Privacy-Preserving K-Means Clustering over Vertically Partitioned Data

Lan Zhang

I implemented Privacy-Preserving K-Means Clustering over Vertically Partitioned Data.

The key idea of this paper is to cluster data that is vertically partitioned into different parties. Each party contain different attributes for a common set of entities. Each party learns the cluster of each entity, but learns nothing about the attributes at other parties.

Algorithm

The algorithm contain 4 parts.

1. Initilization. Each party will initilize a center for each cluster. In our example, we have 4 cluster. So each party holds four vectors as the center of clusters.
2. Find closet cluster. First, we calculate the distance to each cluster for each data points. This calculation is done inside each party and they will hold their distance vectors (X). Then, we will securely find the closest cluster. The master party generates a random vector (V) for each party. Each slave party will generate a public-private keypair for a homomorphic encryption scheme. And each slave party (party i) send the encoded distance vector (x_i) and public key to the master party. The master party encoded the random vector (v_i) , calculated the sum of the encoded distance vector (x_i) and the encoded random vector (v_i), permuted the result, and finally send it back to the slave party. The slave party decode the final result. Now, for each slave party, they have the permuted sum (t_i) of the distance vector (x_i) and the random vector (v_i). It is secure because each slave party don't know the random vector and permutation vector. Next, all parties except two slave party (party 2 and party r) will send the permuted sum (t_i) to party r. So party r can calculate the sum (y) of all permuted sum except the result of party 2. Finally, between party 2 and party r, they will securely add and compare y with t_2 to find the minimal cluster. Notice, till now, the calulation is done on permuted data, so we need the master party to find the true value of the closet cluster.
3. Calculate clustering means for each party. Every point is assigned to the proper cluster. We securely find the cluster with the minimum distance for each point. Once

these mappings are known, the local components of each cluster mean can be computed locally. After cluster all data points, we then calcuate the means of each cluster with attributes in each party.

4. Check Threshold. Find out if the new means are sufficiently close to old means. First, we calculated the distance of new clustering means and original means. Then, we securely sum the distance. Finally, we securely add and compare the threshold with the sum.

Discussion

Advantages

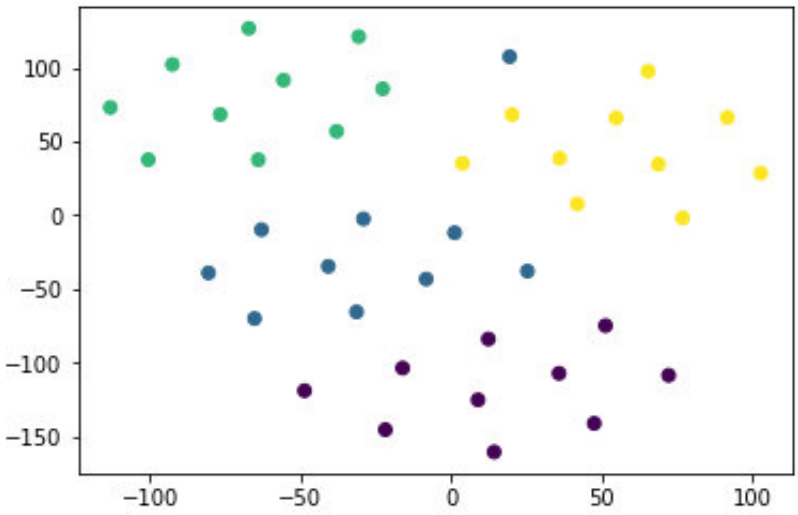
1. The algorithm ensures data privacy by calculating cluster mean locally and securely find the closet cluster.
2. According to their communication analysis , their work ensures reasonable privacy while limiting communication cost. The total number of rounds required is O(r + k). The bit communication cost is O(nrk).
3. I think it maybe possible to apply the some algorithms in this paper to other clustering algorithm like hierarchical clustering.

Disadvantages

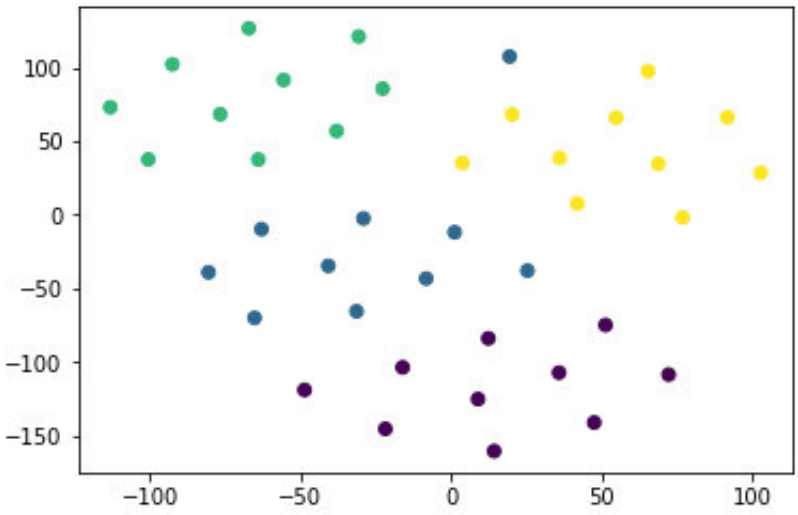
1. I think one assumption in this paper is not practical. They assume each party obtain different attributes of the same entities. But in real world, each party may contain different entities. And also each party need to have the same identity of entities. That means they need to share the identity of the entities. That may not common in the real world.
2. If it is a large dataset containing a great amount of entities, it will take a lot of time to find the closet cluster for each entity.

Application

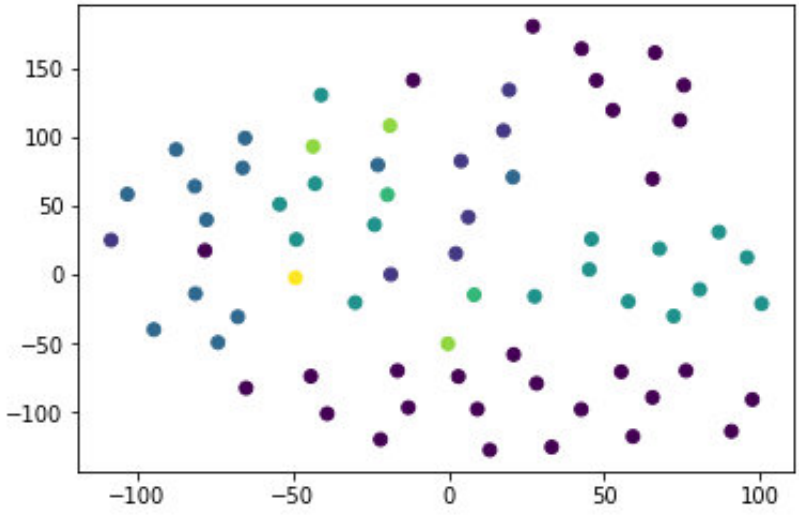
First, I applied the algorithm on a generated dataset. The dataset contains 20 attributes of 40 entities. I randomly assigned attributes to 5 parties. Because the data is high dimentional, so I use t-distributed Stochastic Neighbor Embedding to reduce the dimention. Here is the data visualization with their true class label.



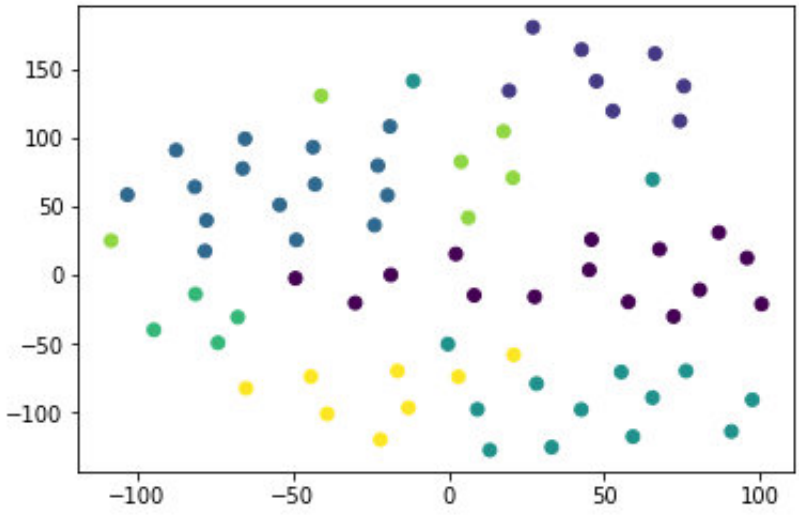
Then I used the privacy-preserving K-means clustering algorithm to the dataset. Here is the clustering result.



I applied the algorithm on the SCADI dataset. This dataset contains 206 attributes of 70 children with physical and motor disability based on ICF-CY. I randomly assigned the attributes to 10 parties. Here is the data visualization with their true class label.



Then I cluster the data using kmeans algorithm in the sklearn package. Here is the cluster results.



Finally, I applied the privacy-preserving K-means clustering algorithm to the dataset.