ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 8, No. 2, 2013, *pp*.119-129

Privacy Preserving Three-Layer Naïve Bayes Classifier for Vertically Partitioned Databases

Alka Gangrade [1] and Ravindra Patel [2]

[1] Technocrats Institute of Technology, Bhopal, India, Email: alkagangrade@yahoo.co.in
[2] Dept. of M.C.A., U.I.T., R.G.P.V., Bhopal, India, Email: ravindra@rgtu.net

(*Received October 28, 2012, accepted February 29, 2013)*

Abstract. Data mining is the extraction of the hidden information from large databases. It is a powerful technology to explore important information in the data warehouse. Privacy preservation is a significant problem in the field of data mining. It is more challenging when data is distributed among different parties. In this paper, we address the problem of privacy preserving three-layer Naïve Bayes classification over vertically partitioned data. Our approach is based on Secure Multiparty Computation (SMC). We use secure multiplication protocol to classify the new tuples. In our protocol, secure multiplication protocol allows to meet privacy constraints and achieve acceptable performance and our classification system is very efficient in term of computation and communication cost.

Keywords: Privacy preserving, Naïve Bayes classification, probability, secure multiplication protocol.

1. Introduction

Boundary heat

Classification is a popular data mining technique used to predict group membership for data tuples. In classification rule mining, a set of database tuples act as a training sample and it is analyzed to produce a model of the data or classifier that can be used for classifying a new tuple. The popular classification rule mining techniques are decision trees, neural networks, Naïve Bayesian classifiers etc. Preserving privacy against data mining algorithms is a new research area. Privacy preserving data mining is the emerging field that protects sensitive data. The goal of privacy preserving classification is to build precise classifiers without disclosing personal information in the data being mined.

1.1. Naïve Bayesian Classification

Bayesian classification is based on Bayes' theorem [1]. A simple Bayesian classifier is known as the Naïve Bayesian classifier, to be comparable in performance with decision tree and selected neural network classifier. Bayesian classifiers have also exhibited high accuracy and speed when applied to large database.

Bayes' theorem is

$$P(H|X) = \frac{P(X|H)\,P(H)}{P(X)} \tag{1}$$

Where H is some hypothesis, such as that the data tuple X belongs to a specified class 'C'. For classification problems, we want to determine P (H|X), the probability that the hypothesis H holds given the "evidence" or observed data tuple X.

P (H|X) is the posterior probability of H conditioned on X.

P (H) is the prior probability of H. For our example; this is the probability that any given customer will buy a computer, regardless of age, income or any other information.

P (X|H) is the posterior probability of X conditioned on H.

Naïve Bayes is extremely effective but straightforward classifier. Due to this combination of straightforward and effectiveness it is used as a baseline standard by which other classifiers are measured. Naïve Bayes represents each class with a probabilistic summary, and classify each new tuple with the most

likely class. It provides a flexible way for dealing with any number of attributes or classes, and is based on probability theory. It is fast learning algorithm that examines all its training input. It has been established to achieve unexpectedly well in a wide variety of problems despite of the simple nature of the model. With various enhancements it is highly effective, and receives practical use in many applications for example content based filtering and text categorization.

For preserving privacy we use the framework defined in Secure Multiparty Computation [2], and several primitives from the Secure Multiparty Computation contents. Complete details of Naïve Bayes classification algorithms can be found in [3]. We assume that the basic formulae are well known. In order to construct a privacy preserving Naïve Bayesian classifier, we must concentrate on two issues, how to calculate the probability or model parameter for each attribute and how to classify a new tuple [4, 5, 6]. The following subsections provide details on both issues. The protocol presented below is quite efficient.

1.2. Our Contributions

Our main contributions in this paper are as follows:

- We present a novel privacy preserving Naïve Bayes classifier for vertically partitioned databases.
- It classifies new tuple by using secure multiplication protocol. We propose a new protocol.

1.3. Organization of the paper

The rest of the paper is organized as follows. In Section 2, we discuss the related work. Section 3, describes proposed work of our novel privacy preserving Naïve Bayes classification model for vertically partitioned data. Section 3.1 describes architecture of our model and secure multiplication protocol. Section 3.2 sets some assumptions. Section 3.3 describes formal algorithms of our proposed work. In Section 4, we present our calculation and results that are conducted by using our proposed model on real-world data sets. In Section 5, we conclude our paper with the discussion of the future work.

2. Related Work

Privacy preserving data mining has been an active research area for a decade. A lot of work is going on by the researcher on privacy preserving classification in distributed data mining. The first Secure Multiparty Computation (SMC) problem was described by Yao [7]. SMC allows parties with similar background to compute result upon their private data, minimizing the threat of disclosure was explained [8].

There have been several approaches to support privacy preserving data mining over multi-party without using third parties [9, 10]. Some techniques, review and evaluation of privacy preserving algorithms also presented in [9]. Various tools discussed and how they can be used to solve several privacy preserving data mining problems [11]. We now give some of the related work in this area. Previous work in privacy preserving data mining has addressed some issues. The aim is to preserve customer privacy by distorting the data values presented in [10]. D. Agrawal and C. C. Aggarwal designed various algorithms for improving this approach [12].

Classification is one of the most widespread data mining problems come across in real life. General classification techniques have been extensively studied for over twenty years. The classifier is usually represented by classification rules, decision trees, Naïve Bayes classification and neural networks. First ID3 decision tree classification algorithm is proposed by Quinlan [13]. Lindell and Pinkas proposed a secure algorithm to build a decision tree using ID3 over horizontally partitioned data between two parties using SMC [14]. A novel privacy preserving distributed decision tree learning algorithm [15] that is based on Shamir [16]. The ID3 algorithm is scalable in terms of computation and communication cost, and therefore it can be run even when there is a large number of parties involved and eliminate the need for third party and propose a new method without using third parties. A generalized privacy preserving variant of the ID3 algorithm for vertically partitioned data distributed over two or more parties introduced in [17, 18, 19, 20] and horizontally partitioned data distributed over multi parties introduced in [21, 22]. Privacy preserving Naïve Bayes classification for horizontally partitioned data introduced in [4] and vertically partitioned data introduced in [5, 6]. Centralized Naïve Bayes classification probability calculation is introduced in [23].

3. Proposed Work

This paper addresses classification over vertically partitioned data, where different parties hold different attributes. We consider the case where all party holds the class attributes. In this case, all party calculates

probabilities (model parameters) of all class value for each attribute value for every attribute individually, causing no privacy breaches. For classifying the new tuple, in vertical partition database all party has to collaborate with each other in a secure manner. We use secure multiplication protocol for multiplying the probabilities (model parameters) of particular attribute value of all attribute for all class value and compare total probability of all class value and find out the maximum total probability. The class having maximum total probability will be the predicted class.

For vertically partitioned case, different parties must collaborate to find the classification result for every new tuple because no party has all the attributes for a given tuple. Since parties do not know all the attributes of a new tuple, they will not be able to predict the full model even by classifying many tuples.

Therefore, hiding the model brings additional security for vertically partitioned data and is necessary. For classifying the new tuple, we use secure multiplication protocol. Secure multiplication protocol keeps hiding not only the model parameters or probabilities from other party even data also.

3.1. System Architecture

Proposed architecture of our privacy preserving Naïve Bayes Classifier for vertically partitioned databases is shown in Fig. 1. It has three layers.

Input Layer – All participating parties that are involved in the classification process individually calculate probability or model parameters for all class value of each attribute value for every attribute.

Intermediate Layer – In vertically partitioned data, no party has all the attributes. They must collaborate to find total probability for all classes. For this we propose Secure Multiplication Protocol, no party is able to know the probabilities or model parameters, not even data of the other parties. Only first party will know the total probability for all classes. Secure multiplication protocol is shown in Fig. 2.

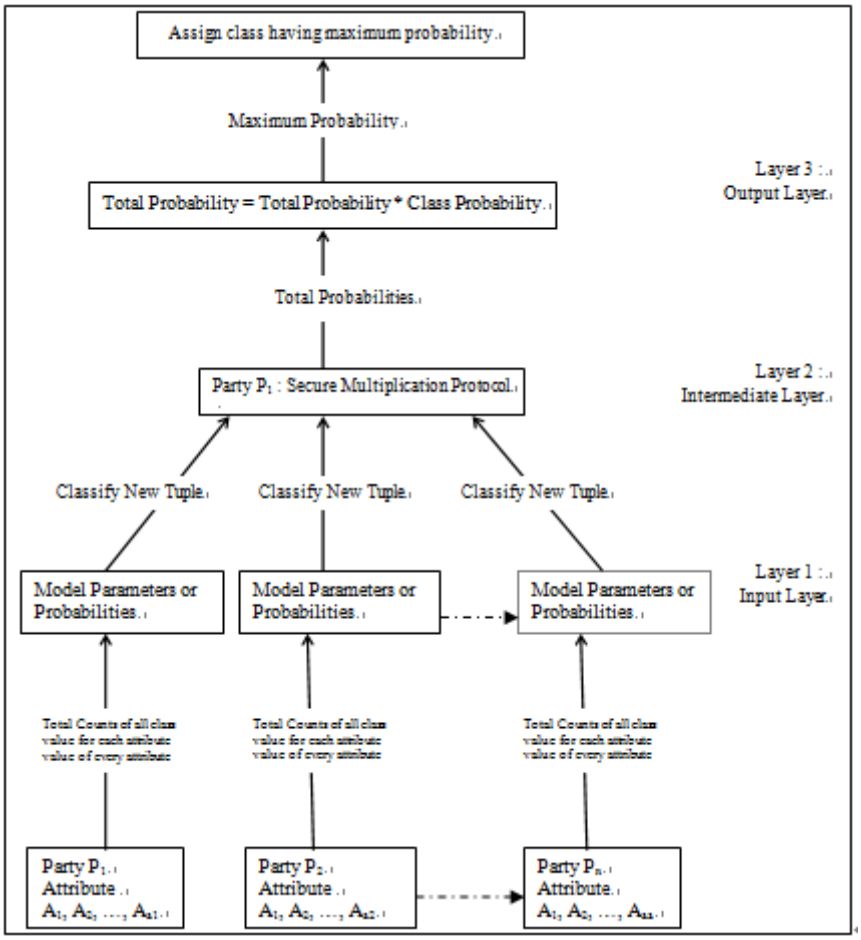

Fig. 1: 3LPPVPNBC system architecture

Output Layer – Based on the total probability of all class value, first party will find the class with the

highest total probability and finally classify the new tuple. Send this class value to all other parties.

Protocol is secure to classify each new tuple.

3.2. Assumptions

The following assumptions have been set:

- All participating parties individually calculate probabilities or model parameters.
- All parties have to collaborate for classify new tuple, therefore security is must.
- Party P_1 (first party) drives the secure multiplication protocol.
- Party P_1 (first party) calculates the total probabilities.
- Send class value to all parties.
- Input data of all parties are secured and privacy is preserved.
- The Secure Multiplication Protocol used by the input parties is secured.

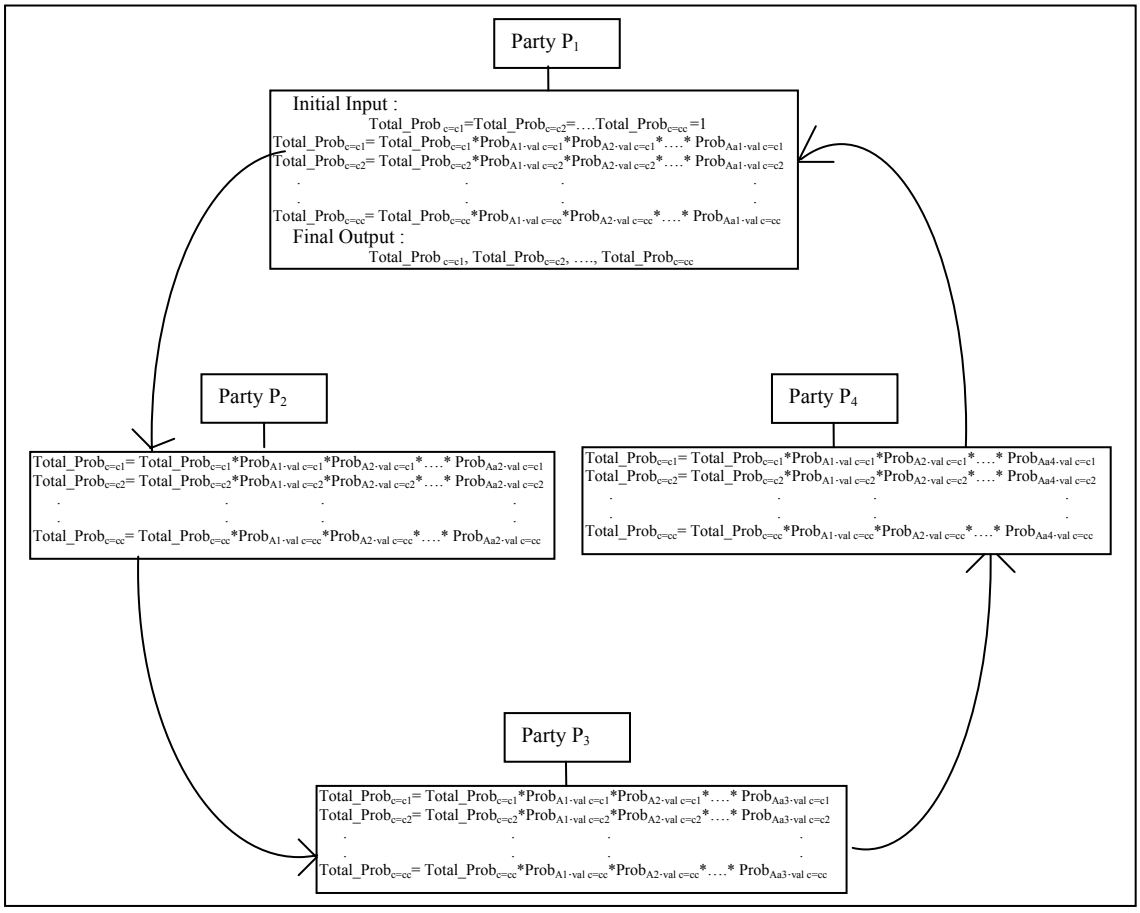

Fig. 2: Secure Multiplication Protocol

3.3. Formal Algorithms

Require:

- n parties i.e. P_1, P_2, .., P_n{Vertically Partitioned}
- c class values i.e. c_1, c_2, .., c_c,
- a attributes i.e. a={a_1+a_2+…+a_n} and A_c is the class attribute.

  P_i —> $P_i.A_1$, $P_i.A_2$, .., $P_i.A_{ai,}$ A_c. { i = 1, 2, …, n}

Note:

- $P_i.C_{xyz}$: represents number of tuples having class z, attribute value y of attribute A_x of party P_i.
- $P_i.A_{xy}$: represents attribute name A_x with attribute value y of party P_i.
- N_z : represents number of tuples having class z.
- T : represents total tuples.

- $P_i.Prob_{xyz}$: represents probability of attribute A_x with attribute value y having class z of party P_i.
- $Prob_z$: represents probability of class z.
- $New.P_i.A_{xy}$: represents new tuple of attribute name A_x with attribute value y of party P_i to be classified.

Algorithm 1: 3LPPVPNBC ( ) - Three-Layer Privacy preserving vertically partitioned NBC. Basically it has three steps to classify the new tuple.

Step 1: Class_Count ( ) { calculate class count }
Step 2: Calculate_Attribute_Prob ( ) { calculate probabilities of all attributes }
Step 3: Find_Max_Prob( ) { calculate maximum probability by calling Cal_Total_Prob( ) }

Input Layer
Algorithm 2: Calculate_Attribute_Prob ( ): Calculate probabilities of all class value of each attribute value for every attribute of all parties.

1. For Party P_i where i = 1 to n do
2. For Attribute A_x where x = 1 to a_i do
3. For Attribute value V_y where y = 1 to v_x do
4. For Class value c_z where z = 1 to c do
5. $P_i.C_{xyz} = 0$
6. For all tuples having class value c_z
7. $P_i.C_{xyz} = P_i.C_{xyz} + 1$
8. End for
9. $P_i.Prob_{xyz} = P_i.C_{xyz} / N_z$
10. End for
11. End for
12. End for
13. End for.

Algorithm 3: Class_Count ( ) : Calculate class count for all class value by first party

1. For Class value c_z where z = 1 to c do
2. $N_z = 0$
3. For all tuples having class value c_z
4. $N_z = N_z + 1$
5. End for
6. End for.

Intermediate Layer: Secure Multiplication Protocol
Algorithm 4: Cal_Total_Prob (c_z) : Calculate total Probability for all class value of new tuple by using secure multiplication protocol.

1. For Class value c_z where z = 1 to c do
2. $Total\_Prob_z = 1$
3. For Party P_i where i = 1 to n do
4. For Attribute A_x where x = 1 to a_i do
5. For Attribute value V_y where y = 1 to v_x do
6. If $P_i.A_{xy} = New.P_i.A_{xval}$ then
7. $Total\_Prob_z = Total\_Prob_z * P_i.Prob_{xyz}$
8. Break
9. End if

10. End for
11. End for
12. End for
13. Return Total_Prob$_z$
14. End for.

Output Layer:

Algorithm 5: Find_Max_Prob ( ) : Find the maximum probability and classify the tuple.

1. Max_Prob = 0
2. Class = Null
3. For Class value c_z where z = 1 to c do
4. Prob = Cal_Total_Prob (c_z) * N_z/T
5. If Prob > Max_Prob then
6. Max_Prob = Prob
7. Class = c_z
8. End if
9. End for
10. For Party P_i where i = 1 to n do
11. A_c = Class
12. End for.

4. Evaluation and Results

In vertical partitioned data, party needs collaboration with other parties to classify the new tuple. Here we are addressing two parties, where parties are vertically distributed. Each party has three attributes including class attribute. Naïve Bayes evaluation procedure uses secure multiplication protocol to classify the new tuple. Our protocol secured the parties actual data in the process. Thus, privacy is being maintained. Its execution time is less than the existing Naïve Bayes classifier with almost same accuracy.

Table 1. Party P_1

Rid	P_1.Age	P_1.Income	P_1.Class:Buys_computer
1	<=30	High	No
2	<=30	High	No
3	31..40	High	Yes
4	>40	Medium	Yes
5	>40	Low	Yes
6	>40	Low	No
7	31..40	Low	Yes
8	<=30	Medium	No
9	<=30	Low	Yes
10	>40	Medium	Yes
11	<=30	Medium	Yes
12	31..40	Medium	Yes
13	31..40	High	Yes
14	>40	Medium	No

Table 2. Party P_2

Rid	P_2.Student	P_2.Credit_rating	P_2.Class:Buys_computer
1	No	Fair	No
2	No	Excellent	No
3	No	Fair	Yes
4	No	Fair	Yes
5	Yes	Fair	Yes
6	Yes	Excellent	No
7	Yes	Excellent	Yes
8	No	Fair	No
9	Yes	Fair	Yes
10	Yes	Fair	Yes
11	Yes	Excellent	Yes
12	No	Excellent	Yes
13	Yes	Fair	Yes
14	No	Excellent	No

Probabilities calculation of the attributes of vertically partitioned databases:

Total Number of tuples = 14

Class Yes: Buys_computer = "Yes" Total tuples for Class Yes = 9

Class No: Buys_computer = "No" Total tuples for Class No = 5

Table 3. Compute probability for P_1.Age

P_1.Age	Class Yes		Class No	
	Total	Probability	Total	Probability
<=30	2	0.2222	3	0.6
31..40	4	0.4444	0	0.0
>40	3	0.3333	2	0.4

Table 4. Compute probability for P_1.Income

P_1.Income	Class Yes		Class No	
	Total	Probability	Total	Probability
High	2	0.2222	2	0.4
Medium	4	0.4444	2	0.4
Low	3	0.3333	1	0.2

Table 5. Compute probability for P_2.Student

P_2.Student	Class Yes		Class No	
	Total	Probability	Total	Probability
Yes	6	0.6667	1	0.2
No	3	0.3333	4	0.8

Table 6. Compute probability for P_2.Credit_rating

P_2.Credit_rating	Class Yes		Class No	
	Total	Probability	Total	Probability
Fair	6	0.6667	2	0.4
Excellent	3	0.3333	3	0.6

Table 7. Compute probability for Class Attribute (Buys_computer)

Class : Buys_computer	Class Yes		Class No	
	Total	Probability	Total	Probability
	9	0.6429	5	0.3571

Table 8. Classify new tuples of Party P_1

Rid	P_1.Age	P_1.Income	P_1.Class:Buys_computer
15	<= 30	Low	?
16	>40	High	?
.	.	.	.

Table 9. Classify new tuples of Party P_2

Rid	P_2.Student	P_2.Credit_rating	P_2.Class:Buys_computer
15	Yes	Excellent	?
16	No	Excellent	?
.	.	.	.

For Rid =15

Likelihood of the two classes:

For Class Yes = 0.2222 * 0.3333 * 0.6666 * 0.3333 * 0.6429 = 0.01058

For Class No = 0.6 * 0.2 * 0.2 * 0.6 * 0.3571 = 0.00514

Conversion into a probability by normalization

P (Class Yes) = 0.01058/ (0.01058 + 0.00514) = 0.673

P (Class No) = 0.00514/ (0.01058 + 0.00514) = 0.327

Here P (Class Yes) > P (Class No) then Class: Buys_computer = Yes

For Rid =16

Likelihood of the two classes:

For Class Yes = 0.3333 * 0.2222 * 0.3333 * 0.3333 * 0.6429 = 0.00529
For Class No = 0.4 * 0.4 * 0.8 * 0.6 * 0.3571 = 0.02743
Conversion into a probability by normalization
P (Class Yes) = 0.00529/ (0.00529 + 0.02743) = 0.162
P (Class No) = 0.02743/ (0.00529 + 0.02743) = 0.838

Here P (Class Yes) < P (Class No) then Class: Buys_computer = No

Table 10. Execution Time Calculation

Number of Instances	NB Execution Time(ms)	3LPPVPNBC Execution Time(ms)
14	70	14
25	83	15
50	99	16
100	112	29
200	135	31

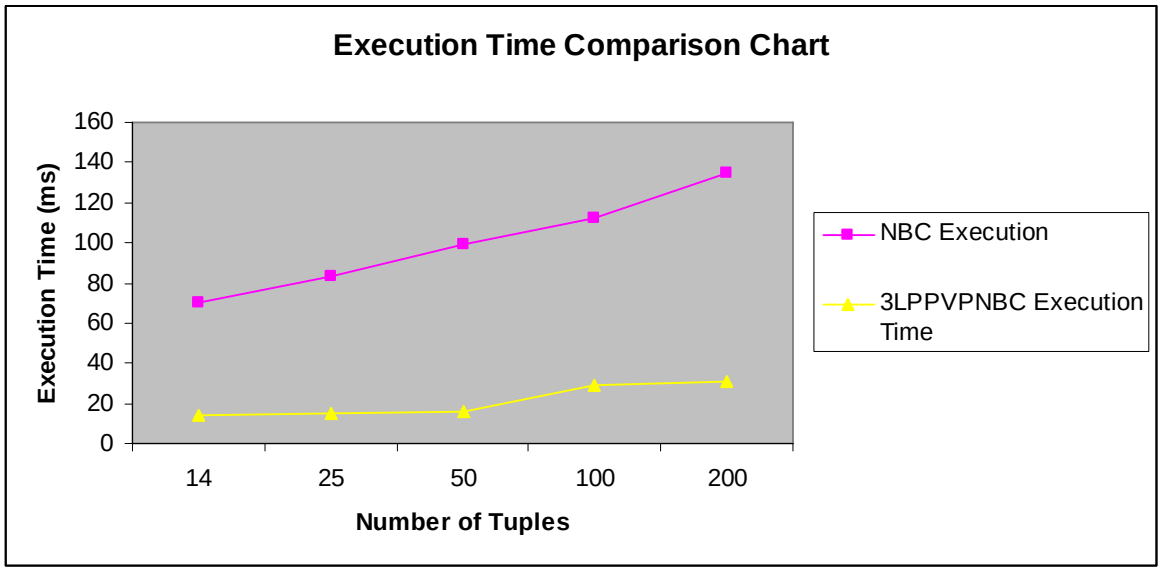

Fig. 3: Execution Time Comparison

Table 11. Accuracy Measurement

Number of Instances	3LPPVPNBC Accuracy(%)
14	72.74%
25	73.54%
50	74.68%
100	76.25%
200	77.51%

5. CONCLUSION

In conclusion, we provide a novel solution for Naïve Bayes classification over vertically partitioned data. Instead of using data transformation, we define a secure multiplication protocol to transform the model parameters while keeping data values secure. Our classification system is quite efficient and fast because the running time of our classifier is less than existing Naïve Bayes classifier. It is also much less than ID3 and C4.5 decision tree classifier because Bayesian classifier only needs to go through the whole training data

once. They are also space efficient because they build up a frequency table in size of the product of the number of attributes, number of class values, and the number of values per attribute not the actual value of the attribute. We are continuing work in this field to develop Naïve Bayes classifier for horizontally partitioned databases and also analysis new as well as existing classifiers.

Acknowledgements

We are thankful to the University and the College for their support. We express gratitude to my colleagues for their technical support and the referees for their beneficial suggestions.

6. References

[1] J. Han, and M. Kamber, *Data Mining: Concepts and Techniques*, Elsevier, India.

[2] Oded Goldreich, *Secure multi-party computation*, Sep 1998. (working draft).

[3] Tom Mitchell, *Machine Learning*, McGraw-Hill Science/Engineering/Math, 1st edition, 1997.

[4] M. Kantarcioglu, and J. Vaidya, *Privacy preserving naive Bayes classifier for horizontally partitioned data*, In IEEE ICDM Workshop on Privacy Preserving Data Mining, Melbourne, FL, pp. 3-9, November 2003.

[5] J. Vaidya, and C. Clifton, *Privacy preserving naive Bayes classifier on vertically partitioned data*, Proc. SIAM International Conference on Data Mining, Lake Buena Vista, Florida, pp. 22-24, April 2004.

[6] Z. Yang, and R. Wright, *Privacy-Preserving Computation of Bayesian Networks on Vertically Partitioned Data*, IEEE Transactions on Data Knowledge Engineering, 18(9), April 2006, pp. 1253-1264.

[7] A. C. Yao, *Protocols for secure computation*, Proc. of 23rd IEEE Symposium on Foundations of Computer Science (FOCS), pp. 160-164, 1982.

[8] W. Du, and Mikhail J. Attalah, *Secure multi-problem computation problems and their applications: A review and open problems*, Tech. Report CERIAS Tech Report 2001-51, Center for Education and Research in Information Assurance and Security and Department of Computer Sciences, Purdue University, West Lafayette, IN 47906, 2001.

[9] V. Verykios, and E. Bertino, *State-of-the-art in Privacy preserving Data Mining*, SIGMOD Record, 33(1), 2004, pp. 50-57.

[10] R. Agrawal, and R. Srikant, *Privacy preserving data mining*, Proc. of the ACM SIGMOD on Management of data, Dallas, TX USA, pp. 439-450, May 15-18, 2000.

[11] C. Cliffton, M. Kantarcioglu, and J. Vaidya, *Tools for privacy preserving distributed data mining*, ACM SIGKDD Explorations Newsletter, 4(2), 2004, pp. 28-34.

[12] D. Agrawal, and C. C. Aggarwal, *On the design and quantification of privacy preserving data mining algorithms*, Proc. of the Twentieth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Santa Barbara, California, USA, pp. 247–255, May 21-23 2001.

[13] J. R. Quinlan, *Induction of decision trees*, in: Jude W. Shavlik, Thomas G. Dietterich, (Eds.), Readings in Machine Learning, Morgan Kaufmann, 1, 1990, pp. 81–106.

[14] Y. Lindell, and B. Pinkas, *Privacy preserving data mining*, Journal of Cryptology, 15(3), 2002, pp. 177–206.

[15] F. Emekci , O. D. Sahin, D. Agrawal, and A. El Abbadi, *Privacy preserving decision tree learning over multiple parties*, Data & Knowledge Engineering 63, 2007, pp. 348-361.

[16] A. Shamir, *How to share a secret*, Communications of the ACM, 22(11), 1979, pp. 612-613.

[17] W. Du, and Z. Zhan, *Building decision tree classifier on private data*, In CRPITS, 2002, pp. 1–8.

[18] J. Vaidya, C. Clifton, M. Kantarcioglu, and A. S. Patterson, *Privacy-preserving decision trees over vertically partitioned data*, Proc. of the 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security, pp. 139–152, 2008.

[19] J. Shrikant Vaidya, *Privacy preserving data mining over vertically partitioned data*, doctoral diss., Purdue University, August 2004.

[20] W. Fang, and B. Yang, *Privacy Preserving Decision Tree Learning Over Vertically Partitioned Data*, Proc. of the 2008 International Conference on Computer Science & Software Engineering, pp. 1049-1052, 2008.

[21] Alka Gangrade, and R. Patel, *A novel protocol for privacy preserving decision tree over horizontally partitioned data*, International Journal of Advanced Research in Computer Science, 2 (1), 2011, pp. 305-309.

[22] Alka Gangrade, and R. Patel, *Privacy Preserving Two-Layer Decision Tree Classifier for Multiparty Databases*, International Journal of Computer and Information Technology (2277 – 0764), 1(1), 2012, pp. 77-82.

[23] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining Practical Machine Learning Tools and Techniques*, Burlington, MA, Morgan Kaufmann, 2011

.