T-79.511 Special Course on Cryptology / Privacy-preserving K-means clustering

Tatiana Issaeva

September 30, 2003

Overview

The paper under review deals with the K-means clustering data. K-means clustering is a data mining technique, which divides data into K groups (clusters) according to data's positioning to each other.

Authors provide algorithms for running the K-means in multiparty mode in a privacy-preserving fashion under the assumption, that different parties contain different data for the same entities.

Introduction

Clustering techniques is widely used in many mathematical and technical disciplines for finding groups of items with similar parameters. K-means clustering is a technique that divides data into K clusters placed in a multidimensional space around K means. The dimension of the space is the number of the attributes that are being used for clustering. A cluster contains items whose attribute values are similar.

There is nothing complicated to clustering of data. The clustering technique is the same if one needs to cluster data located on multiple parties, only the communication aspect makes the task more demanding. The communication of data between the parties is the main problem authors of this paper tried to solve. The communicated data means that other parties learn about specific details of each others data. If the communicated data is in any way private, like bank records, the data communication becomes a privacy violating exercise.

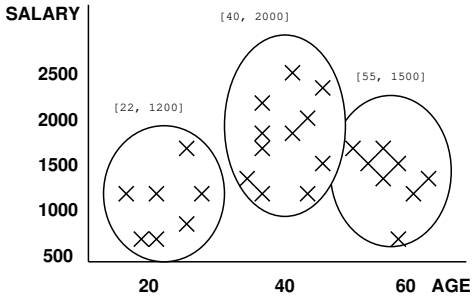
Authors tailored their paper on the data mining between several parties. Each party has different set of attributes for the same set of entities, e.g. for each citizen the banks can have account and transaction information, the police can have criminal records and the health service can hold health records. Direct exchange of the data between these entities is impossible as it violates one's right to privacy.

The proposed privacy-preserving clustering modifies the clustering algorithm and the data communication heavily to withold each party's data from other sites but at the same time allow for multisite clustering.

System details

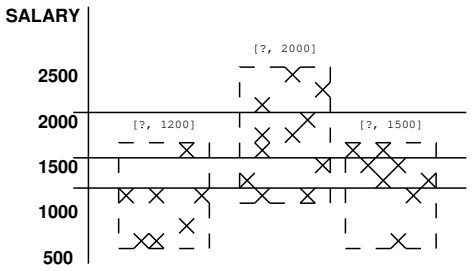
K-means clustering

K-means clustering groups items into K groups according to L attributes. The number of attributes L creates a L-dimensional space in which each item is located based on the values of its attributes. E.g. when clustering by person's age and salary, we create two-dimensional space with axes of age and salary. We can then cluster the data into K clusters, so that each person will at the end belong into one of the K clusters which are centered around K means (cluster centers - they define a cluster, not its shape). The way clusters will form depends on the initial cluster coordinates and on the data we have (e.g. no young and high salaried people - no such cluster).



Privacy-preserving K-means clustering

The task here is to cluster data from R parties, each having different attributes for the same set of entities N. The parties wish to cluster their joint data using the k-means algorithms without disclosing its own data to other parties. The result of the clustering would be positions of the clusters with each party receiving only the relevant part of the cluster means according to the attributes the party contributed to the computation. E.g. the party contributing the salary attributes will receive only the salary coordinates of the means, as demonstrated on the following picture:



Each party contributes only its own data and receives only its own portion of the means' coordinates and assignment of entities to clusters so the privacy of others' data is preserved. This is the idea, now lest have a look how it can be done.

To provide privacy, the parties do not communicate the actual data at all. Each party computes the distance between its data and its part of clusters' means locally and communicates only the results. The transmission of the distances vector is never in plaintext but is always encrypted by homomorphic encryption scheme. The elements of the distances vector are further permuted (scrambled) to hide actual values placement.

The conditions just described above would not give any means for computations at other parties as the data would be encrypted and scrambled, so the algorithm depends on three parties, that take specific roles in the K-means computation. They can be either the parties holding the data or completely external parties introduced just for the purpose of secure K-means clustering computation. In either case they must not cooperate, otherwise the actual data can be exposed by combining special sites' knowledge of the scrambling and encryption parameters.

Party P1 as initiator

The paper introduces the special party P1, which is the initiator of the K-means computation and which is also decrypting and distributing the real results at the end of the computation.

The P1 firstly generated separate vector V for each party. Each vector V has dimension K (number of desired clusters) and sum of all the V vectors (1..K) gives a null vector. Therefore the elements in each dimension of the vectors V are have pseudo random distribution. These vectors are to be used as random shift operators of party's own data - party sums its vector of distances X with the random vector V and the resulting vector of data makes no sense to anybody not knowing the original vector V.

Party P1 also generates permutation π which is same for each party, used to further mangle the data by permuting each party's vector :π(X + V). This way the data is shifted (X + V) and also scrambled π(X + V).

Interesting part that comes to play now is the homomorphic encryption function - H. This function is a public key encryption function. What makes it homomorphic is the fact, that there exists an efficient way to compute H(x + y) from H(x) and H(y) without revealing the neither x nor y. Without more details, lets just take any such homomorphic function H and use it in a permutation part of the K-means algorithm.

Use of the homomorphic function allows for not actually revealing the vector V to its party and not revealing the permutation $\pi$to its party - they both

stay safe with its creator - P1. Using the public-key encryption of the H function, the P1 and other parties exchange only encrypted vectors V and X. Each party encrypts its own X vector and sends it to P1 and receives $\pi H(X^{'} + V^{'})$ - permuted sum of encrypted vectors, from which it can decrypt the actual permuted sum $\pi(X+V)$. After such a setup the following transmissions will be secure.

Parties P2 and Pr - computing the distances

Special parties P2 and Pr (there are r-parties involved in the k-means computation) get involved when the we actually run the k-means algorithm. All parties except P2 and Pr now send their vectors to Pr. The data being send by each party is its own permuted vector of the sum : $\pi(X+V)$. Note that each party does not know its vector V or common permutation π, they just know the result $\pi(X+V)$ from the initial interaction with P1.

Now the parties P2 and Pr will engage in a secure addition and comparison operation to find out the closest cluster. As Pr now has the vectors from all parties except P2, and since the sum of all the V vectors gives a null vector, party Pr can in cooperation with P2 find the closest cluster. This is done by a secure circuit evaluation procedure, that does not reveal the components of the comparison $a_2 + a_r < b_2 + b_r$, but gives the comparison result. The parties Pr and P2 now have to make $k-1$such secure multiparty comparisons to determine the closest cluster.

Party P1 as result distributor

After all the comparisons, parties Pr and P2 know the closest cluster in the permutation π. Because the actual permutation $\pi$is not known to them, they have to send their result (i - the index of the closest cluster) back to P1. P1 removes the permutation from the i-th cluster and broadcasts the results as the proper cluster for the computed entity.

Conclusion

The method for privacy-preserving multiparty k-means clustering computation is given. The paper explains well the basic ideas behind the k-means clustering and defines the privacy problem for the multiparty computation. Authors propose a privacy-preserving scheme under which the computation of the k-means algorithm is centralized at three special non-colluding parties, data is transmitted as encrypted and scrambled and results of the clustering give do not reveal other parties data. Security proof for the multi-party computation, cluster computation and overall k-means is presented.

The security of this solution is based on the assumption that the special parties P1, P2 and Pr will not cooperate to gain other data of parties. In fact, the cooperation of the parties P2 and Pr is enough to reveal the specific data of the other parties. The security aspects add additional load of communication and computation and this is discussed in the paper and optimizations are proposed.

Unfortunatelly the description of the leter stages of the algorithm is not very specific and more information would be needed for better understanding and assesment. For example the actions of P1 at the distribution of the results is not described well at all.