Privacy Preserving Naïve Bayesian Classifier For Horizontal Partitioned Data

Sumana M.
M S Ramaiah Institute of Technology/Information Science and Engineering, Bangalore, India
Email: sumana.a.a@gmail.com

Hareesha K. S.
Manipal Institute of Technology/Department of Computer Applications, Manipal,India
Email: harish_dvg@yahoo.com

***Abstract*—Data is distributed in various sites that need to be mined in a secure manner without revealing anything except the results of mining. This paper converses about privacy-preserving horizontal distributed classification techniques where multiple sites collaborate and broadcast the mining results. However in the process, no information about either the data maintained in the sites or data obtained during computation is divulged. We have presented two protocols to construct a Privacy Preserving Naïve Bayesian classifier using the Pailler's homomorphic encryption techniques.**

We propose that our approach is more secure and efficient than any of the previous privacy preserving Naïve Bayesian methods.

***Index Terms*—secure sum, homomorphic encryption, paillier encryption, privacy preserving data mining, naïve Bayesian, horizontal partitioned**

I. Introduction

Similar organizations at various locations want to obtain patterns, data by jointly collaborating with each other. The purpose of Privacy-preserving data mining is to mine distributed datasets without revealing information while mining. The data mining techniques performed in a private way can be grouped into 3 categories: Randomization, Group-based anonymization and Distributed Privacy Preserving Data mining that uses cryptographical approaches to mine data [1]. The type of mining that we want to perform involves parties having data with similar features, horizontally distributed and inquisitive to obtain relevant information.

With finance datasets multiple sites need to predict low, medium or high risk in providing loan to an individual, given details regarding the income, family dependents, money invested in stocks, type of occupation and education. In the process of prediction each of the sites involved in modeling do not want to reveal their data that they enclose. Similarly various hospitals want to conclude whether a person has cyst is benign or malignant without revealing the patients details, according to the privacy rules in HIPAA, [2] National standards to protect the privacy of personal health information. Various hospitals want to jointly predict the course of treatment for a disease is suitable or risky. However these medical centers cannot disclose the patient data nor the treatment plans legally. These data mining tasks can be performed by training the datasets available at multiple sites in cooperation and then grant the decision for the data to be classified. As observed the kind of data collected by the parties are the same and hence the scenario requires building classification models and predicting values on horizontally partitioned dataset.

Early learner classifiers such as decision tree, Naïve Bayesian, Neural Networks are some of the common methods used for classification. But it has been observed [3] that Naïve Bayesian approach generates proficient results when the attributes of the dataset are categorical and numeric in nature. While multiple sites collaborate to build a model none of the sites should learn anything except the outcome. Maintaining this concept of privacy in this research we have developed protocols to train the multiple datasets and classify new entries.

Our contributions are as follows. First, we investigated the problem of training datasets using the Naïve Bayesian classifier mentioned in [3]. We propose two improved secure approaches of Naïve Bayesian classifier in section II.

A. Related Work

The concept of privacy preserving data mining was proposed by two different papers [1] and [4]. [1], a solution was presented by adding noise to the source data by Agrawal. [4] proposed a decision tree classifier using cryptographic tools. Early learner classifiers such as decision tree, Naïve Bayesian, Neural Networks are some of the common methods used for classification using cryptography. [4] and [5] shows how ID3 decision trees on horizontally partitioned data can be constructed. [3], [6] discusses building of the naïve Bayesian classifier on horizontal and vertical datasets. Existing approaches use Randomization-based [1] or cryptography-based approaches. As told in [7], [8] cryptography approaches guarantee on privacy compared to randomization approach.

Manuscript received November 1, 2013; revised January 11, 2014.

doi: 10.12720/ijeee.2.1.21-25

B. Naïve Bayesian Classification

Naïve Bayesian Classifier [9] uses the Bayes Theorem to train the instances in a dataset and classify new instances to the most probable target value. Each instance is identified by its attribute set and a class variable. Given a new instance X with an attribute set, the posterior probability P(Class1/X),P(Class2/X) etc has to be computed for each of the class variable values based on the information available in the training data. If P (Class1/X)>=P (Class2/X)>=……>=P (ClassN/X) for N class values, then the new instance is classified to Class1or Class2…or ClassN accordingly.

This classifier estimates the class-conditional probability by assuming that the attributes are conditionally independent, given the class label y. The conditional independence can be obtained as follows:

$P(X|Y=y) = \prod_{i=1}^{d} P(X_i|Y=y)$, where each attribute set $X = \{X_1, X_2, \ldots, X_d\}$ consists of d attributes.

Each of the d attributes can be categorical or numeric in nature.

Algorithm 1 indicates the computation of the probability for a categorical attribute and **Algorithm 2** indicates the computation of mean, variance and standard deviation required for calculating probability.

Algorithm 1 : Handling a categorical attribute

Input: r -> # of class values, p -> #of attribute values

C_{xy} –> represents #of instances having class x and attribute value y.

N_x – > represents # of instances that belong to class x

Output: P_{xy} –> represents the probability of an instance having class x and attribute value y

For all class values y do

{Compute N_x

For every attribute value x

{Compute C_{xy}

Calculate $P_{xy} = C_{xy}/ N_x$}}

Algorithm 2 : Handling numeric attribute

Input: r -> # of class values, x_{jy} -> value of instance j having class value y.

S_y -> represents the sum of instances having class value y

N_y -> represents # of instances having class value y

For all class values y do

{Compute $S_y = \sum_j x_{jy}$

Compute n_y

Compute $Mean_y = S_y/ n_y$

Compute $V_{jy} = (x_{jy} - Mean_y)^2$ for every instance j that belongs to the class y

Compute $Var_j = \sum_j V_{jy}$

Compute $Stan\_dev^2_y = Var_j / (N_y-1)$

}

Once the Variance and Standard Deviation is computed the probability for the numeric value provided in the test record for each of the class can be computed as follows:

P (given that (attribute_value = test_record_numeric_value)| $Class_y$)

$$= 1\frac{1}{\sqrt{2\pi} * Stan\_dev}\exp^{-\frac{(\text{test\_record\_numeric\_value} - Mean)_y}{2\ X\ Stan\_dev\ (of\ class\ y)}}$$

On obtaining the Probabilities for each of the attributes with respect to each of the classes the class-conditional probabilities can be computed as follows:

For each of the class value I

Probability (test record having z attribute values | classI)= P(Attr1_value|classI) *P(Attr2_value|classI) *…….* P(Attrz_value|classI)

The test record belongs to the class has the maximum class-conditional probability.

C. Paillier Encryption

This asymmetric public key cryptography[10] approach of encryption is largely used in privacy preserving data mining methods. The scheme is an additive homomorphic cryptosystem that are used in algorithms where secure computations need to be performed.

Key generation

Obtain two large prime numbers p and q randomly selected big integers and independent of each other such that gcd(pq,(p-1)(q-1)) = 1. Compute n = pq and

$$\lambda = lcm(p-1, q-1)$$

Select random integer g where $g \in Z^*_{n^2}$. Check whether n divides the order of g as follows

Obtain $\ell = ((p-1)*(q-1))/gcd(p-1,q-1)$

If(gcd(((g^ℓ mod n^2)-1)/n),n)!=1) then select g once again.

Encryption

Encrypts the plaintext m to obtain the Cipher text $c = g^m * r^n \bmod n^2$.

where m plaintext is a BigInteger and ciphertext is also a BigInteger

Decryption

Decrypts ciphertext c to obtain plaintext $m = L(g^\ell \bmod n^2) * u \bmod n$, where $u = (L(g^\ell \bmod n^2))^{-1} \bmod n$.

D. Homomorphic Encryption

Homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on ciphertext and obtain an encrypted result which decrypted matches the result of operations performed on the plaintext. For instance, one person could add two encrypted numbers and then another person could decrypt the result, without either of them being able to find the value of the individual numbers.

Encryption techniques such as ElGamal [11] and Paillier [10] have the homomorphic property i.e for messages m1and m2 E(m1+m2) = E(m1)+E(m2) without decrypting any of the two encrypted messages.

Also $D(E(m1)*E(m2) \bmod n^2) = m1 + m2 \bmod n$.

D indicate decryption and E indicates Encryption.

E. Probabilistic Property

ElGamal and Paillier schemes are also probabilistic [12], which means beside the plain texts, encryption operation needs a random number as input. Under this property there can be many encryptions for each message. Therefore no individual party can decrypt any message by itself.

II. Improved Privacy Preserving Naïve Bayes

In this section, we focus on securely constructing a Naïve Bayesian Model on horizontally classified data.

Customer information maintained by different banks, patient information maintained at various hospitals can be seen as an example of horizontally partitioned data. Each of the banks hold information about their customers and different banks have different customers.

A protocol is presented in [3] as to see how a privacy-preserving Naïve Bayesian classifier is constructed, but as mentioned in [3], security is compromised.

A. Naïve Bayes Using Homomorphic Encryption

To make our algorithm more secure we need to compute the sum more securely rather than using just a random number as in [3]. In this algorithm we have used homomorphic encryption technique for computing the secure sum. Homomorphic encryption is performed using paillier encryption since this technique is also probabilistic asymmetric algorithm.

1) Handling a categorical attribute

Requirements: k parties, r class values, x attribute values

C^x_{yz} – represents # of instances with party P_x having class y and attribute value z.

n^x_y – represents # of instances with party P_x having class y.

p_{yz} - represents the probability of an instance having class y and attribute value z.

for all class value y do
  for i= 1 to k do
   for every attribute value z, party P_i locally computes C^i_{yz}.
   Party P_i locally computes n^i_y
  end for
end for

Party p1 generates a random number r_i , an integer X encryption(E) , decryption(D) keys and then encrypts its data $C^1_{yz} + r_1$ and forwards it to the its next party. It also forwards the encryption key to the next party.

Each of the remaining parties p_i generate a random number and compute $E(C^1_{yz} + C^2_{yz} +..+ C^i_{yz} + \sum_{k=1}^{i} r_k)$ and passes it to p(i+1).

At last p1 obtains $D(E(\sum_{j=1}^{n} C^j_{yz} + X*\sum_{j=1}^{n} r_j))$mod $X = \sum_{j=1}^{n} C^j_{yz}$, for every class y and attribute value z which is assigned to C_{yz}.

Similarly for every class value y, all parties jointly calculate $n_y = \sum_{i=1}^{k} n^i_y$.

Party 1 then calculates $p_{yz} = C_{yz}/ n_y$ and broadcasts p_{yz} to the other parties.

2) Handling numeric attributes

Requirements: k parties, r class values

x_{iyj} represents the values of instances j from party i having class value y

s^i_y represents the sum of instances from party i having class value y

n^i_j represents the number of instances with party P_i having class value y

for all the class values y do
  for i= 1 to k do
    Party P_i locally computes $s^i_y = \sum_j x_{iyj}$
    Party P_i locally computes n^i_y
  end for

Party p1 generates a random number r_i , an integer X encryption(E) , decryption(D) keys and then encrypts its data $s^1_y + r_1$ and forwards it to the its next party. It also forwards the encryption key to the next party.

Each of the remaining parties p_i generate a random number and compute $E(s^1_y + s^2_y +..+ s^i_y + \sum_{k=1}^{i} r_k)$ and passes it to p(i+1).

At last p1 obtains S_y by D $(E(\sum_{j=1}^{n} S^j_y + X*\sum_{j=1}^{n} r_j))$ mod $X = \sum_{j=1}^{n} S^j_y$, for every class y.

Party 1 again initiates the secure sum addition protocol to compute $n_y = \sum_{i=1}^{k} n^i_y$ collaborating with other sites.

Party 1 then computes mean $\mu_y = s_y/ n_y$

μ_y is circulated to all the other sites.

for i=1 to k do
 for every instance j, $v_{iyj} = x_{iyj} - \mu_y$ and $v_{iy} = \sum_j v^2_{iyj}$
end for

Party 1 again initiates the secure sum addition protocol to compute variance $v_y = \sum_{i=1}^{k} v_{iy}$ collaborating with other sites.

Finally party 1 computes stan_dev $\sigma^2_y = \frac{1}{ny-1}. v_y$.

For numeric and categorical attributes the computations involved in calculating the probability is slightly expensive compared to the earlier approach but secure even if two or more parties involved in the computation reveal their values to each other.

The important assumption for performing secure addition requires more than 2 parties and all the parties involved in the computation in the form of a ring. Also, division is performed by Party 1 itself which has both the numerator and the denominator.

B. Naïve Bayesian Classifier Using Secure Multi-Party Addition.

Another method of performing secure computations is indicated in the following algorithms. In these approaches it is not necessary to place all the parties in a ring. One of the parties (we have assumed the last party), initiates the secure sum and interacts with the other parties as used in [13].

1) Handling a categorical attribute

Requirements: k parties, r class values, x attribute values

C^x_{yz} – represents # of instances with party P_x having class y and attribute value z.

n^x_y – represents # of instances with party P_x having class y.

p_{yz} - represents the probability of an instance having class y and attribute value z.

for all class value y do
  for i= 1 to k do
   for every attribute value z, party P_i locally computes C^i_{yz}.
   Party P_i locally computes n^i_y
  end for
end for

Party P_k select k-1 numbers $x_{k,1}$, $x_{k,2}$, …., $x_{k,k-1}$ such that

$x_k = x_{k,1} + x_{k,2} + \ldots + x_{k,k-1}$
Every other party P_i , $1 \leq i \leq n-1$, uses the homomorphic paillier approach to compute $C^i_{yz} + x_{k,1} = y_{k,i} * y_k$.

$y_{k,1} * y_k + y_{k,2} * y_k + \ldots + y_{k-1,1} * y_k$ will give C_{yz} which is $= \sum_{j=1}^{n} C^j_{yz}$, for every class y and attribute value z which is assigned.
Similarly for every class value y, all parties jointly calculate $n_y = \sum_{i=1}^{k} n^i_y$.
Party 1 then calculates $p_{yz} = C_{yz} / n_y$ and broadcasts p_{yz} to the other parties.

2) Handling numeric attributes

Requirements : k parties, r class values

x_{iyj} represents the values of instances j from party i having class value y

s^i_y represents the sum of instances from party i having class value y

n^i_j represents the number of instances with party P_i having class value y

for all the class values y do
for i= 1 to k do
Party P_i locally computes $s^i_y = \sum_j x_{iyj}$
Party P_i locally computes n^i_y
end for

Party P_k select k-1 numbers $x_{k,1}, x_{k,2}, \ldots, x_{k,k-1}$ such that
$x_k = x_{k,1} + x_{k,2} + \ldots + x_{k,k-1}$
Every other party P_i , $1 \leq i \leq n-1$, uses the homomorphic paillier approach to compute $S^i_y + x_{k,1} = y_{k,i} * y_k$.
$y_{k,1} * y_k + y_{k,2} * y_k + \ldots + y_{k-1,1} * y_k$ will give S_y which is $= \sum_{j=1}^{n} S^j_y$, for every class y .

Similarly as above Party P_k computes $n_y = \sum_{i=1}^{k} n^i_y$ collaborating with other sites.

Party 1 then computes mean $\mu_y = S_y / n_y$
μ_y is circulated to all the other sites.

for i=1 to k do
for every instance j, $v_{iyj} = x_{iyj} - \mu_y$ and $v_{iy} = \sum_j v^2_{iyj}$
end for
Party K again initiates the addition protocol as above to compute variance $v_y = \sum_{i=1}^{k} v_{iy}$ collaborating with other sites.
Finally party 1 computes stan_dev $\sigma^2_y = \frac{1}{ny-1} . v_y$.

For numeric and categorical attributes the computations involved in calculating the probability is slightly expensive compared to the earlier approach but secure even if two or more parties involved in the computation reveal their values to each other.

The computation of secure protocols should involve more than 2 parties but need not place the parties in a ring for calculating the sum.

C. Evaluating an Instance

All the attributes including the class label attributes are available with all the parties hence the party that wants to evaluate an instance uses the probabilities and standard deviations obtained using algorithms in A or B locally to classify an instance. The estimated cost for classifying an instance for census data set [14] is shown in Table I.The other parties do not collaborate in the process. Hence no privacy is compromised.

TABLE I. COMPUTATION COST FOR CLASSIFYING AN INSTANCE

Security Parameter (in bits)	# of categorical attributes	# of numeric attributes	Estimated Time (seconds)
512	5	5	3.655
512	7	5	5.763
512	9	5	8.456
1024	5	5	6.453
1024	7	5	10.789
1024	9	5	17.567

D. Security Issues

Protocols for categorical attributes mentioned above securely compute the probabilities P_{yz}, C_{yz} or n_y without revealing any of the intermediate results as the values are encrypted when moved from one site to another. When we apply the composition theorem [15] where g indicates the categorical attribute probability computation algorithm and f is the secure addition algorithm used in both the protocols.

Also the protocols for numeric attributes securely compute the mean and variances. Here also we apply the composition theorem where g indicates the numeric attribute probability computation algorithm and f is the secure addition algorithm used in both the protocols.

E. Implementation

The algorithms are implemented in Java in Eclipse IDE. The testing data sets are from the Irvine dataset repository [14]. We choose the census data set where we use 14 categorical and 7 numeric attributes for building a model on the salary class attribute. We have performed experiments based on the varied size of the datasets maintained at other parties.

1) Experimental results

We have performed our experiments on the non-privacy naïve Bayesian classification version the privacy versions that we have implemented. Accuracy loss [7] is calculated as T_1-T_2. Where T_1 is the test error rate for non-privacy version and T_2 is the test error rate for privacy.
Test Error Rate = (Number of test samples misclassified)/(Total number of samples).

For the census dataset with the salary attribute as class label attribute our results is mentioned in Table II.

TABLE II. TEST ERROR RATE COMPARISON

Algorithm	**Test error rate(approx)**
Non-privacy Naïve Bayesian	65%
Naïve Bayesian homomorphic	55%
Naïve Bayesian Secure Multiparty Addition	54%

Because of the cryptographical operations we noticed a slight decrease in accuracy. Since the accuracy loss is

within limits, our approaches are quite effective in learning real world datasets. Also cryptographic algorithms are essential whenever there are privacy issues.

III. Conclusion

In this paper we present two versions of the privately generating a naïve Bayesian classifier with more than two parties. We assume that the data is horizontally partitioned at the sites. Secure computations were performed on the data without disclosing any intermediate results. The experimental results performed on real world data show that the accuracy losses are within limits.

In our future work we would further explore ways of privacy preservation in classifiers.

Acknowledgment

The authors wish to thank M S Ramaiah Institute of Technology and Manipal Institute of Technology for providing us the required support in conducting our work.

References

[1] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proc. 2000 ACM SIGMOD Conference on Management of Data*, ACM, Dallas, TX, 2000, pp. 439-450.
[2] *HIPPA. National Standards to Protect the Privacy of Personal Health Information*. [Online]. Available: http://www.hhs.go/or/hipaa/finalreg.html
[3] V. Jaideep, K. Murat, and C. Chris, "Privacy preserving naïve bayes classification," *The VLDB Journal*, pp. 879-898.
[4] Y. Lindell and B, Pinkas, "Privacy preserving data mining," *J. Cryptol*, vol. 15, no. 3, pp. 177-206, 2002.
[5] S. Samet and A. Miri, "Privacy presering ID3 using Gini index over horizontally partitioned data," in *Proc. 6th ACS/IEEE International Conference on Computer Systems and Applications*, Doha, Qatar, 2008, pp. 645-651.
[6] R. Wright and Z. Yang, "Privacy-preserving Bayesian network structure computation on distributed heterogeneous data," in *Proc. 10th ACM SIGKDD Int. Conf.knowledge.Disc. Data Mining*, 2004, pp. 713-718.
[7] A. Bansal, T Chen, and S Zhong, "Privacy preserving back-propagation neural network learning over arbitrarily partitioned data," *Neural Computing and Appications*, vol. 20, pp. 143-150, 2011.
[8] T. Chen and S Zhong, "Privacy-preserving back propagation neural network learning," *IEEE Transactions on Neural Networks*, vol. 20, no. 10, 2009.
[9] T. Mitchell, *Machine Learning*, 1st ed. *McGraw-Hill* Science/Engineering/Math, New York, 1997.
[10] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. International Conference on the Theory and Application of Cryptographic Techniques*, Prague, Czech Republic, 1999, pp. 223-238.
[11] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. IT-31, no. 4, pp. 469-472, July 1985.
[12] M. Blum and S. Goldwasser, "An efficient probabilistic public-key encryption that hides all partial information," in *Advances in Cryptography-Crypto 84 Proceedings*, *R. Blakely*, Ed., Springer, Heidelberg, 1984.
[13] S. Samet and A. Miri, "Privacy preserving back-propagation and extreme learning machine algorithms," *Data and Knowledge Engineering*, vol. 79-80, pp. 40-61, 2012.
[14] C. L. Blake and C. J. MErz. (1998). UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, CA. [Online]. Available: http://www.ics.uci.edu/mlearn/MLRepository.html
[15] O. Goldreich, "The foundations of cryptography," vol. 2, in *General Cryptographic Protocols*, Cambridge: Cambridge University Press, 2009.

Hareesha K. S. born in Chikmagalore, Karnataka State, India on 11th April, 1970. He has received the BCA, MCA in computer applications and Ph.D. in computer science & engineering from Kuvempu Universtiy, Shankaraghatta, Karnataka, India in the year 2000, 2003 and 2008 respectively. His major research interest is digital image processing, data mining, bioinformatics and artificial intelligence. Since 2008, he is working in Manipal University, Karnataka India as Associate Professor, before to this he was worked in Bapuji Institute of Technology, Davangere, Karnataka, India. Also, took charge as Head of Department, Department of Computer Applications, Manipal Institute of Technology, Manipal University from Jan. 2013. His research interests encompass privacy preservations in data mining, spatial data mining and its relevance in society as a whole, bio-informatics, biologically inspired algorithms, soft computing and computer vision. He has published quite a good number of research papers in these areas. He has got fellowship award from Boston University to present his research contributions. Dr. Hareesha K.S. has been a life member of ISTE, New Delhi, India and IACSIT, Singapore. He has been a member of numerous program committee of IEEE, IACSIT conferences in the area of Data Mining, Bioinformatics, Digital Image Processing and Artificial Intelligence.

Sumana M was born in Madikeri, Karnataka, India on 15th December 1978. She has received her B.E. degree in Computer Science and engineering from Manipal Institute of Technology, Karnataka, India, in 2000, and her M.Tech. degree from VTU University in 2007, Karnataka, India and is currently pursuing Ph.D. degree in privacy preserving data mining in computer science and Engineering from the Manipal University, Karnataka, India. She is presently working as an Assistant Professor in the department of Information Science and Engineering in M S Ramaiah Institute of Technology since 2007. Previously she had worked as a lecturer in the Manipal Institute of Technology. Her research interests include data mining, cryptography and secure multiparty computations. She is a Life Member of the Indian Society for Technical Education (ISTE), the System Society of India.

