Research Journal of Applied Sciences, Engineering and Technology 4(22): 4865-4870, 2012
ISSN: 2040-7467

Submitted: May 13, 2012 Accepted: June 08, 2012 Published: November 15, 2012

A Novel K-NN Classification Algorithm for Privacy Preserving in Cloud Computing

Jian Wang
College of Computer and Information Engineering, Henan University of Economics and Law, Zhengzhou, China

Abstract: We can only enjoy the full benefits of Cloud computing if we can address the privacy disclosure problem about individual or company. As large amounts of data which are stored in the cloud contain private information and are in non-aggregate format, sharing the data with third-party service providers in the cloud without strict restriction will cause great danger to data privacy. Although increased awareness of the privacy issues in the cloud, little work has been done in privacy-preserving data classification based cloud environment. To solve this problem, we explore a novel algorithm named PPKC (Privacy Preserving K-NN Classification) for mining data in the cloud without the disclosure of sensitive information.

Keywords: Classification, cloud computing, privacy matching, privacy preserving

INTRODUCTION

With the development of cloud computing, more and more companies, such as Google, HP, Amazon, begin to use this technology to provide service to users. However, the cloud vision does offer some particularly challenging privacy problems that are unlikely to be sufficiently addressed by today's best practice (Jian *et al.*, 2009). Cloud computing raises a range of important privacy issues as acknowledged by a number of recent research work (Siani *et al.*, 2009). Such issues are due to the fact that the input data for cloud services is uploaded by the user to the cloud, which means that they typically result in users' data being present in unencrypted form on a machine that the user does not own or control. This poses some inherent privacy challenges (Jian *et al.*, 2009). As large amounts of data which are stored in the cloud contain private information and are in non-aggregate format, sharing the data with third-party service providers in the cloud without strict restriction will cause great threat to data privacy.

The problem of privacy-preserving data mining has found considerable attention in recent years because of recent concerns on the privacy of underlying data. In recent years, the issue of privacy protection in classification has been raised (Zhang *et al.*, 2005). The objective of privacy-preserving data classification is to build accurate classifiers without disclosing private information in the data being mined. Although increased awareness of the privacy issues in the cloud, little work has been done in privacy-preserving data classification based cloud environment. Therefore our research provides a novel method to classification mining data in the cloud with privacy preserving manner.

METHODOLOGY

Binary Weighted Cosine (BWC) metric to measure similarity: Rawat *et al.* (2006) proposed BWC similarity measure for measuring similarity across sequences of system calls. They showed the effectiveness of the proposed measure on IDS. They applied k-nn classification algorithm with BWC metric measure to enhance the capability of the classifier. BWC similarity measure considers both the number of shared elements between two sets as well as frequencies of those elements in traces. The similarity measure between two sequences A and B is given by:

$$S(A,B) = \frac{A \bullet B}{\|A\|\|B\|} * \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

Assume A and B are sets containing elements. The Algorithm 1 shows the steps required to calculate the BWC similarity function.

Algorithm 1: BWC similarity function: S (A, B)

int $S_{intersection} = 0$; $S_{union} = 0$;
for all $a \in A$ and $b \in B$ {
  if (a = b) {
    $S_{intersection} = S_{intersection} + 1$;
    $S_{union} = S_{union} + 1$;
  }
  else $S_{union} = S_{union} + 2$; }

BWC Notation S (A, B);

$$S(A,B) = \frac{S_{intersection}}{S_{union}};$$

Binary Weighted Cosine (BWC) Metric similarity measure is derived from Cosine similarity as well as Jaccard similarity measure. Since the Cosine similarity measure is a contributing component in a BWC similarity measure hence, BWC similarity measure is also a vector based similarity measure. The transformation step is same as carried out in Cosine similarity measure or Euclidean measure for sets. For two sets, A = {p, s, t, n, q, k, r, m} and B = {k, m, t, k, m, q, m}, the computed BWC similarity measure comes out to be 0.45.

Private matching protocol: The purpose of private matching is to find common data elements (or perform a join) over two datasets without revealing private information (Freedman *et al.*, 2005). Private matching schemes attempt to enable parties to participate in such tasks without worrying that information is leaked (Yaping Li *et al.*, 2005).

A Private Matching (PM) scheme is a two-party protocol between a Client (C) and a Server (S). C's input is a set of inputs of size k_c, drawn from some domain of size N; S's input is a set of size k_s drawn from the same domain. At the conclusion of the protocol, client C learns which specific inputs are shared by both C and S. That is, if C inputs $X = \{x_1, ...x_{k_c}\}$, S inputs $Y = \{y_1, ...y_{k_s}\}$ and C learns $X \cap Y : \{x_u \mid \exists v, x_u = y_u\} \leftarrow PM(X, Y)$.

Algorithm 2: Private matching protocol (PMP (a, b))

(1) Both Record A (named A, for short) and Record B (named B, for short) apply hash function h to their sets.
Xa = h (Va) Xb = h (Vb)
Each party randomly chooses a secret key. And ea is the key for A, eb is the key for B.

(2) Both parties encrypt their hashed sets:
Ya = Fea (Xa) = Fea (h (Va)) Yb = Feb (Xb) = Feb (h (Vb))

(3) A sends to B its encrypted set Ya = Fea (h (Va))

(4) (a) B ships to A its set Yb = Feb (h (Vb))
(b) B encrypts each y∈Ya, with B' key eb and sends back to A the pairs <y, Feb (y)>=<Fea (h (v)), Feb (Fea (h (v)))>

(5) A encrypts each y∈Yb, with A's key ea, obtaining Zb = Fea (y) = Fea (Feb (h (v))), here the v∈Vb.
Also, from pairs <Fea (h (v)), Feb (Fea (h (v)))> obtained in Step 4 (b) for the v∈Va, It creates pairs <v, Feb (Fea (h (v)))> by replacing Fea (h (v)) with the corresponding v.

(6) For each v∈Va, if this element v∈Va meets (Feb (Fea (h (v)))) ∈Zb, then this v∈Vb, thus return True. Else return False.

The main purpose for our research to use private matching technology is to intersect the record of database from one node and another record from other node' s database in the cloud without accessing each record's data, so this can check whether these two compared records are same, meanwhile avoiding the disclosure of each record's data. At last private matching protocol will reply with true if both the records are same else it will reply with false.

We design private matching protocol to ensure BWC similarity measure work well. In the following section we detailed address private matching protocol in privacy preserving manner. The algorithm for private matching protocol is described in algorithm 2.

K-NN classification algorithm for cloud data: The following is the algorithm to compute BWC similarity value in privacy preserving manner.

Algorithm 3: Privacy Preserving BWC Similarity Function

int $S_{intersection} = 0$; $S_{union} = 0$;
for all $a \in A$ and $b \in B$ {
if (PMP (a, b) = True) {
$S_{intersection} = S_{intersection} + 1$;
$S_{union} = S_{union} + 1$; }
else $S_{union} = S_{union} + 2$; }
BWC Notation S (A, B);

$$S(A,B) = \frac{S_{intersection}}{S_{union}};$$

Privacy-preserving data mining becomes an important and practical technology for mining data from multiple private databases owned by different and possibly competing organizations. For example, many insurance companies collect data on disease incidents, seriousness of the disease and patient background. One way for the Center for Disease Control to identify disease outbreaks is to train a classifier across the data held by the various insurance companies for patterns that are suggestive of disease emergence and use it to classify a query pattern as an emergence or the opposite.

However, commercial and legal reasons prevent the insurance companies from revealing their data. It is necessary and beneficial to use a distributed data mining algorithm that is capable of identifying potential disease emergence while protecting the private information of its participants.

There are three aspects that we should consider when designing a privacy preserving classification algorithm, namely, accuracy, efficiency and privacy. Ideally, we would like the algorithm to have a comparable accuracy to its non-privacy preserving counterpart and an absolute privacy wherein no information other than the trained classifier and the classification of the query instance should be revealed to any node. At one end of the spectrum, we have the non-privacy preserving classifier algorithms, which are highly efficient but are not secure. At the other end, we have the secure multi-party computation protocols, using which we can construct classifiers which are provably secure in the sense that they reveal the least amount of information and have the highest accuracy; but are very inefficient. Our design goal is to look for algorithms that can provide a desired level of tradeoff between the accuracy of the classifier constructed and the stringency of the privacy requirements while maintaining efficiency.

To solve the k-nn classification problem, we need to adapt the basic distance weighted k-nn classification algorithm to work in a distributed setting in a privacy preserving manner.

In a distributed setting, the k nearest neighbors of one node could be distributed among the n nodes. Thus, each node will have some records in its database which are among the k nearest neighbors of this comparison. So, for a node to calculate its local classification, it has to first determine which records in its database are among the k nearest neighbors of this comparison. Then, it can calculate its classification of the comparison. Finally, the nodes need to combine their local classifications, in a privacy preserving manner, to compute the global classification of the comparison over the n private databases from different nodes. Thus we can divide the k-nn classification problem into the following two sub-problems: Local Classification and k-nn Classification.

Algorithm 4: Local Classification

Consider the records in database a as the training dataset.
Consider the records b as the test record from database B.
Calculate its BWC similarity with respect to all the records of training set through the Algorithm 3.
Pick the nearest k records from the training set.
Get the records matching the most with the testing record.
Consider these records as the Local Common set for A.
Return a vector containing the pair
(class label, similarity value) of each record for all the k records to the Broker.

There are many service providers in the cloud, we can call each service as a node, each node will exchange data with other nodes. So the data for a specific application is stored in the distributed nodes. As described in Algorithm 4, Local Classification will use BWC similarity measure function for measuring similarity of the compared records and will return pair (class label, similarity value) of all k records to the Broker. Algorithm 5 presents our approach to classify cloud data using weighted k-NN classifications while preserving privacy.

In Algorithm 5, the Broker will consider one record of a selected database as a test record. It sends this test record to all the nodes in the cloud. A pair (class label, similarity values) of k neighbour records is gathered in a set called Local Common set and they are sent to the classifier. As a result the classifier will get k (n-1) number of entries. Now the classifier will consolidate such lists from all the nodes. It will merge them into a single Global common list. From this list it will pick the k most similar nodes and assign the test record with the class name using the weighted k-NN approach.

Algorithm 5: PPKC

Parties involved: Let the P1, P2,...Pn be the participating training nodes with databases DB1, DB2,...DBn, where n is the number of parties. Assume a semi-honest third party calles Broker will pick the k nearest neighbours at global level.

Local phase: Consider the node P1 as the testing node
for all records in the testing node database
{
  for all training nodes {
  Compute k local neighbour using the
  Algorithm 4.
Send pairs (class label, similarity value)
  of k local neighbour records to the Broker.
  }
}
Global phase:
Merge all the pairs for a training record as
a single list called the global list.
Broker sends global list to classifier.
Classifier selects k most similar records based
on the similarity measure at the global level.

Assign the class label for the test record based on the weights (similarity value) of the k neighbors.

SECURITY ANALYSIS

Definition 1: (ε-Differential Privacy (Ilya *et al.*, 2009). A randomized algorithm P is ε-Differential Privacy if for all transaction data sets $T, T' \in D^n$ differing in at most one transaction and all events $E \subseteq Range(A)$: $\Pr[P(T) \in E] \le e^{\varepsilon} \Pr[P(T') \in E]$, where D^n denote the space of transaction data sets containing n transactions.

The ε-Differential Privacy notion limits the probability that the randomized function would leak information from a database that is differing by at most one element. Therefore a data leak and a disclosure of a private data through this function is possible, but the probability of such leak is limited through the leakage parameter ε.

Theorem 1: PPKC algorithm meets ε-Differential Privacy.

Proof: Let D^n be the domain of data sets of n transactions.

Let $S_T = \left\{ \left\langle L_1, \bar{f}_T(L_1) \right\rangle, ..., \left\langle L_K, \bar{f}_T(L_K) \right\rangle \right\}$ represent the output of the algorithm PPKC running on data set $T, T' \in D^n$. Let L_i represents the itemsets and $\bar{f}_T(L_i)$ represent the noisy frequencies. To denote the intermediate noisy frequency, we use $\check{f}_T(L)$.

Now if we want to prove that PPKC algorithm meets ε-
Differential Privacy, we only need to prove:

$$\Pr[PPKC(T) = W] \le e^{\varepsilon} \Pr[PPKC(T') = W] \quad (2)$$

Because

$$\Pr[PPKC(T) = W] = \int_{v_1 \in R} ... \int_{v_K \in R} pdf_T\left[\check{f}_{L_1} = v_1\right] pdf_T\left[\check{f}_{L_K} = v_K\right] \prod_{L \in 2^U - W, |L| = l} \Pr_T\left[\check{f}_L < \min\{v_1, v_2, ..., v_k\}\right] \quad (3)$$

We use the notation $pdf_T[\ ]$ and $\Pr_T[\ ]$ to parameterize the probability density function and the probability mass function. In order to decrease $\Pr[PPKC(T') = W]$, we can either increase or decrease

$\hat{f}_L(T)$ by $\frac{1}{n}$, to obtain $\hat{f}_L(T')$, let $\hat{f}_L(T') - \hat{f}_L(T) = \frac{1}{n}$

where,

$L \in 2^U - W, |L| = l$

Because for any $L \in 2^U$ and for any $v \in R$, we can get:

$$pdf_T\left[\check{f}_L = v\right] = \frac{1}{2\lambda} e^{-\frac{\left|v - \hat{f}_L(T)\right|}{\lambda}} \quad (4)$$

Similarly, when $v < \hat{f}_L(T)$:

$$\Pr_T\left[\check{f}_L < v\right] = \frac{1}{2} e^{-\frac{\left|v - \hat{f}_L(T)\right|}{\lambda}} \quad (5)$$

When $v \ge \hat{f}_L(T)$:

$$\Pr_T\left[\check{f}_L < v\right] = 1 - \frac{1}{2} e^{-\frac{\left|v - \hat{f}_L(T)\right|}{\lambda}} \quad (6)$$

Note that $\Pr_T\left[\check{f}_L < v\right]$ decreases when $\check{f}_T$ increases:

For an itemset $L \in W$, $\frac{pdf_T\left[\check{f}_L = v\right]}{pdf_{T'}\left[\check{f}_L = v\right]}$ is at most $e^{\frac{2}{n\lambda}}$ (since we are changing $\hat{f}_L$ by at most 2/n). Since, in each term in the integration of the expression $\Pr[PPKC(T') = W]$, there are exactly K terms which has $L \in W$. Therefore, when we change from to each term in the integration changes by at most $e^{\frac{2K}{n\lambda}}$.

Therefore, $\frac{\Pr[PPKC(T) = W]}{\Pr[PPKC(T') = W]}$ is upper bounded by $e^{\frac{2K}{n\lambda}}$.

Hence, if we set $\lambda = \frac{2K}{n\varepsilon}$, then we can guarantees $\frac{\Pr[PPKC(T) = W]}{\Pr[PPKC(T') = W]} \le e^{\frac{2K}{n\lambda}} = e^{\varepsilon}$. Therefore, we can

conclude PPKC algorithm meets ε-Differential Privacy.

EXPERIMENTAL EVALUATION

In this experiment we compare our proposed two protocols and their respective non-privacy preserving versions in the aspect of accuracy loss. We have used Java language to implement our protocols. The experiments are carried out on Windows XP operating system with 2.13 GHz Intel Core i3 processor and 4 GB of memory. The experiments are executed on the real world data from UCI dataset repository. Table 1 shows datasets and training parameters. Training architecture contains the number of input nodes, hidden nodes and output nodes (Ilya *et al.*, 2009). We have used Iris, Dermatalogy, Sonar and Landsat datasets. We suppose there are two participants in our experiment.

For each dataset, the test samples for the experiments are taken randomly from the datasets. Specifically, 20 test samples are taken randomly each for Iris and Sonar and 30 each for Dermatalogy and Landsat. The number of training rounds is kept small for large datasets like Landsat and large for other datasets. After training, each test sample is run against the network to observe whether it is misclassified or it belongs to the same class.

To protect the private data of each party and intermediate computation result from disclosing, we apply cryptographic schemes on the non-privacy preserving version of the protocols. Since the requirement of cryptographic operations, accuracy loss takes place. The accuracy loss for each dataset can be computed by the following equation.

Accuracy loss = T1-T2, where T1 is the test error rate for privacy version of the PPKC algorithm and T2 is the test error rate for Non-privacy version. No matter for privacy version or Non-privacy version of this algorithm, test error rate can be computed by the following formula.

> Test error rate = (Number of misclassified test samples) / (Total number of training samples)

When data is horizontally partitioned, experimental results of the difference of the test error rate between PPKC algorithm and his non-privacy preserving version are shown in Table 2.

When the data is vertically partitioned, experimental results of the difference of the test error rate between PPKC algorithm and his non-privacy preserving version are shown in Table 3.

By observing the experimental results in Table 2 and 3, we found that regardless of data horizontally partitioned or vertically partitioned, the proposed two protocols' test error rates are higher than respective non-privacy preserving versions. The reason is the introduction of cryptographic operations to protect the private data. After using the cryptographic operations, the appearance of accuracy loss is inevitable. In the case of data horizontally partitioned, the accuracy loss of these four datasets are 3.51, 6.26, 5.09 and 2.98%, respectively. In the case of data vertically partitioned, the accuracy loss of these four datasets are 5.42, 5.77, 7.10 and 2.05%, respectively. Since the accuracy loss is within certain limits, our proposed two protocols are still very effective in learning these real world datasets.

Table 1: Datasets and training parameters

Dataset	Sample	Class	Training architecture	Number of training rounds
Iris	134	4	4-5-3	60
Dermatalogy	399	7	30-3-6	20
Sonar	219	2	60-6-2	70
Landsat	6786	5	36-3-6	30

Table 2: Test error rates comparison when data horizontally partitioned

Dataset	Non-privacy preserving version (%)	PPKC (%)
Iris	10.90	14.41
Dermatalogy	18.14	24.40
Sonar	13.48	18.57
Landsat	11.32	14.30

Table 3: Test error rates comparison when data vertically partitioned

Dataset	Non-privacy preserving version (%)	PPKC (%)
Iris	15.09	20.51
Dermatalogy	20.16	25.93
Sonar	17.78	24.88
Landsat	14.99	17.04

CONCLUSION

In this study, we introduced a novel approach toward addressing the newly emerging and important issue how to mine the data in cloud with privacy preserving manner. We proposed a new approach, which aims at ensuring the task of classification mining data in the cloud, meanwhile avoiding the disclosure of the private information of users'. We design a new private matching protocol to check whether two values from different records are similar, then using BWC similarity measure function for measuring similarity of these compared records. At last the outcome of BWC similarity measure will be used in the weighted k-nn classification based cloud environment. In the future we will consider other data mining technology such as clustering mining or association rule mining to mine data in the cloud with privacy preserving manner.

ACKNOWLEDGMENT

This research is supported by the National Natural Science Foundation of China under the grant No.

60773100 and by the Education Science and Technology Project of Henan Province under the grant No. 12A520005.

REFERENCES

Freedman, M.J., K. Nissim and B. Pinkas, 2004. Efficient private matching and set intersection. Proceedings International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology-EUROCRYPT 2004, Interlaken, Switzerland, May 2-6, 2004, pp: 1-19.

Ilya, M., P. Omkant, R. Omer and V. Salil, 2009. Computational differential privacy. J. Adv. Cryptol., 5677: 126-142.

Jian, W., Z. Yan, J. Shuo and L. Jiajin, 2009a. Providing privacy preserving in cloud computing. International Conference on Test and Measurement, 5-6 Dec., Coll. of Inf. Sci. and Technol., Donghua Univ., Shanghai, China, 2: 213-216.

Jian, W., L. Yongcheng, J. Shuo and L. Jiajin, 2009b. A survey on anonymity-based privacy preserving. International Conference on E-Business and Information System Security, 23-24 May, Coll. of Inf. Sci. Technol., Donghua Univ., Shanghai, pp: 1-4.

Rawat, S., V.P. Gulati, A.K. Pujari and V.R. Vemuri, Intrusion detection using text processing techniques with a binary-weighted cosine metric. J. Info. Assur. Secur., 1: 43-50.

Siani, P., S. Yun and M. Miranda, 2009. A privacy manager for cloud computing. Cloud Comput., 5931: 90-106.

Yaping, L., J. Tygar and H. Joseph, 2005. Private matching. Comput. Sec. 21st Cent., 1: 25-50.

Zhang, N., S. Wang and W. Zhao, 2005. A new scheme on privacy-preserving data classification. Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, (KDD '05), ACM New York, USA, pp: 374-383.