Journal of Computer Science and Cybernetics, V.26, N.2 (2010), 87–101

ENHANCING PRIVACY IN DISTRIBUTED DATA CLUSTERING

LUONG THE DUNG[1], HO TU BAO[2]

[1]*Information Technology Center, Vietnam Government Information Security Commission*
[2]*Japan Advanced Institute of Science and Technology Nomishi, Ishikawa, Japan*

Abstract. The protocol of privacy-preserving clustering with distributed EM mixture modeling was proposed. However, it is not completely secure in the situation that something more than just the model parameters are revealed. Specially, when the dataset is horizontally partitioned into just two parts, this reveals extra information. The aim of this work is firstly to develop a more general protocol which allows the number of participating parties to be arbitrary and more secure. Secondly, we propose a better method for the case in which the dataset is horizontally partitioned into only two parts. This method allows computing covariance matrices and final results without revealing the private information and the clustering centers.

Tóm tắt. Một số giao thức đảm bảo tính riêng tư trong bài toán phân cụm dữ liệu dựa trên thuật toán EM đã được đề xuất trong cộng đồng nghiên cứu. Tuy nhiên, các giao thức này không hoàn toàn đảm bảo tính riêng tư, vì trong một số tình huống một vài tham số của mô hình có thể bị lộ, đặc biệt khi tập dữ liệu chỉ được phân thành hai phần theo chiều ngang. Bài báo này giới thiệu hai đóng góp chính trong một phương pháp mới giải bài toán trên. Một là giao thức mới cho phép một số lượng tùy ý các thành viên tham gia vào việc phân cụm dữ liệu và đảm bảo tốt hơn tính riêng tư cho dữ liệu của các thành viên. Hai là lời giải tốt hơn trong trường hợp tập dữ liệu chỉ được phân thành hai phần theo chiều ngang. Phương pháp này cho phép tính toán các ma trận hiệp phương sai và các kết quả cuối với việc đảm bảo không làm lộ các thông tin riêng tư cũng như đối tượng trung tâm của mỗi cụm dữ liệu.

1. INTRODUCTION

The growth of Internet has been creating a lot of opportunities for cooperative computation, in which cooperative data mining is an emerging area. It allows organizations to be able to cooperate with each others to obtain the mining result on their joint datasets. However, privacy concerns prevent organizations from revealing their private databases for various legal and commercial reasons. Therefore, the challenge is whether we can obtain the results of mining while still preserving the data secrecy. Privacy preserving data mining (PPDM) techniques have been proposed to address this kind of problem [18].

Clustering is one of the most popular techniques of data mining. The task is to group similar objects in a given data set into clusters with a goal of minimizing an objective function [7]. Expectation-maximization (EM) is an important clustering technique, which uses an iterative

method including two steps, expectation (E) step, which computes an expectation of the log likelihood by using the estimated parameters in M step, and maximization (M) step, which computes the parameters that maximize the expected log likelihood of data set [2], [14]. EM is widely used in many applications such as customer behaviour analysis, targeted marketing, and so on.

To our knowledge, there has been only one secure method for EM-based mixture model clustering from horizontally distributed sources so far [1], [12]. The basic idea of this method is that in each iteration, each party creates a local model from its data objects and computes global information from the previous iteration, then it securely merges its local model with the other's ones to generate the global model. This provides sufficient information to compute the global information needed for the next iteration. Once this process converges, each party can determine the clusters for its objects. However, this method is not completely secure in the situation that something more than just the model parameters are revealed. Specially, when the dataset is horizontally partitioned into just two parts, because the global model is a sum of local models, in case only two parties, which often happens in practice, each party could compute other party's local model by subtracting its local model from the global model.

This work is firstly to develop a privacy preserving EM-based mixture model clustering protocol for the multi-party partitioned data model. Unlike the existing protocol, our protocol allows the number of participating parties to be arbitrary, moreover it does not reveal numerators and denominators in calculating the parameters, therefore, the parties cannot learn extra information of the others. Secondly, we propose a better method for the case in which the dataset is horizontally partitioned into only two parts. This method requires protecting privacy of intermediate global information in particular the intermediate candidate cluster centers without loss of accuracy. To address this problem, we decompose the problem into subsecure computation works, such as the covariance matrices, means and the posterior probabilities computation.

The rest of this paper is organized as follows. In section 2, we briefly discuss related background, such as the EM algorithm and the security model. In section 3, we present the privacy preserving EM-based clustering protocol for the multi-party model. In section 4, we present the privacy preserving EM-based clustering protocol for the two-party model without disclosing cluster centers. In these sections, using the standard method of evaluating the protocols in PPDM studied in [3, 19], we provide an analysis of privacy and the estimation method of communication cost to prove (evaluate) the validity of our proposed methods. Finally, section 6 concludes our work.

2. BACKGROUND

A. EM-based mixture model clustering

In this section, we review the EM algorithm for Gaussian Mixture model. More details on the EM algorithm and mixture models can be found in [2] and [14].

Let D be a data set that has m objects $\{x_1, ..., x_m\}$ described by d attributes. Denote $x_j = (x_j[1],\ x_j[2],\ ...,\ x_j[d])$ the attribute vector of x_j. Assume that there exist k classes in the data set D, each follows some Gaussian distribution. The parameters of the class i are $\psi_i = \{\mu_i, \Sigma_i, \pi_i\}$, in which $\mu_i = (\mu_i[1], ..., \mu_i[d])$ is the center of the Gaussian distribution, Σ_i is the covariance matrix of the distribution and π_i is the probability of the class i. The normal density function of class i can be represented by

$$f(x;\psi_i) = \frac{|\Sigma_i^{-1}|^{1/2}}{(2\pi)^{d/2}} exp(-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)).$$

Thus, given the unknown parameters set $\psi = \{\psi_1,\ ..,\psi_k\}$, the normal mixture model is

$$f(x;\psi) = \sum_{i=1}^{k} \pi_i f(x;\psi_i).$$

The likelihood of the set data D is represented by

$$L(D;\psi) = \prod_{j=1}^{m} f(x_j;\psi).$$

The maximum likelihood principle means that the estimators that maximize the data likelihood are consistent estimators of the true parameters. The maximizing the likelihood of the set D is usually transformed to an equal maximization problem on the following variable, called log likelihood.

$$log(L(D;\psi)) = -\frac{1}{2}log(2\pi) - \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{m} z_{ij}(log|\Sigma_i| + \frac{1}{2}(x_j-\mu_i)^T\Sigma_i^{-1}(x_j-\mu_i)),$$

where z_{ij} is the posterior probability of x_j from class i, it can be calculated by

$$z_{ij} = \frac{f(x_j;\psi_i)\pi_i}{\sum_{l=1}^{k} f(x_j;\psi_l)\pi_l}. \quad (1)$$

The EM algorithm is to estimate the parameters set ψ. To estimate ψ, it starts with a randomly chosen initial parameter configuration ψ^0. Then, it keeps invoking iterations to recompute ψ^{t+1} based on ψ^t. Every iteration consists of two steps:

E-step: Compute the expected value of z_{ij}.

M-step: Update the parameters ψ^{t+1} using the following equations

$$\mu_i^{(t+1)} = \frac{\sum_{j=1}^{m} z_{ij}^{(t)} x_j}{\sum_{j=1}^{m} z_{ij}^{(t)}}, \quad (2)$$

$$\Sigma_i^{(t+1)} = \frac{\sum_{j=1}^{m} z_{ij}^{(t)} (x_j - \mu_i^{(t+1)})(x_j - \mu_i^{(t+1)})^T}{\sum_{j=1}^{m} z_{ij}^{(t)}}, \quad (3)$$

$$\pi_i^{(t+1)} = \frac{\sum_{j=1}^{m} z_{ij}^{(t)}}{m}. \quad (4)$$

The algorithm stops when $|log(L(\psi^{(t+1)}) - log(L(\psi^{(t)})| < \epsilon$, where ϵ is a preselected threshold.

B. The security model

The privacy preservation of the proposed protocols based on the semi-honest security model. In this model, each party participating in the protocol has to follow the rules using its correct input, and it cannot use what it sees during execution of the protocol to compromise the security. This model is reasonable for many real situations, because the parties who want to mine data for their mutual benefit will follow the protocol to get correct results. The definition of secure two party computation in the semi-honest model is stated in [5]. Here is a summary of the definition.

Let x_1 and x_2 be inputs of two parties and Π be a two-party protocol for computing a function $f : (x_1, x_2) \rightarrow (y_1, y_2)$. The view of the i^{th} party ($i \in \{1, 2\}$) after having participated in protocol Π, denoted by $View_i^{\Pi}(x_1, x_2)$ is $(x_i, r_i, m_{i1}, ..., m_{ik})$, which is the input x_i, all messages $(m_{i1}, ..., m_{ik})$ received by the i^{th} party while executing the protocol and r_i are random bits generated by the i^{th} party. We say that Π privately computes f if there exist probabilistic polynomial-time algorithms S_i ($i \in \{1, 2\}$), such that

$$\{S_i(x_i, y_i)\}_{\{x_1,x_2\}} \stackrel{c}{\equiv} \{View_i^{\Pi}(x_1, x_2)\}_{\{x_1,x_2\}},$$

where c denotes computational indistinguishability. Basically, the definition states that a computation is secure if the view of each party during the execution of the protocol can be effectively simulated by the input and the output of the party. Therefore, in order to prove security of the protocol, we have to show that there exists a simulator for each party i that satisfies the above equation.

In this paper, we also use the composition theorem for the semi-honest model that its discussion and the proof can be found in [5]. The composition theorem states that if a protocol can be decomposed into several sub-protocols, then security of the protocol will be proved if we can show that the subprotocols are secure.

Composition theorem. *Suppose that g is privately reducible to f and that there exists a protocol for privately computing f. Then there exists a protocol for privately computing g.*

3. PRIVACY PRESERVING CLUSTERING FOR THE MULTI-PARTY DISTRIBUTED DATA

Assume that the data set D including m objects $\{x_1, ..., x_m\}$ is horizontally partitioned into n parties, where each party l has a data set D_l including m_l objects, where $m_1 + ... + m_n = m$. Assume that the parties want to cluster the joint data set without revealing anything except for the final results. So, each party could learn the cluster to which each of their data objects belongs, but they learn nothing else. We are assuming that clustering on the joint data set of the n parties is more desirable than clustering on the n data sets individually.

As already pointed out in section 2, the goal of the cluster algorithm is to compute z_{ij}. To obtain z_{ij}, each party needs to know the covariance matrix Σ_i, the vector of means μ_i and π_i in each iteration of the algorithm. We rewrite the equations for computing these parameters as follows:

$$\mu_i^{(t+1)} = \frac{\sum_{l=1}^{n} A_{il}}{\sum_{l=1}^{n} C_{il}}, \tag{5}$$

$$\Sigma_i^{(t+1)} = \frac{\sum_{l=1}^{n} B_{il}}{\sum_{l=1}^{n} C_{il}}, \tag{6}$$

$$\pi_i^{(t+1)} = \frac{\sum_{l=1}^{n} C_{il}}{\sum_{l=1}^{n} m_l}, \tag{7}$$

where

$$A_{il} = \sum_{x_j \in D_l} z_{ijl}^{(t)} x_j, \tag{8}$$

$$B_{il} = \sum_{x_j \in D_l} z_{ijl}^{(t)} (x_j - \mu_i^{(t+1)})(x_j - \mu_i^{(t+1)})^T, \tag{9}$$

$$C_{il} = \sum_{x_j \in D_l} z_{ijl}^{(t)}, \tag{10}$$

and z_{ijl} is the posterior probability of x_j from class i, where $x_j \in D_l$. Denote by $\{z_{ijl}\}$ the set of all z_{ijl} values.

Clearly, A_{il}, B_{il},C_{il} values can be computed locally at each site. Therefore, a method proposed in [1] and [12] for preserving privacy clustering is that, each party locally compute the parameters A_{il}, B_{il} and C_{il}. The global parameters are given by the sum of the local parameters. The secure sum protocol is used to secure compute the global parameters. Thus, the required μ_i, Σ_i and π_i parameters can simply be computed by dividing the appropriate

global sums. After obtaining the global parameters μ_i, Σ_i and π_i, z_{ijl} can be computed locally by equation (1).

However, the given protocol is not completely secure, it reveals the sum result of numerator and denominator, this will reveal a bit extra information. For example, at least one party knows the number of object in each class and this party can guess the upper bound and the lower bound values of attributes of remaining parties if the number of participating parties is small. Specially, with just two parties, each party can get exactly the local parameters of the remaining parties. Thus, the problem is to calculate and share μ_i, Σ_i and π_i parameters without knowing the shared numerator and shared denominator.

To solve this problem, we can use the following secure logarithm approximate method together with secure sum computation method. M.Kantarcioglu and J.Vaidya [10] use this idea for secure probability computation. For example, consider computing μ_i, assuming that $A = \sum_{l=1}^{n} A_{il}$ and $C = \sum_{l=1}^{n} C_{il}$ are in the range $[0...M]$.

- In the first step, party 1 chooses two random numbers R_1 and R_2 in $[0...M]$, and then sends $R_1 + A_{i1} \mod M$ and $R_2 + C_1 \mod M$ to party 2.
- Starting from party 2, each party l receives the results $R_1 + \sum_{j=1}^{l-1} A_{ij} \mod M$ and $R_2 + \sum_{j=1}^{l-1} C_{ij} \mod M$ from party $l-1$, adds A_{il}, C_{il} to these results and passes to the next party. This process stops at party n, it receives $A + R_1 \mod M$ and $C + R_2 \mod M$.
- Party 1 and party n use the secure approximate $\ln(x)$ protocol given in [13], party 1 obtains u_1 and v_1, party n obtains u_n and v_n, where $u_1 + u_n = \alpha \ln(A + R_1 - R_1 \mod M) \mod M$, and $v_1 + v_n = \alpha \ln(C + R_2 - R_2 \mod M) \mod M$, where α is a public constant used to make all elements integer.
- Party n computes $s_n = v_n - u_n \mod M$ and sends it to party 1.
- Party 1 computes $s_1 = s_n + v_1 - u_1 \mod M$.
- Finally, all parties can calculate $\mu_i = \exp(s_1/\alpha)$.

We call this method be secure multi-party division. Using this method we can present the privacy preserving EM clustering for multi-party distributed data as in Protocol 1.

Analysis of privacy: In protocol 1, communication only occurs at steps 6, 8 and 12. At step 6 and 8 it uses the secure multi-party division method to compute and share the global information without disclosing private information. At step 12, computing and sharing the log likelihood difference value do not disclose private information, then applying. Composition theorem, we can conclude that protocol 1 is secure. Indeed, we now check whether the revealed values at the above steps can be used to deduce any information on individual data items.

Protocol 1 Privacy preserving clustering for multi-party distributed data

Input: There are n parties, each party l has the data set D_l $(l = 1, ..., n)$

Output: Each party knows the cluster to which each of their data objects belongs

1: For each $l \in \{1, ..., n\}$, party l randomly initializes z_{ijl} to 0 or 1 ($i = 1...k$, $j = 1...m_l$).
2: t:=0
3: **while** $\delta < \epsilon$ **do**
4: **for** $i = 1...k$ **do**
5: For each $l \in \{1, ..., n\}$, party l computes A_{il} and C_{il} by equations (8) and (10), respectively.
6: All parties jointly compute $\mu_i^{(t+1)}$ and $\pi_i^{(t+1)}$ by the secure multi-party division method. All parties obtain $\mu_i^{(t+1)}$ and $\pi_i^{(t+1)}$.
7: For each $l \in \{1, ..., n\}$, party l uses the result at the previous step to locally compute B_{il} by equation (9).
8: All parties jointly compute $\Sigma_i^{(t+1)}$ by the secure multi-party division method. All parties obtain $\Sigma_i^{(t+1)}$.
9: For each $l \in \{1, ..., n\}$, party l locally computes z_{ijl} by equation (1)
10: **end for**
11: $t = t + 1$
12: The parties jointly compute the log likelihood difference $\delta = |log(L(\psi^{(t+1)}) - log(L(\psi^{(t)})|$
13: **end while**

At step 6 and 8, the parties 2, ...n only involve in the secure sum protocol, thus, each party cannot infer anything except the result. Between two parties 1 and n have communication that are involved in the secure approximate logarithm protocol, party 1 (resp. party n) only received the message u_1 and v_1 (u_n and v_n) that are uniformly distributed in $[0...M]$. The messages are easily simulated by choosing uniformly a number in [0,...M] [13]. In short, each party l only knows the global shares $\mu_i^{(t+1)}$, $\Sigma_i^{(t+1)}$ and $\pi_i^{(t+1)}$. They do not know any private information. These shares are distilled from many data items, so deductions on these values are not possible.

At step 12, the parties compute the log likelihood difference value (δ), we have

$$log(L(\psi^{(t)}) = \sum_{l=1}^{k} E_l^{(t)},$$

where,

$$E_l^{(t)} = \sum_{x_j \in D_l} \sum_{i=1}^{k} log(\pi_i f(x_i; \psi^{(t)})).$$

Therefore, to obtain δ, each party l needs to share its log likelihoods $E_l^{(t)}$ and $E_l^{(t+1)}$. Clearly, disclosing the local log likelihood does not make privacy breaches, because it is distilled from many data items.

Communication cost: The communication cost of this protocol at an iteration is dominated by secure sum computation and calculating logarithm. Therefore, total number of bits transferred will be $O(tk(n + rlog(|M|)))$ bit, where r is the order of Taylor series given in [13]. So, this depends on choosing r.

4. PRIVACY PRESERVING CLUSTERING WITHOUT DISCLOSING CLUSTER CENTERS

The problem of protocol 1 is that it discloses cluster centers. Therefore, in case with only two parties, we should note that sharing Σ_k to each party does not make privacy breaches [11], but sharing means might allow parties to learn some information of each other. For example, each party can guess the upper bound and lower bound values of an attribute of the other party. Moreover, if the global means and the number of objects of each party are together disclosed, the parties can deduce the local means and the local covariance matrices at each site, and then the probability that a data point belonging to a specified interval can be calculated at each site. Another problem is, protocol 1 is based on secure logarithm approximating, so its accuracy depends on the value of parameter r, to ensure that approximate algorithm has an rational accuracy, we need to choose a high value for the parameter r to result in a high communication cost. Therefore, this section presents a better method for the case in which the dataset is horizontally partitioned into only two parts. Our method is more secure, which allows each party to obtain the final result without sharing means.

A. Protocol

We decompose the privacy preserving EM-based clustering problem into three following subproblems:

1) Secure mean computation: Party 1 has a pair (A_{i1}, C_{i1}) and party 2 has a pair (A_{i2}, C_{i2}). They need to jointly compute $\mu_i = (A_{i1} + A_{i2})/(C_{i1} + C_{i2})$ that party 1 obtains μ_{i1} without other information, party 2 obtains μ_{i2} without other information, where $\mu_{i1} + \mu_{i2} = (A_{i1} + A_{i2})/(C_{i1} + C_{i2})$. In other words, we need to implement secure computation for the following functionality (we propose the secure mean computation protocol for this problem in the next section):

$$((A_{i1}, C_{i1}), (A_{i2}, C_{i2})) \longmapsto (\mu_{i1}, \mu_{i2}) | \mu_{i1} + \mu_{i2} = \frac{A_{i1} + A_{i2}}{C_{i1} + C_{i2}}.$$

2) Secure covariance matrix computation: Party 1 (resp. party 2) has the data set D_1, the vector μ_{i1} and the set $\{z_{ij1}\}$ (resp. D_2, μ_{i2} and $\{z_{ij2}\}$). They need to jointly compute Σ_i as shown in equation (3), thus both parties obtain Σ_i without disclosing the local values. In other words, we need to implement secure computation for the following functionality:

$$((D_1, \mu_{i1}, \{z_{ij1}\}), (D_2, \mu_{i2}, \{z_{ij2}\})) \longmapsto (\Sigma_i, \Sigma_i).$$

We consider computing an element of Σ_i, recall that the Σ_i matrix has d rows and d columns and each element $\Sigma_i(p,q)$ of Σ_i ($p \in \{1,...,d\}$, $q \in \{1,...,d\}$), computed by formula:

$$\Sigma_i(p,q) = \frac{\sum_{j=1}^{m} z_{ij}(x_j[p]-\mu_i[p])(x_j[q]-\mu_i[q])}{\sum_{j=1}^{m} z_{ij}}$$

The numerator of $\Sigma_i(p,q)$ can be presented as the scalar product of two vectors $U = (a_1, b_1, c_1, d_1, e_1, 1)$ and $V = (1, e_2, d_2, c_2, b_2, a_2)$, where, for each $l \in \{1,2\}$

$$\begin{aligned}
a_l &= \sum_{x_j \in D_l} z_{ijl}(x_j[p]-\mu_{il}[p])(x_j[q]-\mu_{il}[q]) + \mu_{il}[p]\mu_{il}[q],\\
b_l &= -\sum_{x_j \in D_l} z_{ijl}(x_j[p]-\mu_{il}[p]),\\
c_l &= -\sum_{x_j \in D_l} z_{ijl}(x_j[q]-\mu_{il}[q]),\\
d_l &= \mu_{il}[p],\\
e_l &= \mu_{il}[q].
\end{aligned}$$

It should be noted that U can be computed by party 1 alone and V can be computed by party 2 alone. Therefore, in order to compute $\Sigma_i(p,q)$. Firstly, two parties privately compute the numerator using the scalar product protocol in [6], party 1 obtains u and party 2 obtains v, where $u+v = U \bullet V$. Secondly, they can use Protocol 3 to compute $\Sigma_i(p,q) = (u+v)/(C_{i1}+C_{i2})$

3) Secure posterior probability computation: party 1 (resp. party 2) has μ_{i1} (resp. μ_{i2}). Both party share π_i and Σ_i. One party (assume party 1) having an object x_j wants to compute z_{ij1} (the posterior probability of x_j from class i), it can cooperate with party 2 to compute z_{ij1}, thus party 1 obtains z_{ij1}, party 2 obtains nothing. In other words, we need to implement a secure computation for the following functionality:

$$((\Sigma_i, \pi_i, \mu_{i1}, x_j), (\Sigma_i, \pi_i, \mu_{i2})) \longmapsto (z_{ij1}, \phi).$$

To obtain z_{ij}, party 1 needs to obtain $T_j = (x_j - \mu_i)^T \Sigma_i^{-1}(x_j - \mu_i)$ and then we can

compute z_{ij} by equation (1). Denote

$$\begin{aligned}
\alpha_1[q] &= \sum_{p=1}^{d}(x_j[p]-\mu_{i1}[p])\Sigma_i(p,q),\\
\beta_1[q] &= -\sum_{p=1}^{d}\mu_{i2}[p]\Sigma_i(p,q),\\
\alpha_2[q] &= x_j[q]-\mu_{i1}[q]),\\
\beta_2[q] &= -\mu_{i2}[q],\\
\alpha &= \sum_{q=1}^{d}\alpha_1[q]\alpha_2[q],\\
\beta &= \sum_{q=1}^{d}\beta_1[q]\beta_2[q]
\end{aligned}$$

.We can rewrite the equation of T_j as a scalar product of two following vectors:

$$X = (\alpha, \alpha_1[1],\ ...\ \alpha_1[d], \alpha_2[1],\ ...\ \alpha_2[d], 1),$$

$$Y = (1, \beta_2[1], ..., \beta_2[d], \beta_1[1], ..., \beta_1[d], \beta),$$

where, X can be computed by party 1 alone; Y can be computed by party 2 alone. Therefore, the parties can compute the dot products $X \bullet Y$ using the scalar product protocol in [6]. Party 1 obtains T_j without disclosing private information and then he can compute z_{ij} as equation (1).

Based on the above addressed problems, the protocol is formally described in Protocol 2.

Analysis of privacy: In protocol 2, only interaction occurs at steps 6, 7, 8, 9 and 12. At each step, it uses the secure scalar product and protocol 3 to compute and share the global information, we should note that where there already exist many scalar product protocols that are correct and secure [6], [19]. During the execution of this protocol, the parties participating in the protocol are not able to learn anything other than the final result. Therefore, assume that the used protocol 3 is secure (the security of this protocol is proved in the next section). Then applying the composition theorem, we can conclude that protocol 2 is secure. Indeed, we now check whether the revealed values at the above steps can be used to deduce any information on individual data items.

At step 6, each party l only obtains the random values μ_{il}, they do not know any other information including μ_i, so deduction on these values is not possible.

The Σ_i matrix and the π_i value are shared at step 7 and 8, respectively, by themself, they do not reveal private information, because they are distilled from many data values at sites. At step 9, each party l obtains its z_{ijl} without sharing with the other party. At step 12, the

Protocol 2 Privacy preserving EM-based clustering without disclosing cluster centers

Input: Party 1 and party 2 have sets D_1 and D_2, respectively
Output: Each party knows the cluster to which each of their data objects belongs

1: For each $l \in \{1, 2\}$, party l randomly initializes z_{ijl} to 0 or 1 ($i = 1...k$, $j = 1...m_l$).
2: t:=0
3: **while** $\delta < \epsilon$ **do**
4: **for** $i = 1...k$ **do**
5: For each $l \in \{1, 2\}$, party l computes A_{il} and C_{il} by using equations (8) and (10).
6: Two parties jointly compute $\mu_i^{(t+1)}$ by the secure mean computation method (Protocol 3). Each party l obtains $\mu_{il}^{(t+1)}$.
7: Two parties jointly compute $\Sigma_i^{(t+1)}$ by the secure covariance matrix computation method.
8: Two parties jointly compute $\pi_i^{(t+1)}$ by Protocol 3.
9: For each $l \in \{1, 2\}$, party l cooperates with the other party to compute z_{ijl} by the secure posterior probability computation method. Party l obtains z_{ijl}, the other party obtains nothing.
10: **end for**
11: $t = t + 1$
12: Two parties jointly compute the log likelihood difference $\delta = |log(L(\psi^{(t+1)}) - log(L(\psi^{(t)})|$
13: **end while**

parties compute the log likelihood difference value, and to obtain this value, each party l has to share its log likelihood with the other party. As analyzed in the previous section, disclosing the local log likelihood does not make privacy breaches.

The communication analysis: We give an analysis of the communication cost of the protocol at its one iteration. The total cost is dependent on the number of iterations required to converge, which is dependent on the data. We should note that the communication cost of the scalar product protocol is $O(tn)$ bit (see in [6]), where n is the size of input vectors and the communication cost of the oblivious polynomial evaluation protocol is $O(tk)$ exponentiations or $O(tk|F|)$ [16], where k is the degree of the input polynomial and $|F|$ is the size of the field used and depends on the range of the variables in calculation.

Assume that the communication cost for problems 1, 2 and 3 are P_1, P_2 and P_3, respectively. To address problem 1, protocol 3 calls the oblivious polynomial evaluation protocol three times (with the degree 1 polynomials), so $P_1 = O(t)$. To address problem 2, it calls the scalar product protocol d^2 times with the size of input vectors 6, $P_2 = O(td^2)$. Similarly, $P_3 = O(t)$.

In one iteration of the Protocol 2, the communication occurs at steps 6, 7, 8 , 9. Then, its communication cost is $P = 2P_1 + P_2 + mP_3$. Finally, we have $P = O(t) + O(td^2) + mO(t) = O(t(d^2 + m))$. In fact, d is a small constant, thus $P = O(tm)$, this is quite reasonable.

B. Secure mean computation protocol

In this section, we propose a protocol for the secure mean computation problem based on the oblivious polynomial evaluation. The problem of the oblivious polynomial evaluation was first considered in [15]. As with oblivious transfer, this problem involves a sender and a receiver. The senders input is a polynomial Q of degree k over some finite field F and the receivers input is an element $z \in F$ (the degree k of Q is public). The protocol is such that the receiver obtains $Q(z)$ without learning anything else about the polynomial Q, and the sender learns nothing. An efficient solution to this problem was presented in [16]. Protocol 3 defines the secure mean sharing protocol.

Protocol 3 Secure mean sharing

Input: Assume that two parties Alice and Bob have $(n,\ x)$ and $(m,\ y)$, respectively.
Output: Alice obtains r_1, Bob obtains r_2

1: Alice uniformly chooses an element p from F and defines the linear polynomial

$$Q_1(z) = pz + pn.$$

2: Alice and Bob engage in a private evaluation of Q_1, in which Bob obtains

$$b_1 = Q_1(m) = pm + pn.$$

3: Bob chooses a random element $q \in F$ and defines the linear polynomial

$$Q_2(z) = yz - (pm + pn)q.$$

4: Alice and Bob engage in a private evaluation of Q_2, in which Alice obtains

$$a_1 = Q_2(p) = py - (pn + pm)q.$$

5: Alice chooses a random element $r \in F$ and defines the linear polynomial

$$Q_3(z) = -rz + py + px - (pn + pm)q.$$

6: Alice and Bob engage in a private evaluation of Q_3, in which Bob obtains

$$b_2 = Q_3(pn + pm) = -r(pn + pm) + py + px - (pn + pm)q.$$

7: Alice has $r_1 = r$ and Bob computes

$$r_2 = \frac{b_2}{b_1} + q = -r + \frac{x + y}{n + m}.$$

So, the respective outputs of Alice and Bob are r_1 and r_2, giving us that

$$r_1 + r_2 = \frac{x + y}{n + m}.$$

Analysis of privacy: We can easily prove that Protocol 3 constitutes a private protocol for computing the mean value as stated. Indeed, we can show that each party's view of the protocol can be simulated based on its input and its output.

During execution of the protocol, Alice only sees the message $a_1 = py - (pn + pm)q$, where q uniformly selected from F, and y, n and m are constants. Assume that $p' \in F(\neq p)$, $a'_1{=}p'y - (p'n + p'm)q$, we have a_1 and a'_1 be the uniform distribution on a specified set, the probability that Alice see some values during the execution is $1/|F|$. Therefore, the two ensembles a_1 and a'_1 are statistically indistinguishable. In other words, the simulator for Alice will be a uniform number generator. Similarly, Bob sees the messages $pm + pn$ and $-r(pn + pm) + py + px - (pn + pm)q$, these two messages are independent because p and r are independently chosen by Alice, moreover they have the uniform distribution on a specified set and thus they can be simulated by uniform number generators.

5. CONCLUSION

We have presented the expectation maximization mixture model clustering method for distributed data that preserves privacy for data of participating parties. Firstly, privacy preserving EM-based clustering method for multi-party distributed data proposed. Unlike the existing method, our method does not reveal sum results of numerator and denominator in the secure computation for the parameters of EM algorithm, therefore, the proposed method is more secure and it allows the number of participating parties to be arbitrary. Secondly, we propose the better method for the case in which the dataset is horizontally partitioned into only two parts, this method allows computing covariance matrices and final results without revealing the private information and the means. To solve this one, we have presented a protocol based on the oblivious polynomial evaluation and the secure scalar product for addressing some problems, such as the means, covariance matrix and posterior probability computation. The approach of paper allows two or many parties to cooperatively conduct clustering on their joint data sets without disclosing each party's private data to the other.

REFERENCES

[1] Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X., and Zhu, M. Y. 2002. Tools for privacy preserving distributed data mining. *SIGKDD Explor. Newsl.* 4, 2 (2002), 28-34.

[2] Dempster, A. P., Laird N. M., Rubin D. B. 1977. 'Maximum likelihood from incomplete data via the EM algorithm (with discussion)' *J Roy Stat Soc B 39* (1977), 1-38.

[3] Du, W., Chen, S. Han, Y.S. 2004. 'Privacy-preserving multivariate statistical analysis: Linear regression and classification', *Proceedings of the Fourth SIAM* (Lake Buena Vista, Florida, USA 2004), 222-233.

[4] Evfimievski, A. 2002. 'Randomization in Privacy Preserving Data Mining', *ACM SIGKDD Explorations Newsletter*, Volume 4, Issue 2 (2002), 43-48.

[5] Goldreich, O. 2004. *Foundations of Cryptography: Volume 2, Basic Applications*, Cambridge University Press (2004).

[6] Goethals, B., Laur, S., Lipmaa, H. and Mielikainen, T. 2004. 'On private scalar product computation for privacy-preserving data mining', *In Proc. of the Seventh Annual International Conference in Information Security and Cryptol- ogy, LNCS.* Springer-Verlag, volume 3506 of Lecture Notes in Computer Science (2004), 104-120.

[7] Han, J. and Kamber, M. 2001. *Data Mining Concepts and Techniques*, Morgan Kaufmann Publishers (2001).

[8] Jagannathan, G. and Wright, R. N. 2005. Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In Proceedings of the Eleventh ACM SIGKDD international Conference on Knowledge Discovery in Data Mining (Chicago, Illinois, USA, August 21 - 24, 2005). KDD '05. ACM, New York, NY, 593-599.

[9] Jha, S., Kruger, L., McDaniel, P. 2005. 'Privacy Preserving Clustering', *In Proc. of the 10th European Symposium on Research in Computer Security* (2005), 397-417.

[10] Jaideep Vaidya, Murat Kantarcioglu and Chris Clifton, 2008. "Privacy Preserving Naive Bayes Classification" The VLDB Journal, VLDB Endowment, 17(4) (2008), 879-898.

[11] Kantarcioglu, M. 2005. Privacy-Preserving Distributed Data Mining and Processing on Horizontally Partitioned Data. Doctoral Thesis. UMI Order Number: AAI3191494., Purdue University (2005).

[12] Lin, X., Clifton, C., and Zhu, M. 2005. Privacy-preserving clustering with distributed EM mixture modeling. Knowl. Inf. Syst. 8, 1 (Jul. 2005), 68-81.

[13] Lindell, Y. and Pinkas, B. 2000. 'Privacy Preserving Data Mining', *Advances in Cryptology (CRYPTO'00)*, Springer-Verlag (LNCS 1880), 36-53.

[14] McLachlan, G.J., Basford, K.E. 1988). 'Mixture models: inference and applications to clustering', Dekker, New York (1988).

[15] Naor, M. and Pinkas, B. 1999. Oblivious transfer and polynomial evaluation. In Proceedings of the Thirty-First Annual ACM Symposium on theory of Computing (Atlanta, Georgia, United States, May 01 - 04, 1999). STOC '99. ACM, New York, NY, 245-254.

[16] Naor, M. and Pinkas, B. 2001. Efficient oblivious transfer protocols. In Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (Washington, D.C., United States, January 07 - 09, 2001). Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, Philadelphia, PA, 448-457.

[17] Pinkas, B. 2002. Cryptographic techniques for privacy-preserving data mining. SIGKDD Explor. Newsl. 4, 2 (Dec. 2002), 12-19.

[18] Verykios, V. S., Bertino, E., Fovino, I. N., Provenza, L. P., Saygin, Y., and Theodoridis, Y. 2004. State-of-the-art in privacy preserving data mining. SIGMOD Rec. 33, 1 (Mar. 2004), 50-57

[19] Vaidya, J. and Clifton, C. 2002. Privacy preserving association rule mining in vertically partitioned data. In Proceedings of the Eighth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining (Edmonton, Alberta, Canada, July 23 - 26, 2002). KDD '02. ACM, New York, NY, 639-644.

[20] Vaidya, J. and Clifton, C. 2003. Privacy-preserving k-means clustering over vertically partitioned data. In Proceedings of the Ninth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining (Washington, D.C., August 24 - 27, 2003). KDD '03. ACM, New York, NY, 206-215.

Received on May 7 - 2010