BUILDING PRIVACY-PRESERVING C4.5 DECISION TREE CLASSIFIER ON MULTI-PARTIES

ALKA GANGRADE[1], RAVINDRA PATEL[2]
[1]Technocrats Institute of Technology, Bhopal, MP.
[2]U.I.T., R.G.P.V., Bhopal, MP
email – alkagangrade@yahoo.co.in, ravindra@rgtu.net

Abstract – In this paper, we address Privacy-preserving classification problem in a multi-party sense. We focus the general classification in a secured manner and introduce a Privacy-preserving decision tree classifier using C4.5 algorithm without involving third party. C4.5 algorithm is a software extension of the basic ID3 algorithm designed by Quinlan. Our protocol is considerably more efficient than any existing solutions.

Keywords – Decision tree classification, C4.5 algorithm, Privacy-preserving.

I. INTRODUCTION

In the modern world, huge amount of information of customers is kept in the databases. Thus data mining can be very effective for extracting knowledge from huge amount of data. Classification has many applications in real world, such as stock planning of large superstores, medical diagnosis, etc. Classification is separation or ordering of objects into classes. There are various classification techniques i.e. Decision tree, K-nearest neighbour, Naïve bayes classifier, neural network. In this paper we discuss decision tree.

A decision tree is a popular classification method. The most important feature of decision tree classifier is their ability to break down a complex decision making process into a collection of simpler decisions, thus providing a solution which is often easier to interpret. The characteristics of decision tree methods are:

Decision trees are able to generate understandable rules.

They perform classification without requiring much computation.

They are able to handle both continuous and categorical variables.

They provide a clear indication of which fields are most important for classification.

Decision tree algorithms such as ID3 [1] or C4.5 [2] are among the most powerful and popular methods for classification. The ID3 algorithm is used to design a decision tree based on a given databases. The tree is constructed top-down in a recursive manner. At the root, each attribute is tested to determine how well it alone classifies the transactions. Then, the 'Best' attribute is chosen and remaining records are partitioned by it [3, 4]. ID3 is then recursively called on each partition. C4.5 is a software extension of the basic ID3 algorithm designed by J. R. Quinlan to address the following issues not solved by ID3:

- Avoiding over fitting the data.
- Reduced error pruning.
- Handling continuous attributes also. Example-temperature.
- Handling training data with missing attribute values.

In this paper, we study Privacy-preserving classification rule mining. The objective of Privacy-preserving classification is to build accurate classifiers without disclosing private information in the data being mined. We address the issue of Secure Multi-party Computation (SMC) for classification rule mining. Specifically, SMC enables Privacy-preservation without trusted third party. SMC is one of the great achievements of modern cryptography, enabling a set of untrusting parties to compute any function of their private inputs while revealing nothing but the result of the function. We wish to run Privacy-preserving C4.5 Decision tree classification algorithm on the union of their databases, without revealing any private information.

II. RELATED WORK

Classification is one of the most widespread data-mining problems found in real life. Decision tree classification is one of the best-known solution approaches. ID3, first proposed by Quinlan is a particularly elegant and instinctive solution [1]. This

ISSN : 0975-3397

article presents an algorithm for privately building an ID3 decision tree. While this has been done for horizontally partitioned data [5], Lindell *et al* has proposed a secure algorithm to build a decision tree using ID3 over horizontally partitioned data among two parties using SMC. An algorithm for vertically partitioned data [6] and introduced a generalized Privacy-preserving variant of the ID3 algorithm for vertically partitioned data distributed over two or more parties. A portion of each instance is present at each site, but no site contains complete information for any instance. This problem has been addressed [7], Du *et al* has also proposed a method to build a decision tree over vertically partitioned data using secure scalar product protocol, but the solutions are limited to cases where both parties have the class attribute. Zhang *et al* developed a new scheme based on algebraic techniques [8].

In data mining, several efforts have been made to preserve the privacy of individual records using randomization techniques [9, 10] and to preserve the privacy of the database while running data mining algorithms over multiple data sources using cryptographic techniques such as SMC and encryption [11, 12, 13]. Although their technique can be generalized to more than two parties, it is inefficient and not scalable for a large number of parties [14]. Lindell *et al* discussed the relationship between SMC and Privacy-preserving data mining [15]. Recently, there has been a great interest in the database area for Privacy-preserving database operations such as intersection, join and aggregation operations. Agrawal *et al* used a commutative encryption to answer intersection and join queries over two private databases [16]. F. Emekci *et al* proposed a novel Privacy-preserving distributed decision tree learning algorithm that is based on 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 method without using third parties [17].

Vaidya *et al* proposed algorithms on building decision tree, however, the tree on each party doesn't contain any information that belong to other party, the drawback of this method is that the resulting class can be altered by a malicious party [18]. Fang *et al* proposed algorithms a Privacy-preserving distributed decision-tree mining algorithm, which is based on idea of Privacy-preserving decision tree and passing control from site to site [19]. The drawback of this method is that each party has the class attribute. Missing attribute values are not handled by these methods.

III. PRIVACY-PRESERVING C4.5 DECISION TREE CLASSIFICATION FOR MULTI-PARTY COMPUTATION

The methods of Privacy-preserving data mining depend on the data mining task and the data sources distribution manner such as centralized-where all records are reside in only one party; horizontally-where every party has different records of a database, but each record contains same set of attributes; vertically-where every party has the same number of records, but each record contains different attributes. In this paper, we particularly focus on applying Privacy-preserving C4.5 decision tree classification on vertically partitioned data without using third party. It is based on to calculate the union of all parties databases, no matter that only one party having the class attribute or more than one or all parties. Apply data mining algorithm on these data and sends the output.

Secure set union protocol without using third party

Secure union methods [21] are useful in data mining where each party needs to give rules, frequent itemsets etc., without revealing the owner. The union of items can be evaluated using SMC methods if the domain of the items is small. Each party creates a binary vector where 1 in the i^{th} entry represents that the party has the i^{th} item. After this point, a simple circuit that or's the corresponding vectors can be built and it can be securely evaluated using general secure multi-party circuit evaluation protocols. However, in data mining the domain of the items is usually large. To overcome this problem a simple approach based on commutative encryption is used. An encryption algorithm is commutative if given encryption keys $K_1,\ldots,K_n \in K$, for any m in domain M, and for any permutation i, j, the following two equations hold:

$$E_{K_{i1}}(\ldots E_{K_{in}}(M)\ldots) = E_{K_{j1}}(\ldots E_{K_{jn}}(M)\ldots) \quad (1)$$

$M_1, M_2 \in M$ such that $M_1 = M_2$ and for given k, $\epsilon < 1/2^k$

$$\Pr(E_{K_{i1}}(\ldots E_{K_{in}}(M_1)\ldots) = E_{K_{j1}}(\ldots E_{K_{jn}}(M_2)\ldots)) < \epsilon \quad (2)$$

With shared p the Pohlig-Hellman encryption scheme [22] satisfies the above equations, but any other commutative encryption scheme can be used. The main idea is that each site encrypts its items. Each site then encrypts the items from other sites. Since equation 1 holds, duplicates in the original items will be duplicates in the encrypted items, and can be deleted. (Due to equation 2, only the duplicates will

ISSN : 0975-3397

be deleted.) In addition, the decryption can occur in any order, so by permuting the encrypted items we prevent sites from tracking the source of an item. The algorithm for evaluating the union of the items is given in Algorithm 1, and an example is shown in Figure 1. Clearly algorithm 1 finds the union without revealing which item belongs to which site. It is not, however, secure under the definitions of secure multi-party computation. It reveals the number of items that exist commonly in two sites, e.g. if k sites have an item in common, there will be an (encrypted) item duplicated k times. This does not reveal which items these are, but a truly secure computation (as good as each site giving its input to a "trusted party") could not reveal even this count. Allowing innocuous information leakage (the number of items that is owned by two sites) allows an algorithm that is sufficiently secure with much lower cost than a fully secure approach. We can prove that other than the size of intersections and the final result, nothing is revealed. By assuming that the count of duplicated items is part of the final result, a Secure Multiparty Computation proof is possible.

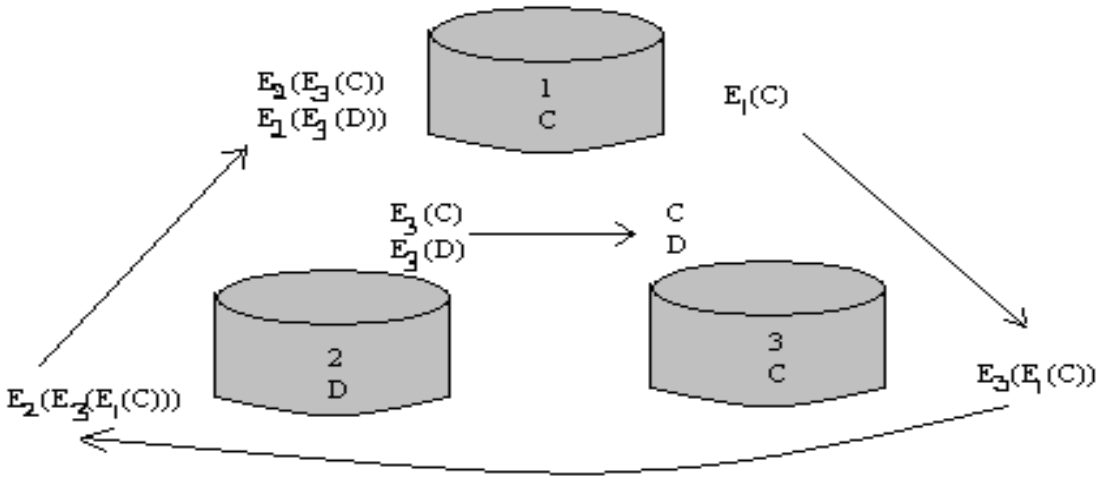

Fig 1. Secure Set Union

Algorithm 1: Finding secure set union of items

Require: N is number of parties or sites and Union_set = ø; initially

{Encryption of all the rules by all sites}

begin

for each site i do

for each X Є S_i do

M = newarray[N];

Xp = encrypt(X, e_i);

M[i] = 1;

Union_set U (Xp, M);

end for

end for {Site i encrypts its items and adds them to the

global set. Each site then encrypts the items

it has not encrypted before}

for each site i do

for each tuple (r, M) Є Union_set do

if M[i] = = 0 then

rp = encrypt(r, e_i);

M[i] = 1;

Mp = M ;

Union_set = (Union_set - (r, M)}) U {(rp, Mp)};

end if

end for

end for

for (r, M) Є Union_set and (rp, Mp) Є Union_set do

{ check for duplicates }

if r = = rp then

ISSN : 0975-3397

Union_set = Union_set - {(r, M)} {Eliminate duplicate

items before decrypting};

end if

end for

for each site i do {Each site decrypts every item to get

the union of items}

for all (r, M) Є Union_set do

rd = decrypt(r, d_i) ;

Union_set = (Union_set - {(r, M)}) U {(rd)};

end for

permute elements in the Union_set

end for

return Union_set

end.

Secure Size of Set Intersection protocol

Consider several parties having their own sets of items from a common domain. The problem is to securely compute the cardinality/size of the intersection of these local sets. Formally, given k parties $P_1 \ldots P_k$ having local sets $S_1 \ldots S_k$, we wish to securely compute $|S_1 \cap \ldots \cap S_k|$. We can do this is using a parametric commutative one way hash function. One way of getting such a hash function is to use commutative public key encryption, such as Pohlig Hellman, and throw away the decryption keys. Commutative encryption has already been described in previous Section. All k parties locally generate their public key-pair (E_i, D_i) for a commutative encryption scheme. (They can throw away their decryption keys since these will never be used.) Each party encrypts its items with its key and passes it along to the other parties. On receiving a set of (encrypted) items, a party encrypts each item and permutes the order before sending it to the next party. This is repeated until every item has been encrypted by every party. Since encryption is commutative, the resulting values from two different sets will be equal if and only if the original values were the same (i.e., the item was present in both sets). Thus, we need only count the number of values that are present in all of the encrypted itemsets. This can be done by any party. None of the parties is able to know which of the items are present in the intersection set because of the encryption. The complete protocol is shown in algorithm 2 [11].

Algorithm 2 : Securely computing size of intersection set

Require: k sites or parties and each site has a local set S_i

begin

Generate the commutative encryption key-pair (E_i, D_i)

{Throw away the decryption keys, since they will not be

needed.}

M = S_i

for k - 1 steps do

M′ = newarray[|M|]

j=0;

for each X Є M do

M′ [j + +] = encrypt(X, E_i)

end for

permute the array M′ in some random order

send the array M′ to site i + 1 mod k

receive array M from site i - 1 mod k

end for

M′ = newarray[|M|]

j=0;

for each X Є M do

M′ [j + +] = encrypt(X, E_i)

end for

permute the array M′ in some random order

send M′ to site i mod 2 {This prevents a site from seeing

it's own encrypted items}

sites 0 and 1 produce array I_0 and I_1 containing only

(encrypted) items present in all arrays received.

site 1 sends I_1 to site 0

site 0 broadcasts the result $|I_0 \cup I_1|$

end.

IV. INFORMAL ALGORITHM [COMPUTING PRIVACY-PRESERVING C4.5 ALGORITHM]

ISSN : 0975-3397

The previous attempts solved Privacy-preserving classification problem using ID3 algorithm. Our protocol challenges to resolve the problems of previous works by proposing a new scheme for solving the Privacy-preserving classification problem using C4.5 algorithm, shown in figure 2. Our protocol is based on the observation that each node of the tree can be computed separately, with the output made public. Here we are using secure size of set intersection protocol. The computation starts from the root of the tree. Once the attribute of a particular node has been found, all parties can separately partition their remaining records according to the next recursive calls.

V. FORMAL ALGORITHM

Problem definition: There are n parties $P_1,\ldots,P_n$ and each party P_i has a same transaction set T and different necessary attribute set R_i, which take part for classification, R_i where I = 1,…, n. Let R = $R_1 U \ldots U R_n$, each transaction in T contains several general attributes and a class attribute. Let T = $\{T_1,\ldots, T_T\}$ denote the set of transaction and m class values i.e. C = $\{c_1,\ldots, c_m\}$ denote the class attribute, no matter which party hold the class attribute. The n parties want to jointly build a decision tree classifier without revealing their private transaction sets using C4.5 algorithm extension of ID3 algorithm [6].

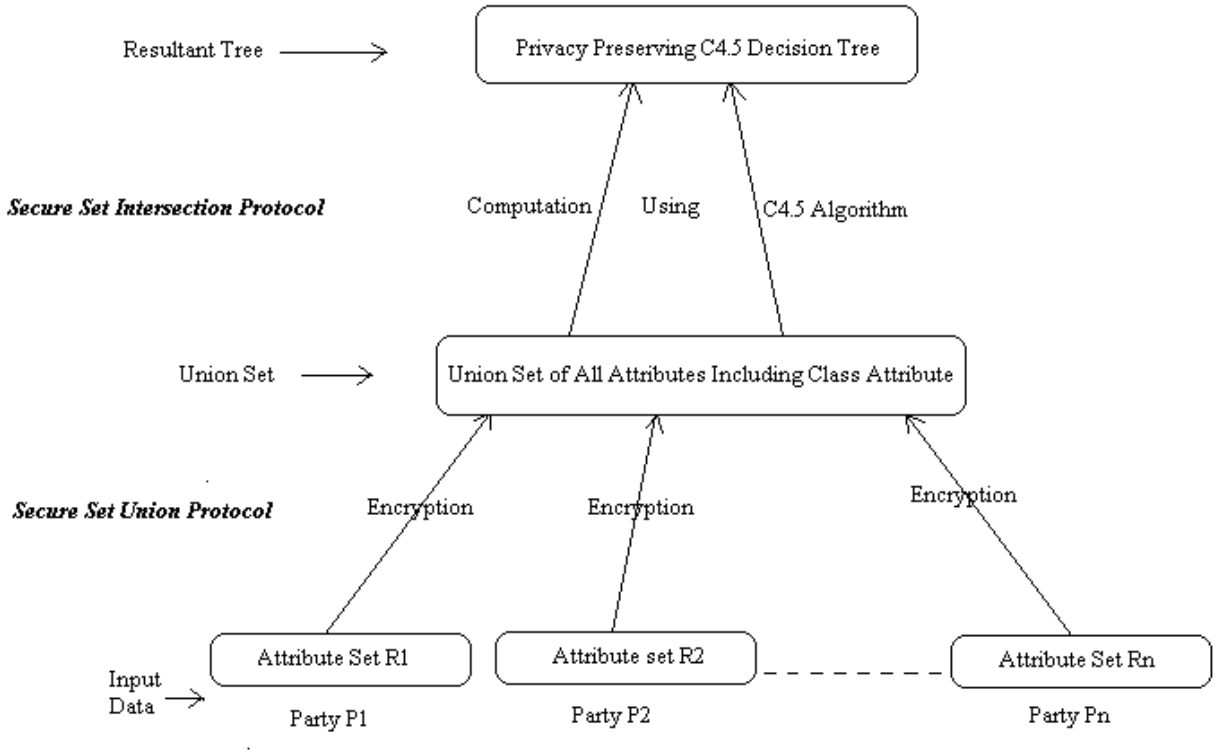

Fig 2. Proposed Privacy-preserving decision tree

Algorithm: PPC4.5(P1: R1,T ; … ; Pn: Rn,C,T)

Require : n sites or parties and each party has T transaction set and no matter which party has the class attribute. After running secure set union protocol gather the set of unique attributes.

1. If R is empty,

then return a leaf-node with the class value assigned to the most transactions in T.

To find the class value with the most transactions in T,

P1,…, Pn conduct the following sub-steps:

a. (cnt1,…, cntm) → Compute class distribution from given current constraints

ISSN : 0975-3397

(Note: Use secure multi-party set intersection protocol)

b. Build a leaf node with (cnt1,…, cntm)

Return the leaf-node with the class attribute.

2. If T contains transactions which all have the same value ci for the class attribute

(Note : we construct a leaf node)

then return a leaf-node with value ci.

3. Otherwise

a. Determine the attribute with the highest information gain used to classify

the transactions in T.

best party ← Assign the site with the particular attribute i.e. best attribute,

having highest information gain

Create interior node Nd with attribute Nd.A ← best attribute of the best party

for each attribute value vali Є Nd.A do

Constraints.set(Nd.A, vali)

nodeId ← PPC4.5()

Nd.vali ← nodeId

{ Add proper branch to interior node}

end for

b. Store node Nd

return node Id of interior node Nd

{ Execution continues at party owning parent node}

endif

VI. CONCLUSION

We believe that it is feasible to construct a Privacy-preserving decision tree classifier that can be used SMC techniques. Further development of the protocol is expected in the sense that for joining multi-party attributes using a trusted third party and an untrusted third party can be used. We are continuing work in this field, both to develop new classifier for building Privacy-preserving decision tree and to analysis new as well as existing classifier for solving different problems i.e. missing attributes etc.

VII. ACKNOWLEDGEMENTS

Please acknowledge teammates or anyone who guided and helped with the paper at the end of the text.

VIII. REFERENCE

[1] J.R. Quinlan, “Induction of decision trees,” In Jude W. Shavlik, Thomas G. Dietterich, (Eds.), Readings in Machine Learning. Morgan Kaufmann, 1990. Originally published in Machine Learning, vol. 1, 1986, pp 81–106.

[2] J.R. Quinlan, “C4.5: Programs for Machine Learning,” Morgan Kaufmann, 1993.

[3] Jiawei Han, Micheline Kamber, “Data Mining: Concepts and Techniques,” Indian Reprint ISBN-81-8147-049-4, Elsevier.

[4] Arun K Pujari, “Data Mining Techniques,” Universities Press (India) 13Th Impression 2007.

[5] Y. Lindell, B. Pinkas, “Privacy preserving data mining,” In Journal of Cryptology vol. 15, no. 3, 2002, pp 177–206.

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

[7] Wenliang Du, Zhijun Zhan, “Building decision tree classifier on private data,” In CRPITS, 2002, pp 1–8.

[8] Nan Zhang, Shengquan Wang, and Wei Zhao, **“**A new scheme on Privacy-preserving data classification,” KDD 2005, Aug 21–24, 2005, Chicago, Illinois, USA.

[9] R. Agrawal, R. Srikant, “Privacy Preserving Data mining,” In proceeding of the ACM SIGMOD on Management of data, Dallas, TX USA, May 15-18, 2000, pp 439-450.

[10] Wenliang Du, Zhijun Zhan, “Using randomized response techniques for privacy-preserving data mining,” In KDD, 2003, pp 505–510.

[11] Verykios V, Bertino E, “State-of-the-art in Privacy preserving Data mining,” SIGMOD, 2004, vol. 33, no. 1.

[12] Jaideep Vaidya, Chris Clifton, “Leveraging the “Multi” in Secure Multi-Party Computation”.

[13] D.K. Mishra, Samiksha Shukla, “Preserving Privacy during Data Mining: A Review,” 8th National Conference on Domestic brilliance to universal excellence: Quest for organizational Success, 2005.

[14] Pinkas B., “Cryptographic techniques for privacy-preserving data mining,” ACM SIGKDD Explorations Newsletter, 2006, vol. 4, no. 2, pp 12-19.

[15] Y. Lindell, B. Pinkas, “Secure Multiparty Computation for Privacy-preserving Data Mining,” In the Journal of Privacy and Confidentiality, 2009, vol. 1, no. 1, pp 59-98.

[16] R. Agrawal, A. Evfimievski, R. Srikant, “Information sharing across private databases,” In SIGMOD, 2003, pp 86–97.

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

[18] J. Shrikant Vaidya, “Privacy preserving data mining over vertically partitioned data,” PH.D Thesis of Purdue University, Aug 2004, pp 28-34.

[19] Weiwei Fang, Bingru Yang, “Privacy Preserving Decision Tree Learning Over Vertically Partitioned Data,” In Proceedings of the International Conference on Computer Science & Software Engineering, 2008.

[20] Wenliang Du, Mikhail J. Attalah, “Secure multi-problem computation problems and their applications: A review and

ISSN : 0975-3397

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.

[21] Cliffton C, Kantarcioglu M, Vaidya J., "Tools for privacy preserving distributed data mining," ACM SIGKDD Explorations Newsletter, 2004, vol. 4, no. 2, pp 28-34.

[22] S. C. Pohlig and M. E. Hellman, "An improved algorithm for computing logarithms over GF (p) and its cryptographic significance," In IEEE Transactions on Information Theory, IT-24, 1978, pp 106-110.

[23] http://www.cs.uregina.ca/~dbd/cs831/noes/m1/dtrees/4_dtrees1.html.

ISSN : 0975-3397