then a decision hyperplane protecting privacy of Alice can be represented as $f_a(z) = {{w_a}^*}^T z + {b_a}^*$ and another hyperplane protecting privacy of Bob as $f_b(z) = {{w_b}^*}^T z + {b_b}^*$.

3.2. How collaborative mechanism works

Several natural questions for the linear version of C^2MP^2 are how to get Σ_x and Σ_y, why disclosure of covariance will not disclose the privacy, and how to achieve a final decision hyperplane from the separated $f_x(z)$ and $f_y(z)$. In this section, we address those problems.

The whole positive class data X are horizontally split into $X^{(A)}$ and $X^{(B)}$, let $N_x = N_x^{(A)} + N_x^{(B)}$ denote the total number of X, the mean value of X can be estimated by

$$\begin{aligned}\bar{x} &= \frac{1}{N_x}\sum_{i=1}^{N_x} x_i \\ &= \frac{1}{N_x}\left(N_x^{(A)} \times \bar{x}^{(A)} + N_x^{(B)} \times \bar{x}^{(B)}\right).\end{aligned} \tag{14}$$

The covariance of X can be estimated from

$$\begin{aligned}\Sigma_x &= \frac{1}{N_x}\sum_{i=1}^{N_x}(x_i - \bar{x})(x_i - \bar{x})^T \\ &= \frac{1}{N_x}\left(\sum_{i=1}^{N_x^{(A)}}\left(x_i^{(A)} - \bar{x}\right)\left(x_i^{(A)} - \bar{x}\right)^T\right) \\ &\quad + \frac{1}{N_x}\left(\sum_{j=1}^{N_x^{(B)}}\left(x_j^{(B)} - \bar{x}\right)\left(x_j^{(B)} - \bar{x}\right)^T\right) \\ &= \Sigma_x^{(A)} + \Sigma_x^{(B)}.\end{aligned} \tag{15}$$

As observed form (14) and (15), for obtaining $\bar{x}$, Alice (resp. Bob) can firstly require $N_x^{(B)}$ and $\bar{x}^{(B)}$ (resp. $N_x^{(A)}$ and $\bar{x}^{(A)}$) from Bob (resp. Alice), then Alice and Bob can respectively calculate $\Sigma_x^{(A)}$ and $\Sigma_x^{(B)}$. Finally, combining opponent's component, Σ_x can be shared by both Alice and Bob without disclosure $x_i^{(A)}$ and $x_j^{(B)}$. Σ_y can be calculated likewise.

Some researches have shown that disclosure of statistical values may lead to leakage of privacy [20], in this paper, we focus on preventing opponent to deduce raw data from the covariance matrix. Assuming that Bob wants to deduce $x_i^{(A)}$ from $N_x^{(A)}$, $\bar{x}^{(A)}$ and $\Sigma_x^{(A)}$ that are all information shared from Alice to Bob. From the formula of $\bar{x}^{(A)}$, Bob can build n linear equations with $n \times N_x^{(A)}$ variables. Generally, those equations are linear independent, thus the equations cannot be solved and actual value of $x_i^{(A)}$ cannot be deduced. On the other hands, deducing $x_i^{(A)}$ from $\Sigma_x^{(A)}$ equals to solve the quadratic multivariate(QM) equations with $n(n+1)/2$ equations and $n \times N_x^{(A)}$ variables, this problem is known to be NP-hard over any field [21,22].

For achieving a joint decision from $f_a(z)$ and $f_b(z)$, theoretically inspired from the schemes of combining classifiers [23] and collaborative learning [24], we can consider those points which locate in the margin area and are equally far from $f_a(z)$ and $f_b(z)$ with Mahalanobis distance, a point set will thus be given by

$$\begin{aligned}&\frac{|f_a(z)|}{\sqrt{{w_a}^{*T}\Sigma_x {w_a}^*}} = \frac{|f_b(z)|}{\sqrt{{w_b}^{*T}\Sigma_y {w_b}^*}}, \\ &\operatorname{sign}(f_a(z) f_b(z)) < 0.\end{aligned} \tag{16}$$

The roots of (16) constitute a final hyperplane. If we denote $s_a = \sqrt{{w_a}^{*T}\Sigma_x {w_a}^*}$, $s_b = \sqrt{{w_b}^{*T}\Sigma_y {w_b}^*}$, and $w = s_y w_a^* + s_a w_b^*$, $b = s_b b_b^* + s_a b_b^*$, then the final decision hyperplane can be jointly expressed as

$$f(z) = (s_b {w_a}^* + s_a {w_b}^*)^{\mathrm{T}} z + (s_b b_a + s_a b_b) = w^T z + b. \tag{17}$$

For a future unlabelled point z, one can predict the label of z by evaluating $f(z)$: if $f(z) > 0$, assigns z as positive label, if $f(z) < 0$, assigns z as negative label.

This collaborative mechanism can be considered as a modified median rule of combining classifiers [23], which uses average over Mahalanobis distance instead of arithmetical average. Moreover, Mahalanobis distance takes into account the global information of the data set, including compactness and orientation, so, the final hyperplane combined by this mechanism can achieve a more reasonable decision than individual classifier. Meanwhile, this collaborative strategy can compensate the previously described bias introduced by individual classifiers. Later experimental results on real data sets also demonstrate its effectiveness.

3.3. Connections with other models

In this section, based on the above linear separable version of C^2MP^2, we build the connections between C^2MP^2 and other model.

3.3.1. Connection with M^4

If one assumes $w_a = w_b = w$, $b_a = b_b = b$, $\rho_a = \rho_b = \rho$, (1)–(3) and (4)–(6) can be combined into the following

$$\max_{\rho, w\neq 0, b} \quad \rho \tag{18}$$

$$\text{s.t.} \quad (w^T x_i + b) \geq \rho\sqrt{w^T \Sigma_x w}, \quad i = 1, 2, \ldots, N_x, \tag{19}$$

$$-(w^T y_j + b) \geq \rho\sqrt{w^T \Sigma_y w}, \quad j = 1, 2, \ldots, N_y, \tag{20}$$

(18)–(20) is exactly the M^4 optimization [10,11].

M^4 uses both global and local private data, however, the local private data are directly shared with each other, so, M^4 is a centralized model without privacy-preserving. In comparison, C^2MP^2 is a distributed model which collaboratively deals with local private data and combines two classifiers to achieve the goal of privacy-preserving.

3.3.2. Connection with MPM

Following the path from C^2MP^2 to M^4, adding up all N_x constraints in (19) together and average this sum, one can immediately obtain the following:

$$\begin{aligned}&\left(w^T\sum_{i=1}^{N_x}x_i+N_xb\right)\geq N_x\rho\sqrt{w^T\Sigma_xw}\\&\Rightarrow(w^T\bar{x}+b)\geq\rho\sqrt{w^T\Sigma_xw}.\end{aligned}\tag{21}$$

Similarly, from the N_y constraints in (20), one can obtain

$$-(w^T\bar{y}+b)\geq\rho\sqrt{w^T\Sigma_yw}.\tag{22}$$

Adding up (21) and (22), then the two optimizations can be combined into one optimization, i.e.

$$\begin{aligned}\max_{\rho,w\neq0}\quad&\rho\\\text{s.t.}\quad&w^T(\bar{x}-\bar{y})\geq\rho(\sqrt{w^T\Sigma_xw}+\sqrt{w^T\Sigma_yw}).\end{aligned}\tag{23}$$

Equation (23) is exactly the MPM optimization [9].

Note, the above derivation cannot be reversed, this means that MPM is looser than C^2MP^2. From the viewpoint of privacy-preserving, MPM is a centralized model of C^2MP^2, the centralization takes effect when setting $w_a=w_b=w$, $b_a=b_b=b$ and $\rho_a=\rho_b=\rho$. On the other hand, only global data are used and all local private data are ignored in MPM, this inobservance may cause inaccurate. Although of those discussions, we must emphasize that the original goal of the MPM is not for privacy-preserving but to provide guarantees with respect to classification accuracy, here, we only explore this model freshly from the viewpoint of privacy-preserving.

3.3.3. Connection with SVM

Under the same assumptions that $w_a=w_b=w$, $b_a=b_b=b$, $\rho_a=\rho_b=\rho$ and a additional assumption $\Sigma_x=\Sigma_y=\Sigma$, one can combine (1)–(3) and (4)–(6) together, one can obtain

$$\begin{aligned}\max_{\rho,w\neq0,b}\quad&\rho&(24)\\\text{s.t.}\quad&(w^Tx_i+b)\geq\rho\sqrt{w^T\Sigma w},\\&i=1,2,\ldots,N_x,&(25)\\&-(w^Ty_j+b)\geq\rho\sqrt{w^T\Sigma w},\\&j=1,2,\ldots,N_y.&(26)\end{aligned}$$

Notice that magnitude of w will not influence the optimization, one can set $\rho\sqrt{w^T\Sigma w}=1$ without loss of generality. Additionally, if one assumes $\Sigma=I$, where I is the unit matrix, then (24)–(26) becomes

$$\begin{aligned}\min_{w\neq0,b}\quad&w^Tw&(27)\\\text{s.t.}\quad&(w^Tx_i+b)\geq1,\quad i=1,2,\ldots,N_x,&(28)\\&-(w^Ty_j+b)\geq1,\quad j=1,2,\ldots,N_y.&(29)\end{aligned}$$

Equations (27)–(29) exactly mean the standard SVM model.

Assuming $w_a=w_b=w, b_a=b_b=b, \rho_a=\rho_b=\rho$ and $\Sigma_x=\Sigma_y=\Sigma$, means SVM is also a centralized model of C^2MP^2. Assuming $\Sigma=I$ means SVM discards orientation or shape information [11], and uses only local private data. So, SVM can be considered as a centralized model without privacy-preserving.

It is worth stressing that the goal of this paper is neither to beat M^4, MPM or SVM from the point view of classification accuracy nor to design a novel cryptosystem, we only try to explore applicability of cooperating global and local data for privacy-preserving based on the M^4 framework.

As the end of this section, we summarize those differences and connections among the four models in Table 1.

Table 1. Differences and connections between C^2MP^2, MPM, SVM and M^4.

Classifier	Capability of privacy preserving	Using global data	Using local data	Distributed (Yes) or Centralized (No)
C^2MP^2	Yes	Yes	Yes	Yes
MPM	Yes	Yes	No	No
SVM	No	No	Yes	No
M^4	No	Yes	Yes	No

3.4. Secure training algorithm for linear version of C^2MP^2

As previously, the theoretical analysis is on $\mathbb{R}$, while for modular operation in testing and computing Gram matrix, all values should be in a bounded region before employing the training algorithm. Usually, attributes are presented with fixed precision floats, we can encode them as integers by scaling each attribute to the range $[-M, M]$. Of course, we have to use the same method to scale testing data before testing.

We then introduce Algorithm 1 which states the training procedure of the linear version of C^2MP^2 and in which the communication between Alice and Bob is considered.

After running Algorithm 1, both parties can compute the final output from the received messages, mean and covariance of both sides. Consequently, neither of them can learn additional information besides the mean, the covariance, and the decision function (17).

Algorithm 1 Training algorithm for linear version of C^2MP^2

Input: Alice with $X^{(A)}$, $Y^{(A)}$ and Bob with $X^{(B)}$, $Y^{(B)}$

1. Alice sends $\{N_x^{(A)}, \bar{x}^{(A)}\}$ and $\{N_y^{(A)}, \bar{y}^{(A)}\}$ to Bob
2. Bob sends $\{N_x^{(B)}, \bar{x}^{(B)}\}$ and $\{N_y^{(B)}, \bar{y}^{(B)}\}$ to Alice
3. Alice and Bob independently calculate $\{\bar{x}, \bar{y}\}$ by (14)
4. Alice and Bob independently calculate $\{\Sigma_x, \Sigma_y\}$ by (15)
5. Alice obtains $\{w_b{}^*, b_b{}^*\}$ by solving (10)-(12)
6. Bob obtains $\{w_a{}^*, b_a{}^*\}$ by solving (7)-(9)
7. Alice sends $\{w_b{}^*, b_b{}^*\}$ to Bob
8. Bob sends $\{w_a{}^*, b_a{}^*\}$ to Alice
9. Alice and Bob compute the final decision hyperplane (17)

3.5. Secure testing algorithm for linear version of C^2MP^2

Once the final decision hyperplane (17) is securely constructed for each party, to predict a future point z held by Tom using (17), we need to guarantee that z does not be disclosed to any other party.

When z is held by either Alice or Bob, because (17) is equally shared by them, testing can be computed only in the party who holds z and no exchanging data is needed, in this scenario, testing is naturally secure. However, for commercial interests or protecting intellectual property, (17) can be considered as classification knowledge rule, so, Alice and Bob do not want to disclose this rule to Tom. For example, an online anti-spam mail provider does not open its classification rule to public, while distinguishing between spam and normal email for customer without revealing the personal privacy.

Based on the existing secure scalar product protocol [18] and the Paillier homomorphic cryptosystem [19], we propose Algorithm 2 to compute scalar production $w \cdot z$ and then predict the label of z. Consider the equality on sharing (17) with Alice and Bob, we can assume that the testing service is provided by Alice.

Algorithm 2 can be seen as a special case of Protocol3 proposed by Goethals et al.[18] on the condition of $S_b \leftarrow 0$. Goethals et al. [18] has given a formal security proof in the semi-honest model. Here we apply this protocol to executing our testing procedure.

After executing Algorithm 2, Alice obtains no more knowledge than $w \cdot z$ and the predicted label of z, Tom obtains no new knowledge than the predicted label of z.

Our proposed Algorithm 2 as well as PP-SVM [25] borrows the same idea from the scalar production protocol [18]. However, a semi-honest third party is avoided in our model, while in PP-SVM model, it works for both testing and training.

Algorithm 2 Testing algorithm for linear version of C^2MP^2

Input: Service provider Alice holds w and secure homomorphic encryption system keypair (sk, pk)

Input: Customer Tom holds a future point z

Output: Tom receives the predicted label of z

1. **for** $k = 1 \cdots n$ **do**
2. Alice generates a random nonce r_k
3. Alice computes $c_k \leftarrow Enc_{pk}(w_k; r_k)$ and sends c_k to Tom
4. **end for**
5. Tom computes $C \leftarrow \prod_{k=1}^{n} {c_k}^{z_k}$
6. Tom sends C to Alice
7. Alice computes $Dec_{sk}(C) = w \cdot z$
8. Alice computes $sign(w \cdot z + b)$ to predict the label of z and sends the label to Tom

4. Kernelization

In order to handle nonlinear classification problems, we seek to use the kernelization trick to map the n-dimensional data points into a high-dimensional feature space $\mathbb{R}^f$ via a mapping function $\varphi : \mathbb{R}^n \mapsto \mathbb{R}^f$, i.e. $x_i \mapsto \varphi(x_i), y_i \mapsto \varphi(y_i), i = 1, \ldots, N_x, j = 1, \ldots, N_y$, then, a linear hyperplane $f_k(z) = a^T\varphi(z) + b_k$ in space $\mathbb{R}^f$ corresponds to a nonlinear hyperplane in the original space $\mathbb{R}^n$ where $a, \varphi(z) \in \mathbb{R}^f, z \in \mathbb{R}^n, b \in \mathbb{R}$. We will demonstrate that, although C^2MP^2 possesses a significantly different optimization from MPM, the kernelization method used in [9] is still viable, provided that suitable estimates for means and covariance matrices are applied therein. This method has been also extended to MEMPM [26] and M^4 [11]. The similar kernelization method for C^2MP^2 is described in the following.

After being kernelized, (1)–(3) in the feature space can be written as

$$\max_{\alpha \neq 0, b_a} \quad \rho_a \tag{30}$$

$$\text{s.t.} \quad \frac{(\alpha^T\varphi(x)_i^{(A)} + b_a)}{\sqrt{\alpha^T\Sigma_{\varphi(x)}^{(A)}\alpha}} \geq \rho_a, \quad i = 1, 2, \ldots, N_x^{(A)}, \tag{31}$$

$$\frac{-(\alpha^T\varphi(y)_j^{(A)} + b_a)}{\sqrt{\alpha^T\Sigma_{\varphi(y)}^{(A)}\alpha}} \geq \rho_a, \quad j = 1, 2, \ldots, N_y^{(A)}. \tag{32}$$

Equations (4)–(6) can be also written as

$$\max_{\beta \neq 0, b_b} \quad \rho_b \tag{33}$$

$$\text{s.t.} \quad \frac{(\beta^T\varphi(x)_i^{(B)} + b_b)}{\sqrt{\beta^T\Sigma_{\varphi(x)}^{(B)}\beta}} \geq \rho_b,$$

$$i = 1, 2, \ldots, N_x^{(B)}, \tag{34}$$

$$\frac{-(\beta^T \varphi(y)_j^{(B)} + b_b)}{\sqrt{\beta^T \Sigma_{\varphi(y)}^{(B)} \beta}} \geq \rho_b,$$

$$j = 1, 2, \ldots, N_y^{(B)}, \tag{35}$$

where $\Sigma_{\varphi(x)}^{(A)}$ and $\Sigma_{\varphi(y)}^{(A)}$ are the covariance matrices of $X^{(A)}$ and $Y^{(A)}$ in the feature space. Likewise for $\Sigma_{\varphi(x)}^{(B)}$ and $\Sigma_{\varphi(y)}^{(B)}$. To carry out the above two optimizations, we need to reformulate them and their final decision hyperplane in term of a given inner product kernel function $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ satisfying Mercer's conditions. We now state Algorithm 4.1 similar to Corollary 5 proposed by Lanckriet *et al.* [9] and Proposition 1 proposed by Huang et al. [11] and prove its validity in solving the kernelized C^2MP^2 model.

Corollary 4.1: *In feature space $\mathbb{R}^f$, on the side of Alice, let $\{\varphi(x_i)\}_{i=1}^{N_x^{(A)}}$ and $\{\varphi(y_j)\}_{j=1}^{N_y^{(A)}}$ be the training data points corresponding to positive class and negative class respectively. If the estimates of means and covariance matrices are given as*

$$\overline{\varphi(x)}^{(A)} = \sum_{i=1}^{N_x^{(A)}} \lambda_i \varphi(x_i)^{(A)},$$

$$\overline{\varphi(y)}^{(A)} = \sum_{j=1}^{N_y^{(A)}} \theta_j \varphi(y_j)^{(A)},$$

$$\Sigma_{\varphi(x)}^{(A)} = \tau_x I_f + \sum_{i=1}^{N_x^{(A)}} \Lambda_i \left(\varphi(x_i)^{(A)} - \overline{\varphi(x)}^{(A)}\right) \left(\varphi(x_i)^{(A)} - \overline{\varphi(x)}^{(A)}\right)^{\mathrm{T}},$$

$$\Sigma_{\varphi(y)}^{(A)} = \tau_y I_f + \sum_{j=1}^{N_y^{(A)}} \Theta_j \left(\varphi(y_j)^{(A)} - \overline{\varphi(y)}^{(A)}\right) \left(\varphi(y_j)^{(A)} - \overline{\varphi(y)}^{(A)}\right)^{\mathrm{T}},$$

where I_f is the identity matrix of dimension f, λ_i, θ_j, Λ_i and Θ_j are the normalized weights for data points $\{\varphi(x_i)^{(A)}\}_{i=1}^{N_x^{(A)}}$ and $\{\varphi(y_j)^{(A)}\}_{j=1}^{N_y^{(A)}}$ respectively. The positive constants τ_x and τ_y can be regarded as the regularization term of covariance matrices. Then, the optimal α^* and β^* for (30)–(32) lie in the space spanned by the training points, i.e. $\alpha^* \in span(\{\varphi(x_i)^{(A)}\}_{i=1}^{N_x^{(A)}}, \{\varphi(y_j)^{(A)}\}_{j=1}^{N_y^{(A)}})$.

Proof: We write $\alpha = \alpha_d + \alpha_p$, where α_d is the projection of α in the vector space spanned by all training data points and α_p is the orthogonal component to this span space. We can then easily check that the denominator of left part in (31) can be changed to

$$\alpha^T \Sigma_{\varphi(x)}^{(A)} \alpha = \alpha_d^T \Bigg(\sum_{i=1}^{N_x^{(A)}} \Lambda_i \left(\varphi(x_i)^{(A)} - \overline{\varphi(x)}^{(A)}\right) \left(\varphi(x_i)^{(A)} - \overline{\varphi(x)}^{(A)}\right)^{\mathrm{T}} + \Bigg) \alpha_d + \tau_x \left(\alpha_d^T \alpha_d + \alpha_p^T \alpha_p\right). \tag{36}$$

For the orthogonality, there are $\alpha_p^T \varphi(x_i)^{(A)} = 0$ for $i = 1, 2, \ldots, N_x^{(A)}$, $\alpha_p^T \varphi(y_j)^{(A)} = 0$ for $j = 1, 2, \ldots, N_y^{(A)}$ and $\alpha_d^T \alpha_p = 0$. It is evident that an orthogonal component α_p of α will not affect the constraints (30) and (32). Since the objective is to be maximized, the denominators $\alpha^T \Sigma_{\varphi(x)}^{(A)} \alpha$ should be as small as possible, this will lead to $\alpha_p{}^* = 0$, hence $\alpha^* = \alpha_d{}^*$. In other words, the optimal α^* lies in the vector space spanned by all the training points, i.e. $\alpha^* \in span(\{\varphi(x_i)^{(A)}\}_{i=1}^{N_x^{(A)}}, \{\varphi(y_j)^{(A)}\}_{j=1}^{N_y^{(A)}})$. ■

According to Algorithm 4.1, α can be written as a linear combinations form of training data points

$$\alpha = \sum_{i=1}^{N_x^{(A)}} \mu_i \varphi(x_i)^{(A)} + \sum_{j=1}^{N_y^{(A)}} \nu_j \varphi(y_i)^{(A)},$$

where the coefficients $\mu_i, \nu_j \in \mathbb{R}$. Represented by vector form

$$\eta^{(A)} = [\mu_1, \mu_2, \ldots, \mu_{N_x}, \nu_1, \nu_2, \ldots, \nu_{N_y}]^T.$$

For the purpose of clarity, let $\{z_i\}_{i=1}^{N^{(A)}}$ denote all $N^{(A)}$ training data points held by Alice, where

$$z_i = x_i, \quad i = 1, 2, \ldots, N_x^{(A)},$$

$$z_{j+N_x^{(A)}} = y_j, \quad j = 1, 2, \ldots, N_y^{(A)}.$$

Following aforementioned denotation, let K_i denote the ith row vector, where $K_i \in \mathbb{R}^{N^{(A)}}$ and $i = 1, 2, \ldots, N^{(A)}$, moreover, let K_x and K_y denote the first $N_x^{(A)}$ rows and the last $N_y^{(A)}$ rows respectively:

$$K^{(A)} := (K_{i,j}) = \begin{pmatrix} K_x \\ K_y \end{pmatrix}.$$

If we use the plug-in estimates to approximate the means and covariance matrices, we can write plug-in

estimated covariance matrices as

$$\hat{\Sigma}_{\varphi(x)}^{(A)} = \frac{1}{N_x^{(A)}} \sum_{i=1}^{N_x^{(A)}} \left(\varphi(x_i)^{(A)} - \overline{\varphi(x)}^{(A)}\right) \left(\varphi(x_i)^{(A)} - \overline{\varphi(x)}^{(A)}\right)^{\mathrm{T}},$$

$$\hat{\Sigma}_{\varphi(y)}^{(A)} = \frac{1}{N_y^{(A)}} \sum_{j=1}^{N_y^{(A)}} \left(\varphi(y_j)^{(A)} - \overline{\varphi(y)}^{(A)}\right) \left(\varphi(y_j)^{(A)} - \overline{\varphi(y)}^{(A)}\right)^{\mathrm{T}}.$$

In order to represent the covariance matrix into an inner product form, we then define $M^{(A)}$ as

$$M^{(A)} := \begin{pmatrix} \sqrt{N_x^{(A)}} M_x \\ \sqrt{N_y^{(A)}} M_y \end{pmatrix} = \begin{pmatrix} K_x - 1_{N_x^{(A)}} m_x^T \\ K_y - 1_{N_y^{(A)}} m_y^T \end{pmatrix},$$

where $m_x, m_y \in \mathbb{R}^{N^{(A)}}$, whose elements are defined as

$$[m_x]_i := \frac{1}{N_x^{(A)}} \sum_{i=1}^{N_x^{(A)}} K_i, \quad i = 1, 2, \ldots, N_x^{(A)},$$

$$[m_y]_j := \frac{1}{N_y^{(A)}} \sum_{i=N_x^{(A)}+1}^{N^{(A)}} K_i, \quad j = 1, 2, \ldots, N_y.$$

Unit vectors $1_{N_x^{(A)}} \in \mathbb{R}^{N_x^{(A)}}, 1_{N_y^{(A)}} \in \mathbb{R}^{N_y^{(A)}}$. Consequently, covariance matrices can be represented as

$$\hat{\Sigma}_{\varphi(x)}^{(A)} = {M_x}^T M_x, \tag{37}$$

$$\hat{\Sigma}_{\varphi(y)}^{(A)} = {M_y}^T M_y. \tag{38}$$

Notice that, in (36), if set $\tau_x = 0$, the objective (30) and the constraints (31) and (32) will not be affected. So, we can set $\tau_x = 0$ and $\tau_y = 0$, and substitute (37) and (38) into (31) and (32) respectively. Finally, the first classifier of kernelized C^{2}MP^2 can be written as the following:

$$\max_{\eta^{(A)} \neq 0, b_a} \quad \rho_a \tag{39}$$

$$\text{s.t.} \quad \frac{({\eta^{(A)}}^T K_i + b_a)}{\sqrt{{\eta^{(A)}}^T {M_x}^T M_x \eta^{(A)}}} \geq \rho_a, \quad i = 1, 2, \ldots, N_x^{(A)}, \tag{40}$$

$$\frac{-({\eta^{(A)}}^T K_j + b_a)}{\sqrt{{\eta^{(A)}}^T {M_y}^T M_y \eta^{(A)}}} \geq \rho_a, \quad j = 1, 2, \ldots, N_y^{(A)}. \tag{41}$$

To solve this problem, the optimal ${\eta^{(A)}}^*$ and ${b_a}^*$ can be obtained. Similarly, the second classifier can be solved with its optimal solution ${\eta^{(B)}}^*$ and ${b_b}^*$. The optimal decision hyperplane can be represented as a linear form in kernel space

$$f_k(z) = {\eta^{(A)*}}^T K_z^{(A)} + {\eta^{(B)*}}^T K_z^{(B)} + {b_a}^* + {b_b}^*, \tag{42}$$

where $[K_z^{(A)}]_i = K(z_i^{(A)}, z)$ for $i = 1, 2, \ldots, N^{(A)}$ and $[K_z^{(B)}]_j = K(z_j^{(B)}, z)$ for $j = 1, 2, \ldots, N^{(B)}$. This combining operation can be considered as learning with hyperkernels [27].

Computing K_{ij} involves an inner product computation, consider the Gaussian kernel, K_{ij} can be presented as $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|) = \exp(-\gamma |x_i \cdot x_i - 2x_i \cdot x_j + x_j \cdot x_j|)$, so the secure scalar product protocol [18] is still viable for testing future points in kernel space.

Table 2. Data sets used in the experiments.

Data set	N_x	N_y	n
ionosphere	225	126	34
glass	70	76	9
musk	207	209	167
parkinsons	48	147	22
pima	268	500	8
sonar	97	111	60
vote	168	267	16
yeast	463	429	6

5. Experiments

In this section, we evaluate C^{2}MP^2 and compare the performance of C^{2}MP^2 with that of SVM on eight benchmark data sets. The covariance matrices are given by the plug-in estimates.

5.1. Evaluations on benchmark data sets

We next perform evaluation on eight benchmark data sets obtained from the University of California at Irvine (UCI) machine learning repository [28]. To evaluate the algorithms in the horizontally distributed scenario, we need to construct a training set which are horizontally split into two sides(Alice and Bob). To this end, for each data set, 70 percent of the data examples are randomly selected for training, and one random half held by Alice and another half held by Bob. All the remaining data are used for test. This simulates the situation that data set is horizontally distributed in two parties. Further details of these data sets are listed in Table 2.

We compared C^{2}MP^2 with SVM on unkernelized linear version and kernelized version with Gaussian kernel respectively. All parameters including penalty parameters C and the width parameters γ in the Gaussian kernel for all three models were tuned via grid search [29] using the tenfold cross-validation, i.e. $C = 2^{-5}, 2^{-3}, \ldots, 2^{15}$ and $\gamma = 2^{-15}, 2^{-13}, \ldots, 2^3$.

We randomly split each data set into training and test sets with the above scheme. Then, the fivefold

Table 3. Comparisons of classification accuracies on UCI data sets between unkernelized linear versions of C^2MP^2 and SVM.

	NO Kernelization		
Data set	C^2MP^2	SVM	*p*-Value
ionosphere (%)	**90.56±5.14**	85.90 ±5.32	*0.0339*
glass (%)	**76.85±2.04**	72.00 ±7.24	*0.0412*
musk (%)	**90.45 ±1.51**	87.25±4.17	*0.0195*
parkinsons (%)	86.34±2.89	**89.49±4.38**	*0.0385*
pima (%)	**80.31 ±1.32**	77.68±3.33	*0.0459*
sonar (%)	**79.88 ±1.79**	76.00±4.46	*0.0409*
vote (%)	**96.85 ±2.63**	94.31±3.07	*0.0380*
yeast (%)	**65.15 ±2.55**	63.14 ±2.87	*0.0305*

Table 4. Comparisons of classification accuracies on UCI data sets between kernelized versions of C^2MP^2 and SVM.

	Gaussian kernel		
Data set	C^2MP^2	SVM	*p*-Value
ionosphere (%)	90.86±3.88	**95.19±4.20**	*0.0391*
glass (%)	76.83±10.97	**79.63±1.13**	*0.4436*
musk (%)	**95.84±1.93**	93.29±3.01	*0.0299*
parkinsons (%)	**97.29±3.05**	94.38±2.25	*0.0334*
pima (%)	**79.82±3.40**	76.93±2.31	*0.0447*
sonar (%)	**92.35±3.45**	88.62±2.24	*0.0212*
vote (%)	**97.15±1.90**	94.52±3.19	*0.0397*
yeast (%)	64.03±3.70	**66.04±6.92**	*0.4325*

cross-validation is performed on the training set for parameter selection. Using the tuned parameters, the experiment is then repeated 10 times independently on each data set, and the averages accuracy and standard deviations are summarized in Tables 3 and 4. Furthermore, the paired *t*-test on 0.05 significance level is performed over the 10 accuracies of each data set and the corresponding *p*-value are also listed.

For the purpose of clarity, we separately analyse the results in unkernelized linear versions and in kernelized nonlinear versions. As can be seen in Table 3, in comparison with SVM, C^2MP^2 achieves the best overall performance, it loses only on *parkinsons*, in which the number of total positive class data is 48, after being horizontally split into two parts by the above scheme, each party only holds 16 training samples, to train a classifier by those less samples may lead to inaccuracy.

In the kernelized version with Gaussian kernel, as can be seen in Table 4, C^2MP^2 wins five out of eight, and is significantly better on *musk*, *parkinsons*, *pima*, *sonar* and *vote*. Although the linear C^2MP^2 wins on *ionosphere*, *glass* and *yeast*, the kernelized C^2MP^2 loses on these data sets. Those differences may be caused by the approximation errors introduced by plug-in estimates of the covariance matrices in their kernelized feature space, in which data points are very sparse, compared with the huge dimensionality in the Gaussian kernel.

5.2. Computation and communication cost

The training linear C^2MP^2 has $O(Nn^3)$ time complexities which is equal to that of M^4. The communication cost of training linear C^2MP^2 is also quite low. In Algorithm 1, step 1 transmits (n^2+1) elements, and step 7 transmits n elements, due to symmetry of communication, the total number of communication messages is $2(n^2+n+1)$ that is *independent* of the number of samples.

Assuming the dimension of samples is n and the max length of each sample is m bits, the communication overhead of Algorithm 2 is $(n+2)m$ bits. In executing this algorithm, Alice must perform n encryptions. Bob has to perform n exponentiations, n multiplications and 1 decryption. The current hardware allows to do approximately 10^6 multiplications per seconds and thus the computational complexity of both Alice and Bob is tolerable.

6. Conclusion

We have proposed a novel two-party privacy-preserving classification solution called Collaborative Classification Mechanism for Privacy-preserving (C^2MP^2) which theoretically based on combing classifiers and practically respected the fact that ones privacy should be shielded only from others and free accessed by oneself, and that sharing global data with others will not disclose ones own privacy. Based on local and global learning theory, two local classifiers are constructed without revealing one's privacy, then they are combined to give a joint decision. From the viewpoint of privacy-preserving, we have also established detailed connections among our model and other models including M^4, MPM and SVM. Moreover, we have designed a training algorithm and a testing algorithm to securely carry out our model without disclosing ones privacy to any others. In addition, we have extended our model to a nonlinear classification approach by exploiting kernel trick. Experimental results on benchmark data sets have demonstrated the advantages of C^2MP^2 in privacy-preserving. How to extend C^2MP^2 to multi-party classifications is also an important future topic.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Natural Science Foundation of PR China [grant number 61772237]; The Fundamental Research Funds for the Central Universities [grant number JUSRP51618B]; The Equipment Development and the Ministry of Education union fund [grant number 6141A02033312]; The Natural Science Foundation of Jiangsu Province, China [grant number BK20151358] and [grant number BK20151202]; The Suzhou Science and Technology project [grant number SYG201702].

References

[1] OECD, Guidelines on the protection of privacy and transborder flows of personal data. OECD, editor. OECD Publishing, 17 March 2005.

[2] OECD, Guidelines for consumer protection in the context of electronic commerce. OECD, editor. OECD Publishing, 15 March 2000.

[3] HIPAA:Health Insurance Portability and Accountability Act, in Fed Regist 66(40), 1996. [Online]. Available: http://www.hipaa.org.

[4] HITECH Breach Notification Guidance and Request for Public Comment, Federal Register, Department of Health and Human Services Rules and Regulations 79, April 2009. [Online]. Available: http://www.hhs.gov/ocr/privacy/hipaa/understanding/coveredentities/federalregisterbreachrfi.pdf.

[5] Chen K, Liu L. Privacy preserving data classification with rotation perturbation, November 2005, p. 4.

[6] Zhong Z, Wright R. Privacy-preserving classification of customer data without loss of accuracy. SIAM International Conference on Data Mining (SDM), Newport Beach. Citeseer, 2005.

[7] Yu H, Vaidya J, Jiang X. Privacy-preserving SVM classification on vertically partitioned data, vol. 3918 LNAI, Singapore, 2006, p. 647–656.

[8] Ho T. Privacy preserving frequency mining in 2-part fully distributed setting. IEICE Trans Inf Syst. 2010;93:2702–2708.

[9] Lanckriet G, El Ghaoui L, Bhattacharyya C, et al. A robust minimax approach to classification. J Mach Learn Res. 2003;3:555–582.

[10] Huang K, Yang H, King I, et al.Local learning vs. global learning: an introduction to maxi–min margin machine. Support Vector Machines: Theory and Applications, 2005, p. 113–132.

[11] Huang K, Yang H, King I, et al. Maxi–min margin machine: learning large margin classifiers locally and globally. IEEE Trans Neural Networks. 2008;19(2):260–272.

[12] Deng Z, Chung F-L, Wang S. A new minimax probability based classifier using fuzzy hyper-ellipsoid. International Joint Conference on Neural Networks, 2007, IJCNN 2007, August 2007, p. 2385–2390.

[13] Chung F-L, Deng Z, Wang S. From minimum enclosing ball to fast fuzzy inference system training on large datasets. IEEE Trans Fuzzy Syst. 2009;17(1):173–184.

[14] Tsang I, Kwok J, Cheung P. Core vector machines: fast SVM training on very large data sets. J Mach Learn Res. 2006;6(1):363.

[15] Bertino E, Sandhu R. Database security-concepts, approaches, and challenges. IEEE Trans Dependable Secure Comput. 2005;2(1):2–19.

[16] Goldreich O. Foundations of cryptography. Vol. 2. New York (NY): Cambridge University Press; 2004.

[17] Shen C-H, Zhan J, Hsu T-S, et al. Scalar-product based secure two-party computation. IEEE International Conference on Granular Computing, 2008, GrC 2008, 2008, p. 556–561.

[18] Goethals B, Laur S, Lipmaa H, et al. On private scalar product computation for privacy-preserving data mining. Proceedings of the Seventh Annual International Conference in Information Security and Cryptology, LNCS. Springer-Verlag; 2004, p. 104–120.

[19] Paillier P. Public-key cryptosystems based on composite degree residuosity classes. Lecture Notes in Computer Science, 1999, p. 223–238.

[20] Cynthia Dwork KNAS, McSherry F. Calibrating noise to sensitivity in private data analysis. TTC, 2006, p. 265–284.

[21] Courtois N, Klimov A, Patarin J, et al. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. Lecture Notes in Computer Science, 2000, p. 392–407.

[22] Courtois N, Goubin L, Meier W, et al. Solving underdefined systems of multivariate quadratic equations. Lecture Notes in Computer Science, 2002, p. 211–227.

[23] Kittler J, Hatef M, Duin R, et al. On combining classifiers. IEEE Trans Pattern Anal Mach Intell. 1998 Mar;20(3):226–239.

[24] Pedrycz W, Rai P. A multifaceted perspective at data analysis: a study in collaborative intelligent agents. IEEE Trans Syst Man Cybern Part B: Cybern. 2009;39(4):834–844.

[25] Vaidya J, Yu H, Jiang X. Privacy-preserving SVM classification. Knowl Inf Syst. 2008 Feb.;14(2):161–178.

[26] Huang K, Yang H, King I, et al. The minimum error minimax probability machine. J Mach Learn Res. 2004;5:1253–1286.

[27] Ong C, Smola A, Williamson R, et al. Learning the kernel with hyperkernels. J Mach Learn Res. 2005;6(07):1043–1071.

[28] Asuncion A, Newman D. UCI machine learning repository. 2007. [Online]. Available: http://www.ics.uci.edu/mlearn/MLRepository.html.

[29] Chang C-C, Lin C-J. LIBSVM: a library for support vector machines, 2001, software available at http://www.csie.ntu.edu.tw/cjlin/libsvm.