Privacy Preserving Data Mining

Dilys Thomas

1 Introduction

We consider protocols for privacy preserving data mining today. This work aims to be of the flavor of secure multi-party computation. We study two problems, association rule mining in vertically partitioned data and association rule mining in horizontally partitioned data from [1, 2].

A common theme among the protocols is that they will be very efficient as compared to the protocols obtained from Secure Multi-Party computation framework. However they will leak a little bit of extra information. This leaked information can be characterized and in some settings it makes sense to tolerate this leaked information for the great performance benefits.

2 Association Rule Mining in Horizontally Partitioned Data

Horizontally partitioned data is data such that each site has complete information on a set of entities. Each site thus has the same set of attributes but different sites have information about different entities. We are interested in finding association rules that satisfy two criteria

1. If s_i is the support at the i^{th} site $1 \leq i \leq n$, then $s_i > c$
2. $\Sigma_{i=1}^{n} s_i > Sum$

A-priori algorithm[3] is run to generate all frequent sets at each site. For each set S finding if the above two conditions hold is tested by two simple protocols. We provide them as black boxes. Using them is an easy task.

2.1 Intersection Protocol

We use commutative encryption for the purposes of intersection. An encryption scheme is called commutative is $E_{k1}(E_{k2}(M)) = E_{k2}(E_{k1}(M))$. The identity $(g^x)^y = (g^y)^x$ makes intuitive the existence of commutative encryption schemes. To compute if a set S is frequent at all sites. Each site encrypts its frequent sets and passes it onto the next site and so on. So that each frequent set at each set is repeatedly encrypted so that all the participants encrypt it exactly once with their key. For example if A is a frequent set (ordered lexicographically before

encryption say) at site X and if X, Y, Z are the three sites with horizontally partitioned data having keys x, y, z respectively. Then X computes $E_x(A)$ and passes it onto Y who in turn computes $E_y(E_x(A))$ and passes it onto Z, who in turn encrypts it under its key to get $E_z(E_y(E_x(A)))$. thus the set is encrypted by all the parties X, Y, Z. If A were frequent at site Y too and Y encrypted it first giving it to X and then Z it would be encrypted as $E_z(E_x(E_y(A)))$ which is equal to $E_z(E_y(E_x(A)))$. Similarly if A were frequent at site Z too it would be encrypted by all to give the same result. One all the items that are encrypted there will be three identical copies among the encrypted items corresponding to only the frequent item sets. And the three copies correspond to the fact the same item was generated by all participants and encrypted by all and since the encryption is commutative the final encrypted version of A after being encrypted by all sites is the same. Now all the encrypted items are shipped to one site and all those having copies whose multiplicity equals the number of sites are decided to belong to the intersection of all the frequent sets. These sets are then disclosed to all by repeatedly decrypting in reverse order from the encrypted item. Thus the intersection set of frequent item-sets can be identified.

2.2 Sum Protocol

Now we present a protocol to check if n sites with values s_i $1 \leq i \leq n$ satisfy $\Sigma s_i > Sum$. Site 1 selects a random number r adds it to s_1 and passes it onto site 2. Site 2 gets $s_1 + r$ from Site 1 it adds s_2 to it and passes it onto site 3. Site i receives $\Sigma_{j=1}^{i-1} s_j + r$ from site $i-1$ adds s_i to it before passing it to site $(i+1)$. Finally the Site n passes $\Sigma_{j=1}^{n} s_j + r$ to Site 1 which then subtracts r to get $\Sigma_{j=1}^{n} s_j$. it then tests whether this is $> Sum$ to decide whether the element satisfies the second criterion.

2.3 Extra information learned

Note that the above protocol shows more information than the secure multi-party computation protocol. All items that satisfy condition 1 but not 2 are known at all sites. Also site 1 in the protocol for the second stage actually learns $\Sigma_{j=1}^{n} s_j$.

3 Association Rule Mining in Vertically Partitioned Data

The problem of association rule mining on vertically partitioned data can be reduced to computing dot products of vectors securely. We will hence only concentrate on this part of the protocol.

3.1 Dot Product Protocol

To compute the dot product of two vectors $X = (x_1, x_2, ...x_n)$ and $Y = (y_1, y_2, ...y_n)$ held by two parties A, B. Let
$a_{1,1}, a_{1,2}, ...a_{1,n/2}$,
$a_{2,1}, a_{2,2}, ...a_{2,n/2}$,
$a_{3,1}, a_{3,2}, ...a_{3,n/2}$,
$a_{n,1}, a_{n,2}, ...a_{n,n/2}$
be $\frac{n^2}{2}$ random numbers shared between A and B. Let A generate $n/2$ random numbers $R_1, R_2, ...R_{n/2}$

A computes and sends
$x_1 + a_{1,1}R_1 + ...a_{1,n/2}R_{n/2}$
$x_1 + a_{1,1}R_1 + ...a_{1,n/2}R_{n/2}$
$x_1 + a_{1,1}R_1 + ...a_{1,n/2}R_{n/2}$
$x_1 + a_{1,1}R_1 + ...a_{1,n/2}R_{n/2}$
to B who computes the dot product of this with its vector to get $\Sigma_{i=1}^{n}(y_i(x_i + a_{i,1}R_1 + ... + a_{i,n/2}R_{n/2} = \Sigma_{i=1}^{n}(x_i y_i) + \Sigma_{i=1}^{n/2}R_i(a_1, i)y_1 + a_{2,i}R_2 + ...a_{n,i}y_n$. Now if B sends to A the $n/2$ values $(a_{1,i}y_i + a_{2,i} + ... + a_{n,i}y_n)$ for $1 \leq i \leq n/2$ A can compute the dot product. By ensuring that B reveals only n/2 equations in n variables it can be ensured that A does not learn the values of B. In fact the $n^2/2$ $a_{i,j}$ values can be selected appropriately for the same.

3.2 Leaked Information

The $n/2$ equations are revealed. Selecting the $n^2/2$ variables appropriately can avoid leaking information about these.

References

[1] Chris Clifton and Jaideep Vaidya "Privacy preserving association rule mining on vertically partitioned data." *SIGKDD 2002*

[2] Chris Clifton and Murat Kantarcioglu "Privacy preserving distributed mining of association rules on horizontally partitioned data" *DMKD 2002*

[3] Ramakrishnan Srikant and Rakesh Agrawal "Fast Algorithms for mining association rules" *VLDB 1994*