Privacy Preserving Naïve Bayes Classification Using Trusted Third Party Computation over Distributed Progressive Databases

Keshavamurthy B.N. and Durga Toshniwal

Department of Electronics & Computer Engineering,
Indian Institute of Technology Roorkee,
Uttarakhand, India
{kesavdec,durgafec}@iitr.ernet.in

Abstract. Privacy-preservation in distributed progressive databases is an active area of research in recent years. In a typical scenario, multiple parties may wish to collaborate to extract interesting global information such as class labels without revealing their respective data to each other. This may be particularly useful in applications such as customer retention, medical research etc. In the proposed work, we aim to develop a global classification model based on the Naïve Bayes classification scheme. The Naïve Bayes classification has been used because of its simplicity, high efficiency. For privacy-preservation of the data, the concept of trusted third party with two offsets has been used. The data is first anonymized at local party end and then the aggregation and global classification is done at the trusted third party. The proposed algorithms address various types of fragmentation schemes such as horizontal, vertical and arbitrary distribution required format. The car-evaluation dataset is used to test the effectiveness of proposed algorithms.

Keywords: privacy preservation, distributed database, progressive database.

1 Introduction

In recent years, due to the advancement of computing and storage technology, digital data can be easily collected. It is very difficult to analyze the entire data manually. Thus a lot of work is going on for mining and analyzing such data.

Of the various techniques of data mining analysis, progressive databases analysis is one of the active areas of research work. Progressive data mining discover the results in a defined period of interest or focus. The Progressive databases have posed new challenges because of the following inherent characteristics such as it should not only add new items to the period of interest but also removes the obsolete items from the period of interest. It is thus a great interest to find results that are up to date in progressive databases.

In many real world applications such as hospitals, retail-shops, design-firms and universities databases, data is distributed across different sources. The distributed database is comprised of horizontal, vertical or arbitrary fragments. In case of horizontal fragmentation, each site has the complete information on a distinct set of entities. An integrated

N. Meghanathan et al. (Eds.): CCSIT 2011, Part I, CCIS 131, pp. 24–32, 2011.

dataset consists of the union of these datasets. In case of vertical fragments each site has partial information on the same set of entities. An integrated dataset would be produced by joining the data from the sites. Arbitrary fragmentation is a hybrid of previous two.

Distributed progressive databases which constitute the characteristics of both database is distributed either in horizontal, vertical or arbitrarily as well as progressiveness of the database that is interested in, data within the focus or period of interest.

The key goal for privacy preserving data mining is to allow computation of aggregate statistics over an entire data set with out compromising the privacy of private data of the participating data sources. The key methods such as secure multiparty computation use some transformation on the data in order to perform the privacy preservation. One of the methods in distributed computing environment which uses the secure sum multi party computation technique of privacy preservation is Naïve Bayes classification [1].

A few of research papers have discussed the privacy preserving mining across distributed databases. One of important drawback with the existing methods of computation is that the global pattern computation is done at one of the data source itself [2]. This paper addresses this issue effectively by using a trusted third party and it also addresses the privacy preservation naïve bayes classification for Progressive distributed databases.

The rest of the paper is organized as follows: Section 2 briefs about related research work. Section 3 presents privacy preservation Naïve Bayes classification using trusted third party computation over distributed databases. Section 4 gives experimental results. Section 5 includes conclusion.

2 Related Work

Initially, for privacy preserving data mining randomization method were used, the randomization method has been traditionally used in the context of distorting data by probability distribution [3] [4]. In [5] [6], it was discussed how to use the approach for classification. In [5] discusses about privacy protection and knowledge preservation by using the method of ananymization, it anonymizes data by randomly breaking links among attribute values in records by data randomization.. In [6] it was discussed the building block to obtain random forests classification with enhanced prediction performance for classical homomorphic election model, for supporting multi-candidate elections.

A number of other techniques [7] [8] have also been proposed for privacy preservation which works on different classifiers such as in [7], it combine the two strategies of data transform and data hiding to propose a new randomization method, Randomized Response with Partial Hiding (RRPH), for distorting the original data. Then, an effective Naive Bayes classifier is presented to predict the class labels for unknown samples according to the distorted data by RRPH. In [8], Proposes optimal randomization schemes for privacy preserving density estimation.

The work in [9] [10] describes the methods of improving the effectiveness of classification such as in [9] proposes two algorithms BiBoost and MultBoost which allow two or more participants to construct a boosting classifier without explicitly sharing their data sets and analyze both the computational and the security aspects of the

algorithms. In [10] it proposes a method which eliminates the privacy breach and increase utility of the released database.

In case of distributed environment, the most widely used technique in privacy preservation mining is secure sum computation [11]. Here when there are n data sources $DS_0, DS_1, ...D_{n-1}$ such that each DS_i has a private data item $d_i, i = 0,1,...,n-1$ the parties want to compute $\sum_{i=0}^{n-1} d_i$ privately, without revealing their private data d_i to each other. The following method was presented:

We assume that $\sum_{i=0}^{n-1} d_i$ is in the range $[0, m-1]$ and DS_j is the protocol initiator:

1. At the beginning DS_j chooses a uniform random number r within [0, m - 1].
2. Then DS_j sends the sum $d_i + r \pmod m$ to the data source $DS_j + 1 \pmod n$.
3. Each remaining data sources DS_i do the following: upon receiving a value x the data source DS_i Sends the sum $d_i + x \pmod m$ to the data source $DS_i + 1 \pmod n$.
4. Finally, when party DS_j receives a value from the data source DS_{n-1} $-1 \pmod n$), it will be equal to the total sum $r + \sum_{i=0}^{n-1} d_i$. Since r is only known to DS_j it can find the sum $\sum_{i=0}^{n-1} d_i$ and distribute to other parties.

The Naïve Bayes technique applies to learning tasks where each instance x is described by a conjunction of attribute values and the target function $f(x)$ can take on any value from some finite set *C*. A set of training examples of the target function is provided, and a new instance is presented, described by the tuple of attribute values $a_1, a_2, ..., a_n$. The learner is asked to predict the target value, or classification, for this new instance. The Bayesian approach to classifying the new instance is to assign the most probable target value, c_{MAP}, given the attribute values $a_1, a_2, ..., a_n$ that describe the instance.

$$C_{MAP} = \arg_{\max} P(c_j / a_1, a_2, ...a_n) \ . \tag{1}$$

Using Bayes theorem,

$$C_{MAP} = \arg_{\max} P(a_1, a_2, ...a_n / c_j) P(c_j) / P(a_1, a_2, ...a_n) \ .$$

$$= \arg_{\max} (P(a_1, a_2, ...a_n / c_j) P(c_j)) \ . \tag{2}$$

The Naïve Bayes classifier makes the further simplifying assumption that the attribute values are conditionally independent given the target value. Therefore,

$$C_{NB} = \arg_{\max} (\mathrm{P}(\mathrm{c_j})) \bigcap \mathrm{P}(\mathrm{a_i}/\mathrm{c_j}) . \tag{3}$$

Where C_{NB} denotes the target value output by the Naïve Bayes classifier.

The conditional probabilities $P(a_i/c_j)$ need to be estimated from the training set. The prior probabilities $P(C_j)$ also need to decide by some heuristics. Probabilities are computed differently for nominal and numerical attributes.

For a nominal attribute in a horizontally partitioned data, the conditional probability $P(C = c/\ A = a)$ that an instance belongs to class c given that the instance has an attribute value $A = a$, is given by

$$\mathrm{P}(\mathrm{C}=\mathrm{c} \cap \mathrm{A}=\mathrm{a}) = \frac{\mathrm{P}(\mathrm{C}=\mathrm{c} \cap \mathrm{A}=\mathrm{a})}{\mathrm{P}(\mathrm{A}=\mathrm{a})} = \frac{\mathrm{n_{ac}}}{\mathrm{n_a}} . \tag{4}$$

n_{ac} is the number of instances in the (global) training set that have the class value *c* and an attribute value of *a*, while n_a is the number of instances in the global training set which simply have an attribute value of a. The necessary parameters are simply the counts of instances. n_{ac} and $\mathrm{n_a}$ Due to horizontal partitioning of data, each party has partial information about every attribute. Each party can locally compute the local count of instances. The global count is given by the sum of the local counts. Secure computing a global count is straightforward. Assuming that the total number of instances is public, the required probability can be computed by dividing the appropriate global sums where the local number of instances will not be revealed.

For an attribute with l different attribute values and a total of r distinct classes, l.r different counts need to be computed for each combination of attribute value and class value. For each attribute value a total instance count also to be computed, this gives l additional counts.

3 Progressive Naïve Bayes Classification Using Trusted Third Party Computation over Distributed Databases

3.1 Horizontal Partition

Here each party locally computes the local count of instances. The global count is given by the sum of the local counts. To count global values by summing all local values we use modified secure sum algorithm as shown in the Fig. 1 which is send to trusted third party.

Assumption: n parties, r class values, z attribute values , j^{th} attribute contain l_j, different values , S– supervisor, P- parties, $C^i_{l,r}$ = No. of instances with party P_i having classes r and attribute values l, N^i_r = No. of instances with party P_i having classes r.

At P:

```
Algorithm getData(POI)
{
While (new data is arriving on the site)
        For (each n parties having specific attributes)
                updateValues(c^i_yz, n^i_y);
Encrypt the values and send to trusted third party;
}
```

At S:

Receive values and decrypt them;

From all parties, we can get all possible $C^1_{l,r}$ and N^1_r

Parties calculate the required probabilities from $C^1_{l,r}$ and N^1_r, on that basis will predict the class.

Fig. 1. Algorithm for Horizontal Scenario

The Update Value function of horizontal scenario shown in Fig.2 will be described as follows: As in the progressive database, the new data is keep on arriving at every timestamp and data become obsolete to keep database up to data, so we also have to n_{ac} and n_a at every timestamp. In the algorithm, we keep on updating the n_{ac} and n_a until $t_{current}$ is less than POI, as no data become obsolete so we keep adding number of instances in both n_{ac} and n_a .

As $t_{current}$ exceeds POI then we also have to remove the instances which are no more in the required period of interest. At every timestamp, we store the records which will be obsolete in next time stamp, as these records have the values which have to be reduced to update n_{ac} and n_a. As the time increases new data will come and also we have the n_{ac} of obsolete data. We update the n_{ac} and n_a by adding the values of n_{ac} and n_a of new data, and by removing values of n_{ac} and n_a by the obsolete data as shown in updateValue function.

```
void UpdateValues (List c^i_yz, List n^i_y)
{
For transaction at t = t_current
{
    For all class values y do
    For all z, Party P_i locally computes c^i_yz
    Party P_i locally computes n^i_y
    End For
}
c^i_yz (new) = c^i_yz (previous) + c^i_yz (transaction at t = t_current)
n^i_y (new) = n^i_y (previous) + n^i_y (transaction at t = t_current)
If ((t_current -POI)>0)
{   For transaction at t = (t_current -POI)
        {  For all class values y do
         For all z, Party P_i locally computes c^i_yz
         Party Pi locally computes n^i_y
         End For   }
   c^i_yz (new) = c^i_yz (current) - c^i_yz (transaction at t = t_current - POI)
   n^i_y (new) = n^i_y (current) - n^i_y (transaction at t = t_current)
  }   }
```

Fig. 2. Update value function

3.2 Vertical Partition

In nominal attributes, each party calculates their local instances of n_{ac} and n_a, of the attribute values they have. As each party have different attributes, so no parties have same value of instance of attribute and class. Hence there is no need to calculate the sum of values. At a particulate timestamp, we calculate the local values of n_{ac} and n_a, and send them to the trusted third party The necessary algorithm is given above in Fig.3.

Assumption: n parties, r class values, z attribute values , j^{th} attribute contain l_j, different values , S– supervisor, P- parties, $C^i_{l,r}$ = No. of instances with party P_i having classes r and attribute values l, N^i_r = No. of instances with party P_i having classes r.

At P:

```
Algorithm getData(POI)
{
While (new data is arriving on the site)
        For (each n parties having specific attributes)
                updateValues(c^i_yz, n^i_y);
Encrypt the values and send to trusted third party;
}
```

At S:

Receive values and decrypt them;

From all parties, we can get all possible $C^1_{l,r}$ and N^1_r

Parties calculate the required probabilities from $C^1_{l,r}$ and N^1_r, on that basis will predict the class.

Fig. 3. Algorithm for vertical fragmentation

3.3 Arbitrary Partition

In this fragmentation scenario, we consider the mix of both horizontal and vertical fragmentation. Some of the parties have their database distributed horizontally and some having the horizontal part distributed vertically. The trusted third party knows about the partition of different parties with all attributes and class values. The method proposed here uses the ideas discussed in the previous two sections. In this case, we apply our new modified scheme of secure multiparty computation.

4 Results

The dataset used for the purpose of experimentation is car-evaluation [12]. The analysis results of different partitions of distributed progressive databases at a specific period of interest are given as follows and the classification accuracy is in percentage.

Table 1. Horizontal partition classification analysis

Sl. No.	Description	Number of parties	Total number of records	%Accuracy
1	Classification of data at single party (No distribution)	1	1728	85
2	Classification of data Distributed in horizontal partition with trusted third party	3	Party1: 500 Records Party2: 500 Records Party3: 728 Records	85

Table 2. Vertical partition classification analysis

Sl. No.	Description	Attributes per parties	Total number of records	%Accuracy
1	Classification of data at single party (No distribution)	7	1728	85
2	Classification of data distributed in vertical partition with trusted third party	Party1: 2 attributes Party2: 2 attributes Party3: 3 attributes	1728	85

Table 3. Arbitrary partition classification analysis

Sl. No.	Description	Total number of parties	Total number of records/attributes per parties	Total number of records	% Accuracy
1	Classification of data Distributed in horizontal partition with trusted third party	2	Party1: 500 records Party2: 500 records	1000	85
2	Classification of data distributed in vertical partition with trusted third party	3	Party1: 2 attributes Party2: 3 attributes Party3: 2 attributes	728	85
3	Classification of data distributed in arbitrary partition with trusted third party (combination of horizontal and vertical partition)	5	7 attributes	1728	85

5 Conclusion

In our proposed work, we have proposed a set of algorithms for classifying data using Naïve Bayes from a group of collaborative parties without breaching privacy. The Naïve Bayes approach is used because of its simplicity and high efficiency. The non-distribution and various distribution scenarios such as horizontal, vertical and arbitrary scenarios are compared and their accuracy is calculated on the data. The accuracy comes out to be the same showing that the algorithm is giving best case results. Privacy is also preserved using privacy preservation techniques such as offset computation and encryption. The third party concept is also introduced to calculate global classification results with privacy preservation. In this case, Naïve Bayes algorithm is applied to the different distributed progressive sequential data streams scenarios such as horizontal, vertical and arbitrary. In future algorithm can be modified for numeric data to widen its scope.

References

1. Vaidya, J., Kantarcıoğlu, M., Clifton, C.: Privacy-Preserving Naïve Bayes classification. International Journal on Very Large Data Bases 17(4), 879–898 (2008)
2. Huang, J.-W., Tseng, C.-Y.: A General Model for Sequential Pattern Mining with a Progressive Databases. IEEE Trans. Knowledge Engineering 20(9), 1153–1167 (2008)
3. Liew, C.K., Choi, U.J., Liew, C.J.: A data distortion by probability distribution. ACM TODS, 395–411 (1985)
4. Warner, S.L.: Randomized Response: A survey technique for eliminating evasive answer bias. Journal of American Statistical Association (60), 63–69 (1965)
5. Agarwal, R., Srikanth, R.: Privacy–preserving data mining. In: Proceedings of the ACM SIGMOD conference, vol. 29, pp. 439–450 (2005)
6. Agarwal, D., Agarwal, C.C.: On the design and Quantification of Privacy–Preserving Data Mining Algorithms. In: ACM PODS Conference, pp. 1224–1236 (2002)
7. Zhang, P., Tong, Y., Tang, D.: Privacy–Preserving Naïve Bayes Classifier. In: Li, X., Wang, S., Dong, Z.Y. (eds.) ADMA 2005. LNCS (LNAI), vol. 3584, pp. 744–752. Springer, Heidelberg (2005)
8. Zhu, Y., Liu, L.: Optimal Randomization for Privacy–Preserving Data Mining. In: KDD ACM KDD Conference (2004)
9. Gambs, S., Kegl, B., Aimeur, E.: Privacy–Preserving Boosting. Journal (to appear)
10. Poovammal, E., Poonavaikko, M.: An Improved Method for Privacy Preserving Data Mining. In: IEEE IACC Conference, Patiala, India, pp. 1453–1458 (2009)
11. Yao, A.C.: Protocol for secure sum computations. In: Proc. IEEE Foundations of Computer Science, pp. 160–164 (1982)
12. Bohanec, M., Zupan, B.: UCI Machine Learning Repository. (1997), http://archive.ics.uci.edu/ml/datasets/Car+Evaluation