International Journal of Security and Its Applications
Vol. 1, No. 1, July, 2007

Protocols for Privacy-Preserving DBSCAN Clustering

XU Wei-jiang, HUANG Liu-sheng, LUO Yong-long, YAO Yi-fei and JING Wei-wei
Department of Computer Science and Technology, University of Science and Technology of China, Hefei, China, 230026
Suzhou Institute for Advanced Study, University of Science and Technology of China, Suzhou, China, 215123
wjxu@mail.ustc.edu.cn, lshuang@ustc.edu.cn

Abstract

Cooperative computation is one of the most important fields in computer science. In recent years, the development of networking increases the desirability of cooperative computation. But privacy concerns often prevent different parties from sharing their data. Secure multi-party computation techniques can dispel parties' doubts about revealing privacy information in this situation. On the other hand, Data mining has been a popular research area for more than a decade. However, in many applications, the data are originally collected at different sites owned by different users. This paper considers the problem of privacy preserving DBSCAN clustering over vertically partitioned data based on some results of SMC. An efficient secure intersection protocol is first proposed. The security and complexity of the protocols are also analyzed. The results show that the protocols preserve the privacy of the data and the time complexity as well as the communication complexity is acceptable.

1. Introduction

Cooperative computation, in which people can jointly take part without the limit of their physical positions, is one of the most important fields in computer science. But the desirability of cooperative computation could occur between partially trusted parties, or even between competitors, privacy concerns may prevent different parties from sharing their data in order to do computation tasks.

Secure multi-party computation techniques aim at the problem. Generally speaking, secure multi-party computation (SMC) deals with the privacy concerns in distributed environment while ensuring correctness of the computation and that no more information is revealed to a participant in the computation than can be inferred from that participant's input and output [12]. SMC problems exist in many computation domains. The theoretic general solution can apply to every SMC problem; however, due to its high complexity problem-specific solutions should be developed [9].

On the other hand, data mining has been a popular research area for more than a decade because of its ability of efficiently extracting statistics and trends from large sets of data. Clustering is considered as one of the most important problems in data mining. The objective of clustering is to partition objects into clusters such that similar objects are in the same group while different objects are in different groups.

At present, most data mining algorithms can only deal with single data source. But in many applications, the data are originally collected at different sites, owned by different users. In order to extract information out of these data, they are brought together first and then clustered. Naturally this raises concerns about the privacy of individuals. For instance, if we want to set up a questionnaire to identify trends and patterns of the

diseases in a city, many citizens may refuse to provide their diseases information or even offer fake data, and the precision of the results will be impacted.

Privacy-Preserving Data Mining (PPDM) which considers how to solve data mining in a cooperative environment while preserving each party's privacy information is a class of specific SMC problems. In the year 2000, two different privacy preserving data mining problems were presented in [1] and [2] and solved using *data perturbation* and *Secure Multi-Party Computation* (*SMC*) protocols. Subsequently many privacy preserving techniques for different data mining models were proposed: In [3], Vaidya and Clifton implemented privacy preserving K-means clustering using *SMC* and *Homomorphic Encryption*. Pinkas proposed cryptographic techniques for privacy preserving classification in [4]. Techniques for privacy preserving association rule mining in distributed environments were proposed by Vaidya and Clifton [5]. Luo Yong-long, Huang Liu-sheng et al. studied privacy preserving Boolean association mining in [6].

In PPDM the way the data are distributed plays an important role in defining the problem. Data could be partitioned into many parts either *vertically* or *horizontally* [7]. In this paper, we apply privacy preserving ideas based on SMC basic protocols to DBSCAN [11] algorithm to mine clusters from vertically partitioned data while protecting privacy. We first outline related work. A secure protocol for computing intersection set is given in section 3. In section 4, we formally define the problem and give the overall solution. Section 5 analyzes the security and complexity of the protocols. We conclude by summarizing the contributions of this paper.

2. Related work

2.1. Secure channel assumption and adversarial behaviors

SMC assumes that the communication channels between participant sites are secure. That is, each pair of parties is connected by a reliable and private (or secret) channel.

SMC research typically considers two types of adversarial behaviors: semi-honest adversaries and malicious adversaries. Loosely speaking, a semi-honest adversary is the one who follows the protocol specification properly with the exception that it keeps a record of all its intermediate computations and attempts to learn additional information by analyzing the messages received during the protocol execution. In the contrary, a malicious adversary may arbitrarily deviate from the protocol specification, such as substituting its local input or aborting the protocol prematurely.

A secure protocol for semi-honest adversaries can be transformed into a protocol that is secure against malicious adversaries using zero knowledge proof. In this paper, we assume that all participant sites are semi-honest (which is usually the truth in reality). For details about adversarial behavior, see [12].

2.2. Secure sum protocol

Secure sum problem can be described as follows: There are $r(r > 2)$ parties, who are numbered $1,2,3,\ldots,r$, and the private input of party i is v_i. All parties want to compute and obtain $\sum_{i=1}^{r} v_i$, while the private input of each party can never be obtained by any other party. The sum is sometimes shared by two parties in order to do further

computation when secure sum is a sub-protocol in an algorithm. An implementation of secure sum protocol can be found in [8].

2.3. Millionaires' protocol

Consider two parties having their own privacy numerical values. The problem of secure comparison protocol is to securely compare the two values without leaking any additional information about the private value to the other party. In [9], A. C. Yao firstly presented this problem and described it as *millionaires' problem*. Yao's solution needs exponential time, but several constant-round solutions have been proposed in resent years [10].

2.4 Commutative encryption

An encryption algorithm is commutative if the following two equations hold for any given feasible encryption keys $K_1, K_2, \ldots, K_n$, any m in plain text domain M, and any permutations of i, j, $E_{k_{i1}}(\ldots E_{k_{in}}(m)\cdots) = E_{k_{j1}}(\ldots E_{k_{jn}}(m)\cdots)$, and $\forall M_1, M_2 \in M, M_1 \neq M_2$ for given $k, \varepsilon < 1/2^k$, $\text{Prob}(E_{k_{i1}}(\ldots E_{k_{in}}(M_1)\cdots) = E_{k_{j1}}(\ldots E_{k_{jn}}(M_2)\cdots)) < \varepsilon$

There are several examples of commutative encryption, such as RSA, Pohlig-Hellman. A detailed discussion can be found in [13].

3. Secure intersection protocol

3.1. Protocol

The protocol requires two sites as leader sites. Without loss of generality, let Site 1 and 2 be leader sites. Here we use RSA cryptography as commutative encryption scheme, and assume the cardinalities of all data sets needn't be protected.

Protocol 1: *Secure Intersection Protocol*

Input: There are r participant sites, which are numbered $1, 2, 3, \ldots, r$. Each site has a data set as its privacy input, which are denoted S_i for Site i.

Output: All sites get $S = \bigcap_{i=1}^{r} S_i$, but no site gets any additional information about other sites' data sets except their cardinalities.

Step 1: Site 1 randomly generates a pair of big primes p and q as the prime-parameters of RSA, and randomly chooses a public-private key pair (E_{k1}, D_{k1}). And then Site 1 sends p and q to Site 2.

Step 2: On receiving p and q from Site 1, Site 2 randomly chooses a public-private key pair (E_{k2}, D_{k2}) and sends the public key E_{k2} to Site i($3 \le i \le r$).

Step 3: Site i ($2 \le i \le r$) encrypts each object in the set S_i with public key E_{k2}, and sends the set of encrypted object (denoted $E_{k2}(S_i)$) to Site 1.

Step 4: After receiving all the sets of encrypted objects $E_{k2}(S_i)(2 \le i \le r)$ from Site i, Site 1 computes the encrypted set $E_{k2}(PS) = \bigcap_{i=2}^{r} E_{k2}(S_i)$.

Step 5: Site 1 encrypts each object in the sets $E_{k2}(PS)$ and S_1 with its own public key E_{k1}, and sends $E_{k1}(E_{k2}(PS))$ and $E_{k1}(S_1)$ to Site 2.

Step 6: After decrypting $E_{k1}(E_{k2}(PS))$ with his private key D_{k2}, Site 2 obtains $E_{k1}(PS)$ because of the property of *commutative encryption*. Then Site 2 sends the encrypted intersection $E_{k1}(S) = E_{k1}(S_1) \cap E_{k1}(PS)$ to Site 1.

Step 7: After receiving $E_{k1}(S)$ Site 1 decrypts it and get the genuine intersection S which is succedently broadcasted.

3.2. Analysis

Theorem 1: *(Correctness) At the end of Protocol 1, all participant sites get the intersection of all* S_i.

Proof: The RSA public-key cryptography scheme is deterministic algorithm. That is, there is one unique cipher text for one plain text. Thus the intersection of encrypted data sets equals to the encrypted intersection.

So $E_{k2}(PS) = \bigcap_{i=2}^{r} E_{k2}(S_i) = E_{k2}(\bigcap_{i=2}^{r} S_i)$ in Step 4 and $E_{k1}(S) = E_{k1}(S_1) \cap E_{k1}(PS)$ $= E_{k1}(\bigcap_{i=1}^{r} S_i)$ in Step 6. After decrypting $E_{k1}(S)$, we obtain the real intersection of all data sets, and the theorem follows. □

In the process of the Protocol 1, only Site 1 and Site 2 can get the objects that may be not contained in the intersection, but these objects are encrypted by a public key they don't know. Obviously Site 1 and Site 2 can't get the real data. We can prove the security of Protocol 1 using the concepts in [12].

Theorem 2: *Suppose the cardinalities of all data sets needn't be protected, then Protocol 1 securely computes the intersection set.*

Proof: An algorithm or protocol is said to be secure if the view of every party can be simulated by a probabilistic polynomial-time algorithm (named *simulator*) given access to the party's *input and output only*. That is, the output of the simulator is computationally indistinguishable from the real view of the party in the algorithm or protocol. A detailed discussion on security can be found in [12].

The *view* of Site 1: Site 1 receives all encrypted sets in Step 4 and receives $E_{k1}(S)$ in Step 7. The view of Site 1 is $View_1 = (S_1, (p, q, E_{k1}, D_{k1}), E_{k2}(S_2), \ldots, E_{k2}(S_r), E_{k1}(S))$ .On input (S_1, S), the simulator can be constructed as follows:

1) The simulator randomly generates a pair of big primes p' and q' as the prime-parameters of RSA, and randomly chooses two public-private key pairs (E'_{k1}, D'_{k1}) and (E'_{k2}, D'_{k2}).

2) The simulator uniformly generates r-1 sets $S'_2, S'_3, \ldots, S'_r$, satisfying $|S'_i| = |S_i|$ for $i = 2, 3, \cdots, r$ and $S_1 \cap (\bigcap_{i=2}^{r} S'_i) = S$ and $|\bigcap_{i=2}^{r} S'_i| = |PS|$. After encrypting each object in the sets $S'_2, S'_3, \ldots S'_r$, we get $E'_{k2}(S'_2), E'_{k2}(S'_3), \ldots, E'_{k2}(S'_r)$.

3) The simulator outputs $(S_1, (p', q', E'_{k1}, D'_{k1}), E'_{k2}(S'_2), \ldots, E'_{k2}(S'_r), E'_{k1}(S))$

Both (p, q) and (p', q') are randomly generated, so $(p, q) \overset{c}{\equiv} (p', q')$. Both (E'_{k1}, D'_{k1}) and (E_{k1}, D_{k1}) are chosen randomly, so they are computationally indistinguishable

provided $(p,q)=(p',q')$. So $(p,q,E_{k1},D_{k1}) \overset{c}{\equiv} (p',q',E'_{k1},D'_{k1})$ due to the definition of conditional probability. According to the property of the encryption scheme, $E'_{k2}(S'_i)$ and $E_{k2}(S_i)$ are computationally indistinguishable. The computational indistinguishability of the simulator and the view of Site 1 follows.

The *view* of Site 2: Site 2 receives *p, q* in Step 2, and receives $E_{k1}(E_{k2}(PS))$ in Step 6. The view of Site 2 is $View_2 = (S_2,(E_{k2},D_{k2}),p,q,E_{k1}(E_{k2}(PS)),S)$.

On input (S_2,S), the simulator for Site 2 can be constructed as follows:

1) The simulator randomly generates a pair of big primes p' and q' as the prime-parameters of RSA, and randomly chooses two public-private key pairs (E'_{k1},D'_{k1}) and (E'_{k2},D'_{k2}).

2) The simulator uniformly generates a set PS' satisfying $|PS'|=|PS|$ and $PS' \cap S_2 \supseteq S$.

3) The simulator outputs $(S_2,(E'_{k2},D'_{k2}),p',q',E'_{k1}(E'_{k2}(PS)),S)$

Similar to the discussion for Site 1, we can deduce that the simulator and the view of Site 2 are computationally indistinguishable.

The view of Site *i*, $i=3,4,\cdots,r$: Site *i* receives E_{k2} in Step 3 and *S* in Step 7 The view of Site *i* is $View_i = (S_i,\lambda,E_{k2},S)$ where λ represents Site *i* has no randomness. The simulator for Site *i* is obvious. On input (S_i,S), the simulator randomly generates a pair of big primes p' and q' as the prime-parameters of RSA, and randomly chooses a public-private key pairs (E'_{k2},D'_{k2}) and outputs (S_i,λ,E'_{k2},S).

Thus, it can be concluded that if the cardinalities of all data sets are *not* private information, we can simulate the views of all the parties on the real input and output only. Theorem 2 follows. □

Complexity: Let *s* be the size of input sets, and let *r* be the number of participant sites. The size of the intersection is $O(s)$. The time complexity of sequential intersection algorithm can generally be considered as $O(n\log n)$ if the set is unsorted. The time complexity for each step is $O(1)$, $O(r)$, $O(s)$, $O(rs\log rs)$, $O(s)$, $O(s\log s)$, $O(rs)$ respectively, so the total time complexity of Protocol 1 is $O(rs\log rs)$ which is the same with sequential intersection algorithm. The communication complexity for each step is $O(1),O(r),O(rs),0,O(s),O(s),O(rs)$ respectively, so the total complexity is $O(rs)$.

4. Privacy preserving DBSCAN algorithm

4.1 Basic concepts

DBSCAN algorithm is the first density-based clustering algorithm, which was proposed by Marting Ester et al. in 1996 [11]. *Eps*-neighborhood and core point are the most involved DBSCAN concepts in this paper. The *Eps*-neighborhood of a point *p* is defined by $N_{Eps}(p)=\{q \in Data\ set \mid \text{distance}(p,q) \le Eps\}$, and a point *p* is called a core point of a cluster if $|N_{Eps}(p)| \ge MinPts$. Here *Eps* and *MinPts* are the parameters of

DBSCAN, which mean the radius of the neighborhood and the minimum number of points in the *Eps*-neighborhood of a core point respectively.

To find a cluster, DBSCAN algorithm checks every point in the database. If a point p satisfies $|N_{Eps}(p)| \geq MinPts$, DBSCAN will create a cluster based on the point p (and the point p is a core point of the new cluster). DBSCAN algorithm retrieves all points density-reachable from a core point wrt. *Eps* and *MinPts* until no new points are added to the cluster. The basic version of DBSCAN can be found in the literature [11]. *regionQuery* is its core procedure which is used to get the set $N_{Eps}(p)$ for any point p, and is the only procedure that visits the real data. So the core problem to design privacy preserving DBSCAN algorithm is to design privacy preserving *regionQuery* procedure.

4.2 Problem formulation

There are $r(r \geq 2)$ sites, which are numbered $1,2,3,\ldots,r$, and n is the count of m-dimensional points in the distributed spatial database D from which the clusters are mined. Each site i has a portion of the database (denoted D_i) with attributes set V_i, satisfying $m = \sum_{i=1}^{r} |V_i|$ and $V_i \bigcap V_j = \phi(\mathrm{i} \neq \mathrm{j})$. There is a join key presented in all $D_i(1 \leq i \leq r)$ called *PID*. The problem is to mine clusters in the database D and meanwhile not to disclose any additional information. Here, we assume all participant sites are semi-honest.

In the environment of multi-party computation, several additional concepts are needed to describe DBSCAN to mine clusters from vertically partitioned data set.

Definition 1 (*projection/portion*): The *projection* (or *portion*) of a point p at Site i (denoted p_i) is a point that has attributes set V_i satisfying the values of these attributes are the same with point p.

Definition 2 (*local distance*): The *local distance* between two points p and q at Site i is defined as $\mathrm{distance}_i(p,q) = \mathrm{distance}(p_i, q_i)$.

Definition 3 (*local Eps-neighborhood*): When Site i independently runs DBSCAN wrt. global *Eps* and *MinPts*, we call the corresponding *Eps*-neighborhood *local Eps-neighborhood*(denoted $N_{Eps}^{i}(p)$). The local *Eps*-neighborhood can be defined as

$$N_{Eps}^{i}(p) = \{q \in Data\ set \mid \mathrm{distance}_i(p,q) \leq Eps\}.$$

Definition 4 (*local core point*): when Site i independently runs DBSCAN algorithm wrt. global *Eps* and *MinPts*, we call the points satisfying $|N_{Eps}^{i}(p)| \geq MinPts$ *local core point* at Site i.

For clearness, we sometime call the global counterpart *global distance, global Eps-neighborhood* and *global core point* respectively.

One may present a solution like this: Each site simply runs DBSCAN on its own data, and then use *secure intersection protocol* to get the common clusters. That preserves complete privacy, but couldn't work because: 1) The method ignores the influence of the attributes across the sites. It requires the attributes in the different sites are *independent*, but it is not always the truth. 2) There is no simple method to negotiate *Eps* and *MinPts*. The sensitivity of DBSCAN algorithm to parameters indicates the impracticality of the simple solution, so we need a more subtle one.

Note that DBSCAN needs to visit the actual data only when it retrieves $N_{Eps}(p)$. If we can redesign the procedure *regionQuery* to get $N_{Eps}(p)$ for any point p in multi-party

environment as well as protect the real data, DBSCAN algorithm can mine clusters without reveal any additional information. The main idea to design privacy preserving *regionQuery* is to use SMC protocols: for any two points *p*, *q*, firstly all participant sites compute independently the local distance between *p* and *q*, and compute the global distance using the *secure sum protocol*. If the global distance is less than *Eps*, we can deduce that $q \in N_{Eps}(p)$ and $p \in N_{Eps}(q)$.

To promote the efficiency, we can firstly get the intersection of all local *Eps*-neighborhood of site *i* when we determine whether a point *p* is a core point. If the cardinality of the intersection is less than *MinPts*, obviously the point *p* is not a global core point, and we can leave out the following computations.

4.3 Secure two-party clustering

As stated earlier, the core procedure of DBSCAN is *regionQuery*, which is used to determine whether a point is a core point. If so, DBSCAN performs the *expand-cluster* operation and gets the corresponding cluster. In the classic centralized database environment, *regionQuery* can be implemented efficiently by *R**-trees [14]. In secure two-party computation, the implementation of *regionQuery* is based on millionaires' protocol and secure intersection protocol.

Protocol 2: *Secure Two-Party regionQuery*

Input: The two participant sites are Site *A* and Site *B*, and the parameter of *regionQuery* is the point *P*. For simplicity, we assume Site *A* to be the leader site.

Output: If *P* is a global core point, each site gets $N_{Eps}(P)$. Otherwise each site gets any point set whose cardinality is less than *MinPts*.

Step 1: Both sites run *regionQuery* locally on its own database, and get the *local Eps-neighborhood* of point *P*, denoted S_A and S_B respectively. If *P* is not local core point in both sites, both sites abort and output empty set.

Step 2: Site *A* and Site *B* invoke Protocol 1, and obtain the intersection $S = S_A \cap S_B$. If $|S| < MinPts$ holds, both sites abort and output empty set.

Step 3: Let *N* be a set initialized as empty set. For every point *Q* in *S*, suppose the local distances of *P* and *Q* in Site *A* and Site *B* are d_1 and d_2 respectively. Site *A* and Site *B* invoke millionaires' protocol with inputs d_1 and $Eps - d_2$ respectively. If the result is "$d_1 \le Eps - d_2$", $N \leftarrow N + \{Q\}$.

Step 4: Each site outputs the set *N*.

The "minus" in the expression "$Eps - d_2$" in Step 3 is not always the way it looks and its definition is related to the distance estimation methods. Suppose the portions of a point *P* in Site *A* and Site *B* are $(x_1, x_2, \cdots, x_t)$ and $(x_{t+1}, x_{t+2}, \cdots, x_n)$ $(t < n)$ respectively, and the portions of a point *Q* in site *A* and Site *B* are $(y_1, y_2, \cdots, y_t)$ and $(y_{t+1}, y_{t+2}, \cdots, y_n)$ $(t < n)$ respectively. If the Euclid distance is used, we can define d_1 as $(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_t - y_t)^2$ and define d_2 as $(x_{t+1} - y_{t+1})^2 + \cdots + (x_n - y_n)^2$ and define "$Eps - d_2$" as $Eps^2 - d_2$. The definitions can be done in a similar way when using other kinds of distance estimation methods.

4.4 Secure multi-party clustering

Protocol 3: *Secure Multi-Party regionQuery*

Input: There are $r(r > 2)$ sites, which are numbered $1, 2, 3, \ldots, r$. The parameter of *regionQuery* is the point *P*. We assume Site 1 and Site *r* to be the leader sites.

Output: If *P* is a global core point, every site gets $N_{Eps}(P)$. Otherwise every site gets any point set whose cardinality is less than *MinPts*.

Step 1: All sites run *regionQuery* locally on its own database, and get the local *Eps*-neighborhood of point *P*, denoted $S_1, S_2, \ldots, S_r$ respectively. If *P* is not local core point in all sites, all sites abort and output empty set.

Step 2: All sites invoke Protocol 1 with input S_1, $S_2, \ldots, S_r$ respectively, and obtain the output $S = \bigcap_{i=1}^{r} S_i$. If $|S| \le MinPts$, all sites abort and output empty set.

Step 3: Let *N* be a set initialized as empty set. For every point *Q* in *S*, suppose the local distances of *P* and *Q* at Site *i* is d_i.

Step 3.1: Site 1 generates a random number *R*, and sends $R + d_1$ to Site 2.

Step 3.2: For the sites $i = 2, \ldots r-1$, after receiving $v_i = R + \sum_{j=1}^{i-1} d_j$, Site *i* sends $v_i + d_i$ to Site $i+1$.

Step 3.3: Now Site *r* gets $v_r = R + \sum_{j=1}^{r-1} d_j$. Site 1 and Site *r* then invoke *millionaires' protocol* with input $R + Eps$ and $v_r + d_r$ respectively. If the result is "$v_r + d_r \le R + Eps$", $N \leftarrow N + \{Q\}$.

Step 4: Each site outputs the set *N*.

The rationality of adding distances in Step 3 can get from the explanation in section 4.3. It needs some transformation according to distance estimation methods which makes it addable while keeping the *invariability of comparison*.

5. Algorithm analysis

5.1 Correctness

Suppose a point *P* is the parameter of *regionQuery*. Protocol 2 and Protocol 3 are correct if we can prove that they output $N_{Eps}(P)$ when *P* is a core point, and output a set with cardinality less than *MinPts* when *P* is not a core point. Firstly two lemmas are presented.

Lemma 1: *If a point Q is in the global Eps-neighborhood of the point P, then Q is in the local Eps-neighborhood of the point P at all sites.*

Proof: Let *r* be the number of participant sites, and let *d* be the global distance of point *P* and *Q*, and let $d_i (1 \le i \le r)$ be the local distance of *P* and *Q* at Site *i*.

Because *Q* is in the global *Eps*-neighborhood of the point *P*, we can deduce $d < Eps$. And according to the definition of distance, $d \ge d_i (1 \le i \le r)$, so for $1 \le i \le r$, $d_i < Eps$ holds which means that *Q* is in the local *Eps*-neighborhood of point *P* at all sites. □

Applying Lemma 1, we can deduce easily that $N_{Eps}(P) \subseteq N^i_{Eps}(P)$ for $1 \le i \le r$.

Lemma 2: *If P is a global core point, then P is a local core point at all sites.*

Proof: Using Lemma 1, we know that the point *Q* is in the local *Eps*-neighborhood at all sites if it is in the global *Eps*-neighborhood of the point *P*. So if *P* is a global core point, then in the global *Eps*-neighborhood there are at least *MinPts* points which are all in the local *Eps*-neighborhood of point *P* at all sites. That means *P* is a local core point at all sites. □

Theorem 3: *(Correctness) Protocol 2 implements regionQuery correctly.*

Proof: Let a point *P* be a core point. Applying Lemma 2, *P* is a local core point at both Site *A* and Site *B*. Because $N_{Eps}(P) \subseteq N^i_{Eps}(P)$, we have $N_{Eps}(P) \subseteq \bigcap_{i=1}^{r} N^i_{Eps}(P)$. So if the tests in the Step 1 and Step 2 fail, *P* cannot be a core point. It is sound to output an empty set, because the cardinality of an empty set is 0 (which is always less than *MinPts*).

If the tests in Step 1 and Step 2 succeed, we then test carefully every point to determine whether it is in the *Eps*-neighborhood of *P*. In Step 4, *Q* is in the *Eps*-neighborhood of *P* if and only if the result is "$d_1 \le Eps - d_2$", therefore $N = N_{Eps}(P)$.

To sum up, if *P* is a core point, it will pass Step 1 and Step 2, and both sites output $N_{Eps}(P)$. Otherwise, both sites output an empty set (can't pass Step 1 or Step 2) or $N_{Eps}(P)$ (whose cardinality is less than *MinPts*). The theorem follows. □

Similar to the proof of Theorem 3, we can prove the correctness of Protocol 3, i.e.:

Theorem 4: *Protocol 3 implements regionQuery correctly.*

For conciseness, the proof of Theorem 4 is omitted.

5.2 Worst-case time analysis

In current popular implementation, the time complexity of millionaire' protocol can be regarded as constant (because we can regard the bit-number of its inputs as constant), and the time complexity of secure-sum-like procedure in Step 3 of Protocol 3 is the same with centralized sum, which is $O(r)$.

Suppose *R**-tree is used in Step 1 to retrieve $N^i_{Eps}(P)$, and let $t = \alpha \times MinPts$ and $u = \beta \times MinPts$ be the size of the sets involved in Step 2 and Step 3 respectively (α and β are constant, $1 \le \beta \le \alpha$). The time complexity for each step of Protocol 2 and Protocol 3 is shown in Table 1. *t* and *u* are much less than *n*, so the actual time complexity of Protocol 2 and Protocol 3 is close to the classic centralized *regionQuery* algorithm.

Table 1 Time complexity

	Protocol 2	*Protocol 3*
Step 1	$O(\log n)$	$O(\log n)$
Step 2	$O(t \log t)$	$O(rt \log rt)$
Step 3	$O(u)$	$O(ur)$
Step 4	$O(1)$	$O(1)$
Total	$O(\log n + t \log t + u)$	$O(\log n + rt \log rt + ur)$

5.3 Worst-case communication analysis

In current implementation, the communication complexity of millionaires' protocol can be regarded as constant and that of secure-sum-like procedure in Step 3 of Protocol 3 is $O(r)$. With the same assumption with Table 1, the communication complexity for each step of Protocol 2 and Protocol 3 is shown in Table 2. t and u are much less than n, so the actual communication complexity of Protocol 2 and Protocol 3 is low.

Table 2 Communication complexity

	Protocol 2	*Protocol 3*
Step 1	$O(1)$	$O(r)$
Step 2	$O(t)$	$O(rt)$
Step 3	$O(u)$	$O(ur)$
Step 4	$O(1)$	$O(1)$
Total	$O(MinPts \cdot (\alpha + \beta))$	$O(r \cdot MinPts \cdot (\alpha + \beta))$

5.4 Security

The use of millionaires' protocol and secure intersection protocol discloses nothing more than their results. But the combination of the intermediate results from these protocols leaks additional information more than the clusters information, i.e. Protocol 2 and Protocol 3 are not perfectly secure. However the leaked information does no harm to privacy data because all the disclosures of protocols are inequations. Thereinafter we will analyze the situation for Protocol 2. The analysis to Protocol 3 is similar.

Suppose the point Q is in the intersection S in Step 3 of Protocol 2. Site A can get that the local distance between P and Q at Site B is less than *Eps*. Suppose the portions of P and Q at Site B are $(x_1, x_2, \cdots, x_t)$ and $(y_1, y_2, \cdots, y_t)$ respectively, the leaking information of Site B can be expressed as:

$$(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_t - y_t)^2 < \varepsilon^2$$

It is an inequation with $2t$ unknowns. In the worse case, there are c points, denoted $P_1, P_2, \ldots, P_c$, and Site A gets the additional information after multiple- invoking the secure intersection protocol that these points are in the local *Eps*-neighborhood of each other at Site B. Site A can only deduce that the portions of the c points at Site B are close, which are in a t-dimensional sphere with diameter *Eps*. See Figure 1.

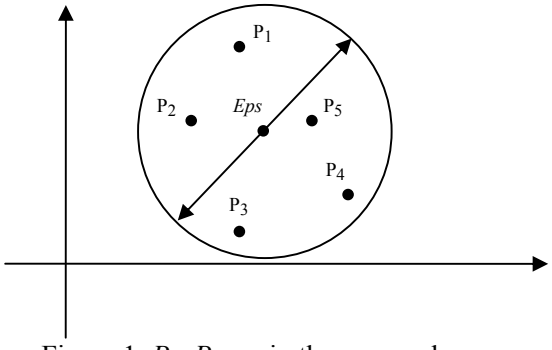


Figure 1. P_1~P_5 are in the same sphere

In Step 4, Site *A* can also get some inequations from the millionaires' protocol result. Similar to the above analysis, Site *A* cannot deduce any real data even from the combination of the results from millionaires' protocol and secure intersection protocol. That is, Protocol 2 is secure provided both sites are semi-honest.

6. Conclusion

Based on some results of SMC three protocols for privacy preserving DBSCAN clustering are presented in this paper when different sites contain different attributes for a common set of entities (vertically partitioned data). The algorithm doesn't alter the structure of the original DBSCAN algorithm. The only change of DBSCAN is its *regionQuery* procedure. In this paper, the time complexity, communication complexity, and security are analyzed elaborately. The result shows that the algorithm preserves the privacy while keeping low complexity.

For the veracity of the clustering, the global parameters *MinPts* and *Eps* are used in the local clustering at each site, which is not necessary. In practice, we can increase *MinPts* or decrease *Eps* or both in the local clustering which will reduce the running time. But how to deduce the local parameters is not an easy question and that is one of the problems we will do further research in the future.

7. Acknowledgement

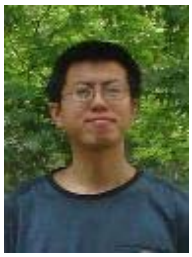
Supported by the National Natural Science Foundation (No. 60573171), the Science Foundation of Anhui Province (No. 070412043), the Ph. D. Program Foundation of Ministry of Education of China (No. 20060358014), the Science Foundation of Jiangsu Province (No. BK2007060).

8. References

[1] Rakesh Agrawal, Ramakrishnan Srikant, "Privacy-Preserving Data Mining", *In Proc. of the 2000 ACM IGMOD International conference on Management of Data*, Dallas, USA, 2000,439-450.
[2] Y Lindel, B Pinkas, "Privacy Preserving Data Mining", *In Advances in Cryptology-CRYPTO'00, volume 1880 of LNCS*. Springer-Verlag, 2000.36-54.
[3] Jaideep Vaidya, Chris Clifton, "Privacy-Preserving K-Means Clustering over Vertically Partitioned Data", *SIGKDD'03*, August 24-27, 2003, Washington, DC, USA.
[4] Benny Pinkas, "Cryptographic Techniques for Privacy-Preserving Data Mining", *SIGKDD Exploration*, 4(2), pp. 12.
[5] Jaideep Vaidya, Chris Clifton, "Privacy Preserving Association Rule Mining in Vertically Partitioned Data", *SIGKDD'02 Edmonton*, Alberta, Canada.
[6] Luo Yong-Long, Huang Liu-Sheng, et al., "An Algorithm for Privacy-Preserving Boolean Association Rule Mining", *ACTA ELECTRONICA SINICA*. 33(5),2005. pp.900-903.
[7] Jaideep Vaidya, "Privacy Preserving Data Mining Over Vertically Partitioned Data", Available: http://cimic.rutgers.edu/~jsvaidya/pub-papers/thesis.pdf .
[8] C. Clifton, et al., "Tools for Privacy Preserving Distri-buted Data Mining", *SIGKDD Exploration,* 4(2), pp. 28-34.
[9] A. C. Yao, "Protocols for secure computations", In proceedings of the 23rd Annual IEEE symposium on Foundations of Computer Science, 1982.
[10] Ioannidis, I. Grama, A., "An efficient protocol for Yao's millionaires'problem", *In Proc. of the 36th Annual Hawaii International Conference on System Sciences,* Jan. 2003.
[11] Ester M., Kriegel H. P., Sander J., Xu X., "A Density-Based Algorithm for discovering Clusters in Large Spatial Databases with Noise", *In Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining* (1996).
[12] O. Goldreich. "Secure multi-party computation (working draft)", Available:

http://www.wisdom.weizmann.ac.il/~oded/PS/prot.ps 1998.
[13] Murat Kantarcioglu, Chris Clifton, "Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data", *IEEE Transactions on Knowledge and Data Engineering*, 16(9), pp. 1026-1037, 2004.
[14] N. Beckmann, H. P. Kriegel, et al., "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", *In Proc. ACM SIGMOD Conf.*, pp. 322–331, Atlantic City, NJ, May 1990.

Authors



XU Wei-jiang

Born in 1981, Ph. D. candidate. His major research interests include distributed computing, data mining, and information security.

HUANG Liu-sheng

Born in 1957, professor, Ph. D. supervisor, His research interests include distributed computing and information security.

LUO Yong-long

Born in 1972, Ph. D., associate professor. His major research interests include distributed computing, computational geometry, and information security.

YAO Yi-fei
Born in 1981, Ph. D. candidate. Her research interests include information security and statistic analysis.

JING Wei-wei

Born in 1980, Ph. D. candidate. His research interests include information security.