PAPER

A New Hybrid Approach for Privacy Preserving Distributed Data Mining

Chongjing SUN[†a)], *Member*, Hui GAO[†b)], Junlin ZHOU[†c)], Yan FU[†d)], *and* Li SHE[††], *Nonmembers*

SUMMARY With the distributed data mining technique having been widely used in a variety of fields, the privacy preserving issue of sensitive data has attracted more and more attention in recent years. Our major concern over privacy preserving in distributed data mining is the accuracy of the data mining results while privacy preserving is ensured. Corresponding to the horizontally partitioned data, this paper presents a new hybrid algorithm for privacy preserving distributed data mining. The main idea of the algorithm is to combine the method of random orthogonal matrix transformation with the proposed secure multi-party protocol of matrix product to achieve zero loss of accuracy in most data mining implementations.

key words: *data mining, privacy preserving, secure multi-party computation, orthogonal transformation, Inner product operation*

1. Introduction

With more computer processing power, continuous development of storage technology and fast growth of the internet, vast amounts of data have been accumulated by all walks of life in recent years. To guide appropriate decision making, people desperately need some technology to explore potential knowledge (model or rules) from these data. Data mining [1], [2] is seen as an increasingly important data analysis tool to meet the requirement. It is currently used in a wide range of profiling practices, such as marketing, surveillance, fraud detection, anti-terrorism, and scientific discovery. Due to the releasing of unprocessed raw data in the traditional data mining, the privacy of data would be revealed before the public. In 1996, a survey [3] of net-users' attitudes towards personal privacy shows that in the United States 17% of Internet users are not willing to provide information to web sites regardless of the privacy-preserving policy taken, 56% of investigators are willing to provide their information to web sites only if some privacy-preserving measures are taken. Since then, a number of research works [4] on how to protect personal privacy while ensuring the accuracy of data mining results have been studied.

Bertino et al. [5] defined privacy as an entity that can prevent unauthorized people from getting sensitive information or the characteristics of the data (model, rules, etc.) from an electronic database. According to this definition, privacy can be divided into two categories: 1) Personal privacy: mainly specific individuals' data items, such as personal medical records, etc. 2) Public privacy: the knowledge which underlies the original data, or the high-level model information which gets from the original data. Such as the association rules which get from an analysis of shopping basket data.

When releasing data for research purposes, one needs to limit disclosure risks to an acceptable level while maximizing data utility. Research in privacy preserving data mining started after 2000, and it mostly focuses on the following two issues: 1) How to prevent the leakage of privacy in the process of data mining; 2) How to retain the accuracy of the data mining results even if privacy preserving is taken. Most of the current privacy preserving methods affect more or less the accuracy of data mining results, and according to their employed techniques they can be roughly classified into three groups.

The first group of privacy preserving algorithms is based on the randomization technique, which uses data distortion methods. In most cases, the individual records cannot be recovered, but only aggregate distributions can be recovered and used for data mining purposes. Typical algorithms include additive data perturbation [6], matrix multiplicative data perturbation [7], data swapping [8], data blocking [9]. The randomization method is a simple technique which can be easily implemented at data collection time. The second group of privacy preserving algorithms is based on data anonymization, which constructs groups of anonymous records that are transformed in a group-specific way. Typical techniques are k-Anonymity [10], l-Diversity [11] and t-Closeness [12]. The third group of privacy preserving algorithms is based on data encryption, which uses cryptographic approaches to minimize the information shared. This group of algorithms is usually designed for distributed data mining to prevent any party from knowing the actual value of local sensitive items. Basic techniques include Secure Multiparty Computation (SMC), Homomorphic Encryption.

In the three groups of privacy-preserving algorithms discussed above, only the third one can guarantee the accuracy of the data mining while providing a relatively high level of privacy preserving at the same time. However, they may involve more redundant communication and heavy computation. The first group of algorithms introduces low

Manuscript received July 23, 2013.
Manuscript revised November 19, 2013.

[†] The authors are with Web Science Center, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China.

[††] The author is with Alibaba Business College, Hangzhou Normal University, Hangzhou 311121, China.

a) E-mail: chingsun00@gmail.com
b) E-mail: huigao@uestc.edu.cn (Corresponding author)
c) E-mail: jlzhou@uestc.edu.cn
d) E-mail: fuyan@uestc.edu.cn

DOI: 10.1587/transinf.E97.D.876

computation and can be implemented easily, but may greatly lose the accuracy of data mining results, or have to make a compromise between the accuracy of data mining results and the level of privacy preserving. As for the second group of algorithms, there is a certain extent of data defects and privacy disclosure, and a heavy computation in order to optimize the data anonymization. More detailed comparison between different privacy protection and data mining algorithms can be found in [4].

In the existing solutions of distributed data mining, there are mainly two types of data partitioning among the parties [4]: data mining over horizontal partitioned data and data mining vertical partitioned data. In a vertical partition approach, the attributes (columns) of the same vectors (rows) are split across the partitions. Each partition has unique columns - with the exception of the key column, which is common to all partitions. In a horizontal partition approach, different rows are described with the same schema in all partitions. Each partition has unique rows, while all unique rows have the same attributes (columns). The idea behind these distributed solutions is that two or more participants want to conduct a computation based on their private partitions.

Privacy-preserving distributed data mining algorithms require participants to collaborate with others to compute the results, while provably preventing the disclosure of any information except the data mining results. Oliveira [13], [14] and Liu [15] proposed the privacy preserving algorithms based on the random projection theory. Random projection has recently emerged as a powerful method for dimensionality reduction, and the distances between the objects (records) can be approximately preserved after random projection. Oliveira extended this method for preserving clustering from centralized environment to distributed environment with data vertically partitioned. Liu [15] found that the privacy preserving technique based on random projection can be successfully applied to different kinds of data mining tasks, including inner product/Euclidean distance estimation, correlation matrix computation, clustering, outlier detection, linear classification, etc. The above mentioned methods can effectively preserve the privacy through matrix dimensions reduction. After perturbation by these methods, the distance-related statistical properties of original data can be well maintained. For example, the distance between two perturbed vectors will be very close to the distance between two original vectors.

Oliveira and Zaïane [16], [17], Chen and Liu [18] applied the geometric data transformation methods for privacy-preserving distance-based clustering and classification algorithms. Data distortion using random geometric transformation can maintain the distance between the records, and achieve the goal of privacy preserving data mining. Scaling, translation, and rotation were proposed in [16] for preserving the sensitive data information. Oliveira [17] proposed Pairwise-Attribute rotation method for distorting the original data, and this method can set different Pairwise-Security Threshold for each attribute pair. Chen and Liu [18] presented an algorithm based on random orthogonal matrix transformation for privacy preserving classification, and gave an approach providing high privacy guarantee while maintaining zero-loss accuracy. However, this method only focused on the privacy preserving in local data mining.

In this paper, we focus on the preservation of personal privacy and present a new hybrid algorithm for privacy preserving distributed data mining. We assume all data is horizontally partitioned and all participants are semi-honest adversaries, which follow the protocol faithfully, but can try to infer the secret information of other participants from the data they see during the execution of the protocol. Because of the low-cost computation of the privacy preserving technology based on data distortion and the nondestructive data mining results of the technology based on SMC, we use tools from data distortion and SMC domains. Since distance metric gives a numerical value that measures the similarity between two data objects, it plays an important role in data mining. The main idea of our algorithm is to combine the method of random orthogonal matrix transformation with the proposed secure multi-party protocol of matrix product to achieve zero lost of accuracy in most data mining implementations.

This paper is organized as follows. In Sect. 2, we give a brief description of multi-party distributed data mining model. In Sect. 3, we propose a new secure multi-party protocol of matrix product, and prove that the protocol can maintain the product of matrices after the data distortion. In Sect. 4, we discuss how to compute all orthogonal matrix transformations for different parties in the same coordinate system, and present a new hybrid algorithm for privacy preserving distributed data mining. Finally, Sect. 5 draws our conclusions.

2. Privacy Sensitive Distributed Data Mining Model

By analyzing the past data, data mining tools usually build models which can predict outcomes of given situations. For example, in an e-mail program, data mining firstly builds a classification model based on the past email data, and then attempts to classify a new e-mail as "legitimate" or as "spam". Therefore, the task of data mining is to build a model which can have a high accuracy on the prediction problem. Then the legitimate email will be classified as "legitimate" with a high probability, and so the spam emails will be. The accuracy of the model can then be measured from how many e-mails it correctly classify. A number of statistical methods may be used to evaluate the accuracy on this kind of data mining models, such as precision, recall, and *ROC* curves.

Distributed Data Mining explores techniques of how to conduct the data mining on the data distributed over several databases. We focus on the distributed data mining model for *horizontally partitioned data* since it is more challenging than the traditional data mining in a centralized way. In this model, the different organizations (or participants) own the same set of attributes for different sets of entities. Dis-

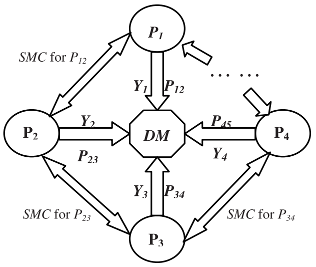

Fig. 1 The architecture of the distributed data mining model.

tributed Data Mining conducts the data mining on the union of these different sets of entities. In order to improve the accuracy of data mining, the data owners need to put their data together and build a shared data mining model. However, considering the personal privacy, participants do not trust each other, or even the data miner who developed the data mining program. Hence, they are reluctant to share their data with others.

The overall architecture of our distributed data mining method is depicted in Fig. 1. In our model, we assume there are k participants (data owner) and one data miner. All the participants sit counter-clockwise along a circle and are distinctively numbered as $P_1, P_2, \ldots P_k$. The data miner seats in the middle of the circle, and is labeled as DM. Each participant P_i owns its original data matrix X_i with n_i rows and m columns, where each row represents a different record and each column represents an attribute.

Suppose that the original dataset is represented by D, and the disturbed dataset is represented by D'. Similar to some previous works [18], [19], the privacy level of the i-th attribute is defined as the variance of $(D^i)' - D^i$, where $(D^i)'$ and D^i are the i-th column of D' and D respectively. Suppose the disturbed dataset is transformed from the the original dataset by a transformation matrix M, i.e. $D' = D \times M$. Then the privacy level can be computed by Formula 1, where E is an identity matrix, and E^i and M^i are the i-th column of E and M respectively.

$$\begin{aligned} & Var((D^i)' - D^i) \\ & = Var(D \times (E^i - M^i)) \\ & = (E^i - M^i)^T \times Cov(D) \times (E^i - M^i) \end{aligned} \tag{1}$$

Different participants may have different privacy concerns. Each participant P_i randomly generates a m by m transformation matrix R_i, which represents different privacy levels for different attributes as illustrated in Formula 1. With privacy concerns, each participant P_i makes a *transformation* of the original data matrix X_i by $Y_i = X_i \times R_i$, and sends the perturbed data matrix Y_i to DM instead of the original data matrix X_i. DM will use the perturbed data to establish a shared data mining model.

Distance metric gives a numerical value that measures the similarity between two data objects, and plays an important role in most data mining algorithms like decision trees, artificial neural network, clustering and classification. E.g., in classification, the class of a new data object having unknown class label is tagged as the class of its similar objects; in clustering, the similar objects are grouped together. The problem of computing most of these metrics between two records (vectors) a and b can be represented as the problem of computing the inner products between them. E.g., the Euclidean distance can be reduced to the inner product computation using the following expression, $d^2(a, b) = a \cdot a + b \cdot b - 2a \cdot b$, where $a \cdot b$ represents the inner product of a and b. For the other metrics, the core problem of computation can be reduced to the similar problem of computing the Manhattan distance between them, i.e. $d_1(a, b) = \sum_i |a_i - b_i|$.

To achieve zero loss of accuracy for a number of data mining implementations, we need to maintain the distance metrics between any two records before and after the *transformation*. Specifically, to maintain the inner product of any two records in the latter data mining process, the transformation matrix R_i should be orthogonal and unit.

Usually in geometry, a *coordinate system* is a system which uses one or more numbers or coordinates, to uniquely determine the position of a point on a manifold such as Euclidean space. In the m-dimensional Euclidean space, each record (row vector) in X_i can be viewed as a data point in the original coordinate system. The value in the j-th attribute (column) of this record can be seen as the j-th coordinate of this point in the original coordinate system.

A *coordinate transformation* converts points from one coordinate system to another. By using different orthogonal transformation matrix R_i, each participant transforms the original coordinate system into a different coordinate system. Therefore, DM cannot directly compare row vectors from different participants after the different coordinate transformations. Specifically, the inner product of the perturbed rows is $Y_i \times Y_j^T = X_i \times R_i \times R_j^T \times X_j^T$, which may not be equal to $X_i \times X_j^T$ when $i \neq j$.

Therefore, every two adjacent participants P_i and $P_{i+1}(P_{k+1} = P_1)$ are required to collaborate with each other to compute $P_{i,i+1} = R_i^T \times R_{i+1}$, and send $P_{i,i+1}$ to DM. In Sect. 4, we will describe how to use $P_{i,i+1}$'s to build a target coordinate system and integrate all the transformations into the same coordinate system. Since the leaking of the orthogonal transformation matrix R_i would allow DM to identify the original data matrix X_i, each $P_{i,i+1}$ will be computed by a new secure multi-party computation (*SMC*) protocol, which will be presented in Sect. 3.

3. New Secure Multi-Party Computation Protocols

In our model, every two adjacent participants P_i and P_{i+1} are required to collaborate with each other to compute the product of their random orthogonal transformation matrices, i.e., $P_{i,i+1} = R_i^T \times R_{i+1}$, and send $P_{i,i+1}$ to DM. Because of the privacy, we choose to use methods from *SMC* domain for the computation of the product matrix $P_{i,i+1}$.

The basic unit of computing the matrix product is to compute the inner product of two vectors. To overcome some limitations of available alternatives in the literature, we propose a new secure multi-party computation protocol of the inner product between two vectors. Secure computation (preserving privacy) of inner products is fundamental for many privacy preserving distributed data mining tasks. As opposed to available alternatives in the literature, our new protocol is more efficient and practical. It can be described as Protocol 1.

Protocol 1 (New Inner Product Protocol): Suppose two participants Alice and Bob have the original vector r_a and r_b of size m, respectively.

1. The trusted third-party randomly generates two orthogonal vectors ε_a and ε_b of size m such that $\varepsilon_a \cdot \varepsilon_b = 0$, and sends ε_a to Alice and ε_b to Bob.
2. Alice selects a random number k_a, calculates $r_a^{'} = r_a + k_a\varepsilon_a$, and sends $r_a^{'}$ to Bob; similarly, Bob selects a random number k_b, calculates $r_b^{'} = r_b + k_b\varepsilon_b$, and sends $r_b^{'}$ to Alice.
3. Alice calculates $s_1 = r_a \cdot r_b^{'}$ and $s_2 = k_a\varepsilon_a \cdot r_b^{'}$, and sends the result of $s_1 - s_2$ to the third-party; similarly, Bob does his calculation and sends the result of $s_3 - s_4 = r_b \cdot r_a^{'} - k_b\varepsilon_b \cdot r_a^{'}$ to the third-party.
4. The third-party computes $0.5(s_1 - s_2 + s_3 - s_4)$ and gets the inner product of two vectors r_a and r_b without knowing any information about them.

Correctness. In Protocol 1, the third-party finally received $s_1 - s_2$ from Alice and $s_3 - s_4$ from Bob, and he obtains the inner product of vectors r_a and r_b by computing the $(s_1 - s_2 + s_3 - s_4)/2$. The correctness of Protocol 1 can be illustrated by the Formula 2.

$$\begin{aligned} & s_1 - s_2 + s_3 - s_4 \\ & = r_a \cdot (r_b + k_b\varepsilon_b) - k_a\varepsilon_a \cdot (r_b + k_b\varepsilon_b) \\ & \quad + r_b \cdot (r_a + k_a\varepsilon_a) - k_b\varepsilon_b \cdot (r_a + k_a\varepsilon_a) \\ & = r_a \cdot r_b + k_b r_a \cdot \varepsilon_b - k_a\varepsilon_a \cdot r_b \\ & \quad + r_b \cdot r_a + k_a r_b \cdot \varepsilon_a - k_b\varepsilon_b \cdot r_a \\ & = 2r_a \cdot r_b \end{aligned} \tag{2}$$

Security analysis. In the second step of Protocol 1, Alice does not learn r_b because of the randomness from the vector ε_b. The random numbers of ε_b are generated from the real domain. By the random parameter k_b, Bob can adjust the domain of ε_b to the domain of r_b, indeed strengthen the privacy protection. Besides, Bob can keep changing the value of k_b to have more privacy if he has more than one vector need to protect. In the third step, the third-party only received two numbers from Alice and Bob. Both numbers cannot help him to find out r_a and r_b. Finally in the last step, the third-party calculates out the inner product of r_a and r_b. We suppose that the inner product is the public information, which can be known by the third-party.

Improvement on the security. The semi-honest condition assures that the third-party should not collude with either Alice or Bob. Even Alice colludes with the third-party, she cannot figure out the true values of r_b, as she does not know about the value of k_b. In order to improve the security of this protocol, the third-party can randomly generate four vectors with the condition that all the vectors are orthogonal to each other. The third-party sends two of them to Alice and the others two to Bob. Then Alice and Bob can randomly choose one from the received two vectors to compute r_b'. Therefore, even Alice colludes with the third-party, it's hard for her to know the values of r_b as the randomness on the randomly choosing from two random vectors.

Computation and communication evaluation. Suppose that Alice has n vectors and Bob has p vectors, and the dimension of each vector is m. In Step 1, the third party sends two random vectors wit $2m$ messages to Alice and Bob. In Step 2, Alice and Bob send one vector with m messages separately, and the total number of messages for computing the $p \times n$ inner products is $2m \times p \times n$. In Step 3, two messages are send by Alice and Bob, and the total number of messages is $2p \times n$. Therefore, the average the average communication cost of Protocol 1 is $(2m/(p \times n) + 2m + 2) \times b_0$. Here b_0 is the bit length of a message.

The computational complexity of Protocol 1 is $O(m)$ in total. Here, m is the dimension number of vectors. The computational cost contains the cost on $6m + 3$ additions, $6m + 3$ multiplications, and $2m + 2$ random number generations. It is analyzed as follows. (1) In Step 1, the third-party generates two orthogonal vectors having $2m$ numbers. (2) In Step 2, Both Alice and Bob generate one random number, and performs m additions and m multiplications. (3) In Step 3, Alice performs m additions and m multiplications to compute s_1, and m additions and $m + 1$ multiplications to compute s_2. Similarly, Bob does the same number of operations to compute s_3 an s_4. Beside, both of them perform one addition to compute $s_1 - s_2$ or $s_3 - s_4$. (4) In Step 4, The third party performs one addition and one multiplication to obtain the inner product.

Suppose Alice has the matrix $A_{n \times m}$ and Bob has the matrix $B_{p \times m}$. Alice and Bob expect that the third party can calculate the product matrix $P_{n \times p} = A \times B^T$ without knowing their original matrices. However, neither Alice nor Bob is allowed to get the product matrix; otherwise, it is possible that one of them breaks down the other's matrix. Naively, each element in the matrix P can be securely computed by Protocol 1 with the input vectors as the row vector of A and the column vector of B. To improve the efficiency of the matrix product computation, we design a new secure multiparty computation protocol for the matrix product, and describe it in Protocol 2.

Protocol 2 (New Matrix Product Protocol): Suppose that two participants Alice and Bob have the original matrix $A_{n \times m}$ and $B_{p \times m}$ respectively.

1. The trusted third-party randomly generates two orthogonal vectors ε_a and ε_b of size m such that $\varepsilon_a \cdot \varepsilon_b = 0$, and sends ε_a to Alice and ε_b to Bob.
2. Alice generate a random vector r_a of size n, and compute a matrix R_a with $(R_a)_{i\cdot}$ as $(r_a)_i\varepsilon_a$, then sends $A' = A + R_a$ to Bob; similarly, Bob generate a random vector r_b of size p, and compute a matrix R_b with $(R_b)_{i\cdot}$ as $(r_b)_i\varepsilon_b$, then sends $B' = B + R_b$ to Alice.
3. Alice calculates the matrix $S_1 = A \times (B')^T$ and $S_2 =$

$R_a \times (B')^T$, and sends the matrix $S_1 - S_2$ to the third party; Bob calculates the matrix $S_3 = A' \times B^T$ and $S_4 = A' \times R_b^T$, and sends the matrix $S_3 - S_4$ to the third party.

4. The third-party computes $0.5(S_1 - S_2 + S_3 - S_4)$ and gets the matrix product $P = A \times B^T$ without knowing any information about the matrices.

In Protocol 2, $(R_a)_{i\cdot}$ is the i-th row of the matrix R_a and $(r_a)_i$ is the i-th element of the vector r_a. Similar to Protocol 1, the correctness of Protocol 2 can be proved by Formula 3.

$$\begin{aligned}
&S_1 - S_2 + S_3 - S_4 \\
&= A \times (B + R_b)^T - R_a \times (B + R_b)^T + \\
&\quad (A + R_a) \times B^T - (A + R_a) \times R_b^T \\
&= A \times B^T + A \times R_b^T - R_a \times B^T - R_a \times R_b^T + \\
&\quad A \times B^T + R_a \times B^T - A \times R_b^T - R_a \times R_b^T \\
&= 2A \times B^T
\end{aligned} \tag{3}$$

In Formula 3, $R_a \times R_b^T$ equals the zero matrix. The (i, j)-th element in $R_a \times R_b$ is equal to $(r_a)_i\varepsilon_a \cdot (r_b)_i\varepsilon_b$, indeed equals 0. Similar to the analysis on Protocol 1, Protocol 2 need to send $2m + (n + p) \times m + 2n \times p$ messages, which is much smaller than the naive solution with $n \times p \times (2m + 2) + 2m = 2m + 2(n \times p) \times m + 2n \times p$. The computational cost of Protocol 2 is similar to the naive solution, and the complexity is $O(npm)$.

4. New Hybrid Approach for Privacy-Preserving Distributed Data Mining

In our model, each participant sends out its perturbed data using different orthogonal transformation matrix R_i. The DM has to integrate the perturbed data from different participants into the same coordinate system before data mining. For efficiency and privacy concerns, we require every two adjacent participants P_i and $P_{i+1}(P_{k+1} = P_1)$ to collaborate with each other to compute $P_{i,i+1} = R_i^T \times R_{i+1}$, and send $P_{i,i+1}$ to DM. We designate R_k as a target coordinate system, and define a coordinate matrix T_i as Formula 4.

$$T_i = \begin{cases} P_{i,i+1} \times P_{i+1,i+2} \times \ldots \times P_{k-1,k} & 1 \le i < k \\ E & i = k \end{cases} \tag{4}$$

It can be calculated by DM since he has received all the $P_{i,i+1}$ for every two adjacent participants P_i and $P_{i+1}(P_{k+1} = P_1)$. Then, the DM can transform the perturbed data matrix Y_i into the target coordinate system by Formula 5.

$$\begin{aligned}
Y'_i &= Y_i \times T_i \\
&= X_i \times R_i \times (R_i^T \times R_{i+1}) \times \ldots \times (R_{k-1}^T \times R_k) \\
&= X_i \times R_k
\end{aligned} \tag{5}$$

Suppose that two arbitrary participants P_i and P_j ($i <= j$) have got two original data matrices X_i and X_j respectively. The two participants first perturb their original data using different orthogonal transformation matrices R_i and R_j, and then send the perturbed matrices Y_i and Y_j to DM. After the DM transforms the perturbed data into the target coordinate system by Formula 3, the product of two original matrices X_i and X_j can be calculated by DM as Formula 6.

$$\begin{aligned}
&(Y_i \times T_i) \times (Y_j \times T_j)^T \\
&= (X_i \times R_k) \times (R_k^T \times X_j^T) \\
&= X_i \times (R_k \times R_k^T) \times X_j^T \\
&= X_i \times X_j^T
\end{aligned} \tag{6}$$

This means that the inner product of any two original records (vectors) from any two participants P_i and P_j can be computed by DM without knowing any information about them. Let $X_{i,s}$ represent the s-th object(row) in X_i of the participant P_i, $X_{j,t}$ represents the t-th object(row) in X_j of the participant P_j, $Y'_{i,s}$ represents the s-th vector(row) in Y'_i that transformed from X_i, and $Y'_{j,t}$ represents the t-th vector(row) in Y'_j that transformed from X_j. From Formula 7, we can see that the similarity between the s-th object in X_i and the t-th object in X_j can be obtained by computing the inner product of s-th vector(row) in Y'_i and the t-th vector(row) in Y'_j.

$$\begin{aligned}
&Y'_{i,s} \times (Y'_{j,t})^T = X_{i,s} \times R_k \times R_k^T \times Y_{j,t} = X_{i,s} \times Y_{j,t} \\
&where\ \ 1 \le i, j \le k, 1 \le s \le n_i, 1 \le t \le n_j
\end{aligned} \tag{7}$$

Our final hybrid algorithm for the privacy preserving distributed data mining can be described in Algorithm 1.

Algorithm 1 The new hybrid algorithm on privacy preserving distributed data mining.

```
Input:
    Suppose that all participants are distinctively numbered as $P_1, P_2, \ldots, P_k$, and own original data matrices $X_1, X_2, \ldots, X_k$, respectively. Each matrix has $m$ columns, each of which represents an attribute.
Output:
    A shared data mining model built on $X_1, X_2, \ldots, X_k$.
1: Each participant Pi randomly generates an orthogonal and unit transformation matrix $R_i$, and sends the perturbed data matrix $Y_i = X_i \times R_i$ to $DM$.
2: Every two adjacent participants $P_i$ and $P_{i+1}(P_{k+1} = P_1)$ collaborate with each other to compute $P_{i,i+1} = R_i^T \times R_{i+1}$ using Protocol 2.
3: Upon receiving all $P_{i,i+1}$'s, $DM$ starts to compute $T_i$ for $1 \le i \le k$ using formula (2).
4: $DM$ transforms the perturbed data from each participant $P_i$ into the target coordinate system of $R_k$. by $Y_i^{'} = Y_i \times T_i$.
5: If the data mining algorithm uses dot product as the main component, then $DM$ calculates the product of every pair of $Y_i^{'}$ and $Y_j^{'}$, and obtains the inner product of any two original records from different participants; then $DM$ does the rest part of the data mining and builds the data mining model.
```

For the other data mining algorithms, which have been implemented using something like Manhattan distance as the main component, we have to require furthermore that the basis of the target coordinate system (i.e., the rows in the orthogonal transformation matrix R_k) should be a random permutation of the basis $(1, 0, ..., 0); (0, 1, 0, ..., 0); ...; (0, ..., 0, 1)$ in the original data space. Then the Manhattan distance between any two records before and after the transformation can be maintained obviously. However, the above described requirement is a little bit strong, and furthermore, the DM may be able to peek some private information if the ranges of some attributes are quite different from each other. If the proposed data mining algorithm use Manhattan distance as the main component, then DM calculates the Manhattan distance of every pair of row vectors in $Y_i^{'}$ and $Y_j^{'}$, and obtains the Manhattan distance of any two original records from different participants.

An application case in data mining. Classification is a classic tasks of data mining. It tries to identify the category of a new object on the basis of a training dataset containing objects whose category membership is already known. k-nearest neighbors algorithm (k-NN) [1] is a classic classification method, which predicts category memberships of new objects based on the k closest training objects in the feature space. For example, in the e-mail program, each email document is represented as a vector by the Vector Space Model, such as the term frequency-inverse document frequency (TFIDF) [1]. To predict the category ("legitimate" or "spam") of a new e-mail, k-NN firstly finds the k nearest training emails of this email based on their text vectors, and then assigns this email category as the category most common amongst its k nearest neighbors.

Under the distributed environment, we suppose there are k participants and the i-th participate has the data X_i. The j-th row $(X_i)_j$ of X_i represents the j-th object, and its category is represented by $(C_i)_j$. As the category of objects is public information, each participate directly sends the category vector C_i to DM. To get the training data, DM and all participants apply the Algorithm 1 without knowing the true privacy data. Finally, DM gets $Y_i' = X_i \times R_k$, where i is from 1 to k. Then DM gets $Y' = (Y_1'; Y_2'; \ldots, Y_k')$. Besides, we suppose that $X = (X_1; X_2; \ldots; X_k)$, $D = (X_1, C_1; X_2, C_2; \ldots, X_k, C_k)$ and $D' = (Y_1', C_1; Y_2', C_2; \ldots, Y_k', C_k)$.

If the i-th participate wants to predict the category of a new object x, he firstly preserves the original vector by $y = x \times R_i$, then sends y to DM. After receive y from the i-th participate, DM transforms y to $y' = y \times T_i$, where T_i is defined in Formula 4, then $y' = x \times R_k$. For the t-NN algorithm, t nearest neighbors of y' are selected from Y' based on the distance defined in Formula 8.

$$\begin{aligned} \|(Y')_i - y\|_2 &= (Y')_i \cdot (Y')_i + y' \cdot y' - 2(Y')_i \cdot y' \\ &= (X)_i \cdot (X)_i + x \cdot x - 2(X)_i \cdot x \\ &= \|(X)_i - x\|_2 \end{aligned} \tag{8}$$

Where $(Y')_i$ and $(X)_i$ are the i-th rows of Y' and X respectively. From Formula 8, we can conclude that the t nearest neighbors found from transformed space Y' are same with the t nearest neighbors found from original space X. At the last step, the t-NN algorithm assigns the category to y' as the category most common amongst its t nearest neighbors in D', i.e. the category for x.

5. Performance Comparison and Experiment Results

In this section, we compare our proposed inner product protocol with the priori related protocols. The inner product protocols can be divided into two categories. The first one contains the protocols based on the *encryption techniques*, such as homomorphic encryption. Three priori protocols in this category are Goethals2004 [20], Amirbekyan2007 [21], and FenXu2010 [22]. Here we rename the priori inner product protocols based on the author's name. The computational complexity of the above three protocols is $O(n * H)$, where n is the dimension of private vectors and $O(H)$ is the computational complexity of an encryption by homomorphic cryptosystem. All the above three protocols applied the Paillier Cryptosystem.

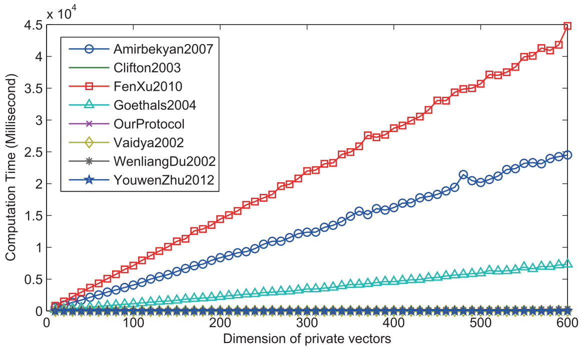

Fig. 2 Running time of all the inner product protocols.

The protocols in the second category usually designed based on the *algebra* such as the matrix operation, and four related protocols are Vaidya2002 [23], WenliangDu2002 [24], Clifton2003 [25] and YouwenZhu2012 [26]. The computational complexity of WenliangDu2002 is $O(n^3)$, in which the computation of matrix inverse costs most of the time. The complexity will be $O(n^2)$ if the computation of matrix inverse is not included. The computational complexity of Vaidya2002 and Clifton2003 is $O(n^2)$, as both of them apply the matrix multiplication. The computational complexity of YouwenZhu2012 and our protocol is $O(n)$.

We compare our proposed inner product protocol with the above 7 protocols. We conduct the experiments in Java on a computer with Intel Core 2 Duo 3.30GHz CPU and 8.0G memory. The encryption based protocols are implemented with Paillier's Homomorphic Cryptosystem[†]. In the experiments, the number of bits of modulus is 1024, and the time of encrypting a plaintext as Integer is 10.20 milliseconds while decrypting a ciphertext is 20.10 milliseconds.

We randomly generate the private vectors with the given dimension, and get the running time of all the protocols to compute the inner product. Figure 2 shows their running time. Three protocols based on the encryption techniques (i.e. Goethals2004, Amirbekyan2007, and FenXu2010) cost much more time than the ones based on algebra. It can be seen that the running time of the three protocols is linear with the dimension of vectors. As the Goethals2004 only encrypts n plaintexts and decrypts one ciphertext, it's running time is less than the time of other two protocols. For the Amirbekyan2007, the applied Add

[†] Java Implementation can be downloaded at http://www.csee.umbc.edu/~kunliu1/research/Paillier.html

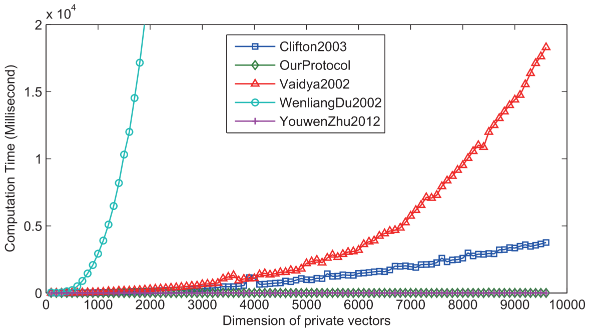

Fig. 3 Running time of protocols based on the algebra.

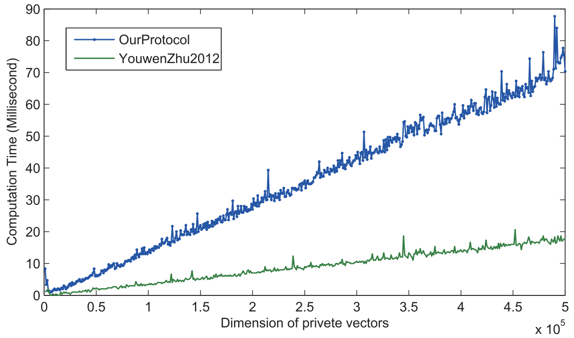

Fig. 4 Running time of the our protocol and YouwenZhu2012.

Vector Protocol needs to encrypts $2n$ plaintexts and decrypts n ciphertexts. FenXu2010 applies the Add Vector Protocol twice, and it needs to encrypts $3n$ plaintexts and decrypts $2n$ ciphertexts. In Figure 2, the running time of protocols based on algebra cannot be displayed clearly.

In order to compare the performance of protocols based on algebra, we set the dimension of vectors from 100 to 9600 with step length as 100 and show the results in Figure 3. It can be seen that the running time of WenliangDu2002 shows the tendency with complexity as $O(n^3)$, meanwhile Vaidya2002 and Clifton2003 with $O(n^2)$. Our proposed protocol is much more efficient than the above three protocols. To compare our protocol with YouwenZhu2012, we conduct the experiment on the vector with dimension from 100 to 500000, and Figure 4 shows the simulated results. Clearly, both of them have linear relationship with the vector dimension, and YouwenZhu2012 cost less time than our protocol. But YouwenZhu2012 only can be used for the even-dimension vectors. Besides for the vectors $x = (x_1, x_2, \ldots, x_{2k})$ of Alice and $y = (y_1, y_2, \ldots, y_{2k})$ of Bob, Alice can learn $y_{2i} - y_{2i-1}$ and Bob can figure out $x_{2j-1} + x_{2j}$. To some extent, this leads to the privacy disclosure of the private vectors.

From the experimental results we can conclude that our proposed inner product protocol is much more efficient than the others protocols. Besides, our protocol can provide more privacy protection than the efficient YouwenZhu2012. Therefore, we method is suitable to securely compute the inner product in large scale systems.

6. Conclusions and Future Work

In this paper, we present a new hybrid approach for privacy preserving distributed data mining. The main idea of the hybrid approach is to use the orthogonal transformation to maintain the inner product between records, and a new secure multi-party protocol for the collaboration of required computation between participants. We prove that all perturbed data by the orthogonal matrix transformation from different participates can be integrated into the same coordinate system, and zero loss of accuracy in most data mining implementations can be achieved.

Most data mining algorithms employ inner product as their main component in the implementations. For those others that are implemented using something like Manhattan distance we have to require furthermore that the basis of the target coordinate system should be a random permutation of the basis in the original data space. However, this further requirement is a little bit strong. In the future work, we will explore some alternative solutions.

Acknowledgments

This work is supported by Natural Science Foundations of China (No.61003231, No.61103109 and No.11105024), and by the Fundamental Research Funds for the Central Universities under ZYGX2011J057. The authors are grateful for the anonymous reviewers who made constructive comments.

References

[1] J.W. Han and M. Kamber, Data Mining: Concepts and Techniques. 2nd edition, Morgan Kaufmann Publishers, San Francisco, 2006.
[2] J. Mena, Investigative Data Mining for Security and Criminal Detection, Butterworth-Heinemann, Edinburgh, 2003.
[3] L.F. Cranor, J. Reagle, and M.S. Ackerman, Beyond concern: Understanding net users' attitudes about online privacy, Report no.AT&T Labs-Research Technical Report TR 99.4.3. AT&T Labs-Research, 1999.
[4] C.C. Aggarwal and P.S. Yu, Privacy-Preserving Data Mining: Models and Algorithms, Springer, 2008.
[5] E. Bertino, I.N. Fovino, and L.P. Provenza, "A framework for evaluating privacy preserving data mining algorithms," Data Mining and Knowledge Discovery, vol.11, no.2, pp.121–154, 2005.
[6] R. Agrawal and R. Srikant, "Privacy-preserving data mining," ACM SIGMOD Record, vol.29, no.2, pp.439–450, 2000.
[7] K. Liu, H. Kargupta, and J. Ryan, "Random projection-based multiplicative data perturbation for privacy pre-serving distributed data mining," IEEE Trans. Knowl. Data Eng., vol.18, no.1, pp.92–106, 2006.
[8] S. Gomatam, A.F. Karr, and A.P. Sanil, "Data swapping as a decision problem," J. Official Statistics, vol.21, no.4, pp.635–655, 2005.
[9] Y. Saygin, V. Verykios, and C. Clifton, "Using unknowns to prevent discovery of association rules," ACM SIGMOD Record, vol.30, no.4, pp.45–54, 2001.
[10] L. Sweeney, "k-anonymity: A model for protecting privacy," Int. J. Uncertainty, Fuzziness and Knowledge-based Systems, vol.10, no.5, pp.557–570, 2002.
[11] A. Machanavajjhala, D. Kifer, and J. Gehrke, "Venkitasubramaniam

M. l-diversity:Privacy beyond k-anonymity," ACM Trans. Knowledge Discovery from Data, vol.1, no.1, pp.1–47, 2007.

[12] N. Li and T. Li, "t-closeness: Privacy beyond k-anonymity and l-diversity," Proc. 23rd International Conference on Data Engineering (ICDE), pp.106–115, April 2007.

[13] S.R.M. Oliveira and O.R. Zaïane, "A privacy-preserving clustering approach toward secure and effective data analysis for business collaboration," Computers & Security, vol.26, no.1, pp.81–93, 2007.

[14] S.R.M. Oliveira and O.R. Zaïane, "Privacy-preserving clustering by object similarity-based representation and dimensionality reduction transformation," Workshop on Privacy and Security Aspects of Data Mining (PSDM'04). pp.21–30, 2004.

[15] K. Liu, H. Kargupta, and J. Ryan, "Random projection-based multiplicative data perturbation for privacy preserving distributed data mining," IEEE Trans. Knowl. Data Eng., vol.18, no.1, pp.92–106, 2006.

[16] S.R.M. Oliveira and O.R. Zaïane, "Privacy preserving clustering by data transformation," Proc. Brazilian Symposium on Databases, pp.304–318, 2003.

[17] S.R.M. Oliveira and O.R. Zaïane, "Achieving privacy preservation when sharing data for clustering," Proc. International Workshop on Secure Data Management in a Connected World, pp.67–82, Toronto, Canada, 2004.

[18] K. Chen and L. Liu, "Privacy preserving data classification with rotation perturbation," Proc. IEEE ICDM, pp.589–592, 2005.

[19] W.J. Yang and S.T. Huang, "Data privacy protection in multi-party clustering," Data and Knowledge Engineering, vol.67, no.1, pp.185–199, 2007.

[20] B. Goethals, S. Laur, H. Lipmaa, and T. Mielikainen, "On private scalar product computation for privacy-preserving data mining," Proc. 7th Int. Conf. Information Security and Cryptology, pp.104–120, Dec. 2004.

[21] A. Amirbekyan and V. Estivill-Castro, "A new effcient privacy-preserving scalar product protocol," Proc. 6th Aus-tralasian Data Mining Conference, pp.209–214, Dec. 2007.

[22] F. Xu, S. Zeng, and et al. "Research on secure scalar product protocol and its application," Proc. 6th International Conference on Wireless Communications Networking and Mobile Computing, pp.1–4, 2010.

[23] J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partioned data," Proc. 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.639–634, 2002.

[24] W. Du and Z. Zhan, "A practical approach to solve secure multiparty computation problems," 2002 Workshop on New Security Paradigms, pp.127–135, 2002.

[25] C. Clifton, M. Kantarcioglu, X. Lin, et al., "Tools for privacy preserving distributed data mining," SIGKDD Explorations, vol.4, no.2, pp.28–34, 2003.

[26] Y. Zhu, T. Takagi, and L. Huang, "Efficient secure primitive for privacy preserving distributed computations," Proc. 7th International Workshop on Security, LNCS 7631, pp.233–243, 2012, Fukuoka, Japan.

Chongjing Sun received the B.S. degree in Mathematics and Information Science from Yantai University in 2008. He is currently working as a Ph.D. student in the School of Computer Science and Engineering, University of Electronic Science and Technology of China. He received the CSC scholarship in 2011 and visited University of Illinois in 2012 for one year. His main research interests include data mining, privacy preserving, and complex network.

Hui Gao received the PhD degree in computing science from the University of Groningen (the Netherlands) in 2005. He is now a professor and Ph.D. supervisor in the School of Computer Science and Engineering, University of Electronic Science and Technology of China. He has published more than 30 papers in international conferences and journals. His research interest includes data mining, privacy preserving and parallel programming.

Junlin Zhou received the Ph.D. degree in Computer Science from University of Electronic Science and Technology of China in 2010. He is now an associate professor in University of Electronic Science and Technology of China. He received the CSC scholarship in 2007 and visited University of Minnesota in 2008 for one year. His main research interest includes data mining and recommender system.

Yan Fu received the M.E. in Computer Science from University of Electronic Science and Technology of China in 1988. She is now a professor and Ph.D. supervisor in University of Electronic Science and Technology of China. She has published more than 50 research papers in international conferences and journals. Her research interest includes data mining and intelligence computing.

Li She received the Ph.D degree in Computer Science from Chengdu Institute of Computer Applications, Chinese Academy of Sciences in 2006. She is now a professor of Alibaba Business College,Hangzhou Normal University. Her main research interest includes big data mining and consumer behavior of E-commerce.