Privacy-preserving K-means clustering

Tatiana Issaeva

November 5, 2003

Introduction

- K-means clustering is a technique that divides data into K clusters placed in a multidimensional space around K means

- in some cases one needs to cluster data located on multiple parties

- communication of data between the parties is added to the task

- hence other parties learn about specific details of each others data

- if the communicated data is private -> data communication is a privacy violation

K-means clustering

- K-means clustering groups items into L groups according to L attributes

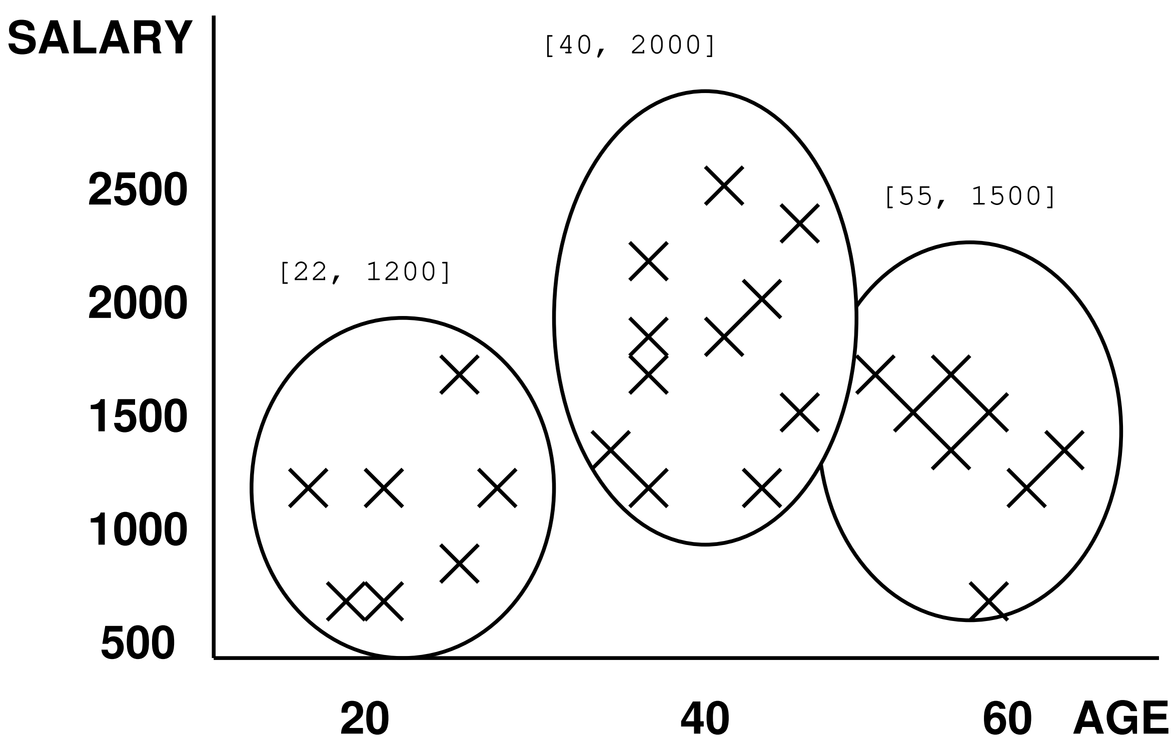
- number of attributes L creates a L-dimensional space in which each item is located based on the values of its attributes

- cluster is defined by its center (mean), not its shape

- data is clustered into K clusters, which are centered around K means

K-means clustering

Example: clustering by person's age and salary, we create two-dimensional space with axes of age and salary

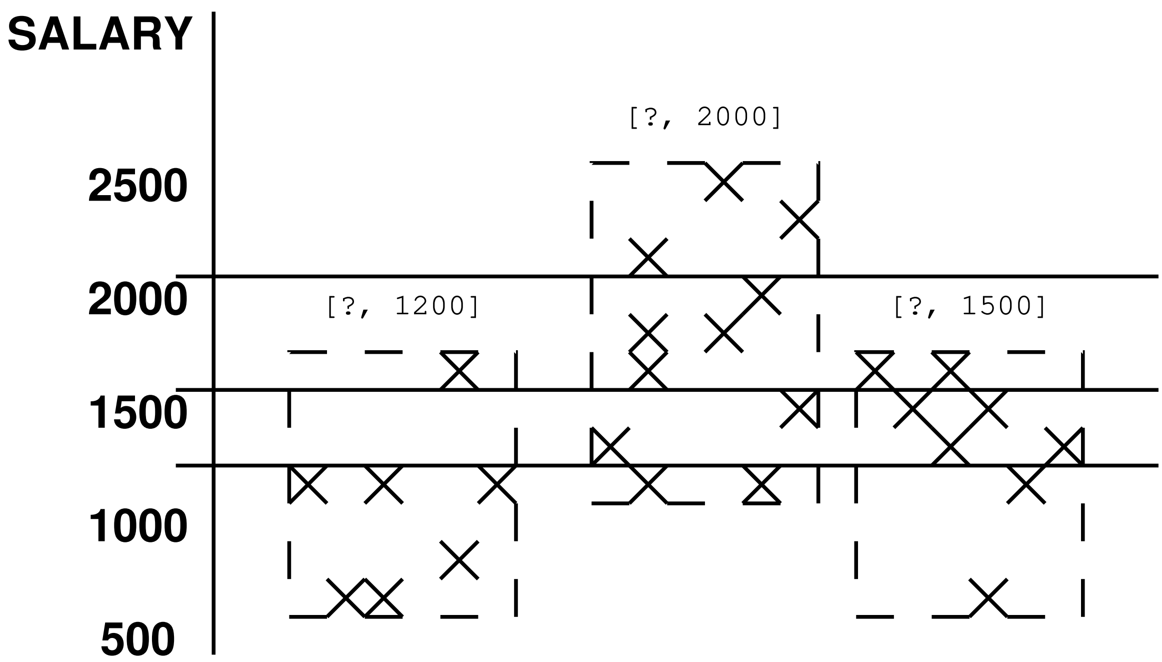


Privacy-preserving K-means clustering

- task here is to cluster data from R parties
- each party has different attributes for the same set of entities
- parties wish to cluster their joint data using the K-means algorithms without disclosing its own data to other parties
- 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

Privacy-preserving K-means clustering

Example: the party contributing the salary attributes will receive only the salary coordinates of the means:



Privacy-preserving K-means clustering

Idea:

- to provide privacy, the parties do not communicate the actual data at all
- transmission of data is always encrypted by homomorphic encryption scheme
- algorithm requires also involvement of three parties, with specific roles in the K-means computation: P_1, P_2 and P_r
- they can be either the parties holding the data or completely external parties
- they must not cooperate

Party P_1 as algorithm initiator

- special party P_1 is the initiator of the K-means computation

- P_1 generates vector V_i for each party and permutation π that is the same for all parties

- $\sum_1^K V_i = 0$

- data X_i is shifted $(X_i + V_i)$ and also scrambled $\pi(X_i + V_i)$

Party P_1 as algorithm initiator

- algorithm takes use of homomorphic encryption function H

- H is a public key encryption function

- there exists an efficient way to compute $H(x+y)$ from $H(x)$ and $H(y)$ without revealing neither x nor y

Algorithm step 1

- each party encrypts its own X_i vector and sends it to P_1

- and receives $\pi H(X_i^{'} + V_i^{'})$ - permuted sum of encrypted vectors, from which it can decrypt the actual permuted sum $\pi(X_i + V_i)$

- by using H function, the P_1 and other parties exchange only encrypted vectors V_i and X_i

- vector V_i and permutation π are not revealed to any party

P_2 and P_r computing the distances, step 2

- all parties except P_2 and P_r now send their vectors to P_r

- the data being send by each party is its own permuted vector of the sum : $\pi(X_i + V_i)$

P_2 and P_r computing the distances, step 3

- P_r now can in cooperation with P_2 find the closest cluster

- this is done by a secure circuit evaluation procedure, which aim of is to disclose nothing but the comparison result

- P_r and P_2 will make $K-1$ such secure multiparty comparisons to determine the closest cluster

P_1 as result distributor, step 4

- P_r and P_2 know the closest cluster in the permutation π and π is not known to them

- P_r and P_2 send their result (the index of the closest cluster) back to P_1

- P_1 removes the permutation from the i-th cluster and broadcasts the results as the proper cluster for the computed entity

Handling collusion

- special parties P_1, P_2 and especially P_r learn quite much during the computation process

- if the parties P_2 and P_r would collude they could easily derive the distances between the points and the means for each party

- parties P_2 and P_r must be selected very carefully so that it is in their own interest to not collude

Communication cost

The cost of one iteration of the K-means algorithm requires:

- $O(r + k)$ rounds of communication, with each round cost of $O(nrk)$ bits
- r is the number of parties, k is the number of desired clusters, n is the number of entities

Total communication cost is dependent on the number of iterations required for the K-means algorithm to converge and is unpredictable

Security aspects of the algorithm

- security is based on the assumption that the special parties P_1, P_2 and P_r will not cooperate to gain data of other parties

- cooperation of P_2 and P_r is enough to reveal the specific data of the other parties

- security aspects add additional load of communication and computation

- requirement for three special parties P_1, P_2 and P_r limits usability of this solution to larger scale multiparty computations.

Conclusion

- method for privacy-preserving multiparty K-means clustering computation is presented

- privacy-preserving scheme is proposed

- 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 do not reveal other parties' data.

References

[1] Jaideep Vaidya, Chris Clifton. Privacy-Preserving K-Means Clustering over Vertically Partitioned Data. In Proceeedings of the Ninth ACK SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM 2003.