A Fast and Efficient Privacy Preserving Data Mining Over Vertically Partitioned Data

P.S. Annakkodi
Head, Department of Information Technology
Sri Ramalinga Sowdamibai college of Science and Commerce
Coimbatore -641 109

ABSTRACT

The goal of data mining is to extract or "mine" knowledge from large amounts of data. However, data is often collected by several different sites. Privacy, legal and commercial concerns restrict centralized access to this data. Theoretical results from the area of secure multiparty computation in cryptography prove that assuming the existence of trapdoor permutations, one may provide secure protocols for any twoparty computation as well as for any multiparty computation with honest majority. However, the general methods are far too inefficient and impractical for computing complex functions on inputs consisting of large sets of data. What remains open is to come up with a set of techniques to achieve this efficiently within a quantifiable security framework. The distributed data model considered is the heterogeneous database scenario with different features of the same set of data being collected by different sites. This paper argues that it is indeed possible to have efficient and practical techniques for useful privacy-preserving mining of knowledge from large amounts of data. The dissertation presents several privacy preserving data mining algorithms operating over vertically partitioned data. The set of underlying techniques solving independent sub-problems are also presented. Together, these enable the secure "mining" of knowledge.

Keywords

Vertical partitioning, Distributed Data Mining (DDM)

1. INTRODUCTION

The purpose of data mining is to find useful information in the dataset. It is possible to efficiently extract or "mine" knowledge from large amounts of vertically partitioned data within quantifiable security restrictions. Knowledge Discovery in Databases (KDD) is the term used to denote the process of extracting knowledge from large quantities of data. The KDD process assumes that all the data is easily accessible at a central location or through centralized access mechanisms such as federated databases and virtual warehouses. Moreover, advances in information technology and the ubiquity of networked computers have made personal information much more available. The irony is that data mining results rarely violate privacy. The objective of data mining is to generalize across populations, rather than reveal information about individuals. The hitch is that data mining works by evaluating individual data that is subject to privacy concerns. Thus, the true problem is not data mining, but the way data mining is done.

However, the concern among privacy advocates is well founded, as bringing data together to support data mining makes misuse easier. Much of this information has already been collected, however it is held by various organizations. Separation of control and individual safeguards prevent correlation of this information, providing acceptable privacy in practice. However, this separation also makes it difficult to use the information for purposes that would benefit society, such as identifying criminal activity. Proposals to share information across agencies, most recently to combat terrorism, would eliminate the safeguards imposed by separation of the information.

In contrast to the centralized model, the Distributed Data Mining (DDM) model assumes that the data sources are distributed across multiple sites. Algorithms developed within this field address the problem of efficiently getting the mining results from all the data across these distributed sources. Since the primary (if not only) focus is on efficiency, most of the algorithms developed to date do not take security consideration into account. However, they are still useful in framing the context of the paper.

A simple approach to data mining over multiple sources that will not share data is to run existing data mining tools at each site independently and combine the results [7, 8, and 18]. However, this will often fail to give globally valid results. Issues that cause a disparity between local and global results include:

• Values for a single entity may be split across sources. Data mining at individual sites will be unable to detect cross-site correlations.

• The same item may be duplicated at different sites, and will be over-weighted in the results.

• Data at a single site is likely to be from a homogeneous population. Important geographic or demographic distinctions between that population and others cannot be seen on a single site.

Vertical partitioning (a.k.a. heterogeneous distribution) of data implies that though different sites gather information about the same set of entities, they collect different feature sets. In horizontal partitioning (a.k.a. homogeneous distribution), different sites collect the same set of information, but about different entities.

The gold standard for security is the assumption that we have a trusted third party to whom we can give all data. The third

party performs the computation and delivers only the results – except for the third party, it is clear that nobody learns anything not inferable from its own input and the results. The goal of secure protocols is to reach this same level of privacy preservation, without the (potentially insoluble) problem of finding a third party that everyone trusts.

The paper can be arranged as follows : Section II provides the related works involved in privacy, security and data mining. Section III reveals the proposed methodology and section IV gives the experimental results of the proposed work.

2. RELATED WORKS

The basic idea of data perturbation is to alter the data so that real individual data values cannot be recovered, while preserving the utility of the data for statistical summaries. Since the data doesn't reflect the real values of private data, even if a data item is linked to an individual that individual's privacy is not violated. (It is important that such data sets are known to be perturbed, so anyone attempting to misuse the data knows the data cannot be trusted.) This approach has been brought to a high art by the U.S. Census Bureau with the perturbation technique used is data swapping: exchanging data values between records in ways that preserve certain statistics, but destroy real values [12]. An alternative is randomization: Adding noise to data to prevent discovery of the real values. Since the data no longer reflects real-world values, it cannot be (mis)used to violate individual privacy. The challenge is obtaining valid data mining results from the perturbed data.

In [4], Agrawal and Srikant presented the first solution to this problem. Given the distribution of the noise added to the data, and the randomized data set, they were able to reconstruct the distribution (but not actual data values) of the data set. This enabled a data mining algorithm to construct a much more accurate decision tree than mining the randomized data alone, approaching the accuracy of a decision tree constructed on the real data. Other methods for distribution reconstruction have also been developed.

Agrawal and Aggarwal [2] developed an approach based on Expectation Maximization that also gave a better definition of privacy, and an improved algorithm. Evfimievski et al. [10] applied a similar technique to mine association rules. Rizvi and Haritsa [19] consider the case where different item values (0 and 1) have differing privacy requirements.

Polat and Du [17] propose a technique for doing collaborative filtering using randomized perturbation techniques. Solutions for other data mining tasks are certainly feasible. While one will not get the exact same data mining results postrandomization as pre-randomization, the results have been experimentally shown to be accurate enough in the case of both classification [4] and association rule mining [10].

There has been some other work that does not properly fall into either the perturbation or cryptographic categories. Atallah et. al [5] explore the disclosure limitation of sensitive rules. Saygin et al. [20] present a way of using special values, known as "unknowns", to prevent the discovery of association rules. Oliveira and Zaiane [13–16] develop several different methods for association rule mining, clustering and access control for privacy preserving data mining. There has also been extensive work done in statistical databases. This work is outside the scope of this paper, however, Adam and Wortmann [1] provide a good starting point. There has also been extensive work in cryptography creating building blocks, which is also outside the scope of this paper. Many examples can be found in [9].

3. METHODOLOGY

This describes the building block primitives developed as part of this dissertation. The three or more party association rule mining algorithm requires computing the size of the intersection set of local sets. Apart from this, it is an interesting problem in its own right. However, in terms of computation complexity, it is not scalable to the sizes required for data mining (since some of the assumptions used in their analysis no longer hold).

A. Securely Computing the Size of Set Intersection

Assume k > 2 parties, $P_0, \ldots, P_{k-1}$. Each party P_i has set $S_i \subseteq U$ chosen from a common global universe. The problem is to securely compute the size of the intersection set, $|\cap_{i=0}^{k-1} S_i|$.

The key idea behind the algorithm is simple. It is not necessary to have the actual set elements to compute the cardinality of the intersection set. Instead, the parties jointly generate a mapping from U that no party knows in its entirety. The mapping is used to transform the sets S_i , and then the intersection is performed on the transformed sets. Since no party knows the mapping, they cannot reverse the mapping to find the value of any element.

A secure keyed commutative hash function can be used to perform such a mapping, and has other properties that will be useful in proving the security properties of the algorithm.

There are three stages to the protocol: hashing, initial intersection, and final intersection.

- Hashing: In this stage the sets of all the parties are hashed by all parties. Since each party hashes with a key known only to itself, and the order of items is randomly permuted, no other party can determine the mapping performed by the previous party.
- Initial Intersection: In this stage, every party finds the intersection of all sets except its own. The hashing prevents learning the actual values corresponding to the hashed items received. The reason a site does not get its own set is to prevent probing attacks: a site could initially generate a singleton set to probe if that item existed at another site, i.e., if the intersection of its set with thatof another site is empty or of size 1. Aborting prevents probes for sets of size less than r. This also shows the reason that we require k > 2 parties. With two parties, no intersection could be performed without access to the hashed values of one's own set. This prevents the probe detection/prevention.
- Final Intersection: Each party sends the remaining piece of the puzzle to its left neighbor. This enables all parties to compute the final intersection and find the final result, viz. the cardinality of the total intersection set. The collision resistance property of the hash function ensures that no collisions can occur. Thus the algorithm clearly generates the correct result for the size of the intersection set.

A similar technique was used by Agrawal et al. [3] to compute intersection, equijoin, intersection size and equijoin size. However, their technique is limited to two parties and to semi-honest adversaries.

B. A More Efficient Set Intersection Protocol

The symmetric algorithm we have presented in is simple and proven effective at controlling the disclosure of information. Here it present a more complex variant that gives asymptotically improved performance in number of rounds, number of messages, and total number of bits transmitted. It also provides a practical improvement in information disclosure; the same total information is disclosed, but each party only sees a piece of that information.

The key insight behind this protocol is to overlap the hashing and intersection phases. Note that any arbitrary parenthesization of the intersection expression still gives the same result.

$$S_o \bigcap S_1 \bigcap \dots \bigcap S_k$$

$$\equiv$$

$$(\dots(S_o \cap S_1) \cap S_2 \cap \dots \cap S_k)$$

$$\equiv$$

$$(S_o \cap S_1) \cap (S_2 \cap S_3) \cap \dots \cap (S_{k-1} \cap S_k)$$

The second observation is that it is not necessary to hash every set with all keys before intersecting the sets. Any time two items have been hashed by the same set of keys, they can be tested for equality. With careful ordering of the hashing we can perform the innermost intersections early. Repeating this at each level, the intersections can be carried out in the form of a binary tree, reformulating the intersection as

$$\left(\dots\left(\log k\right) - 1 \dots \left(S_o \bigcap S_1\right) \bigcap \left(S_2 \bigcap S_3\right) \bigcap \dots \bigcap \left(S_{k-1} \bigcap S_k\right) \dots \log k \dots\right)$$

Instead of sending sets to be hashed by other sites, a site sends its key. The numbering of the tree ensures that no site sees items hashed with any key it knows (except root, which knows only it own key and sees items hashed with that plus several others.) Thus, in the absence of collusion, sending a key gives the receiver no additional information.

C. Algebraic Method for Computing Dot Product

Consider two real-valued vectors $\vec{X}$ and $\vec{Y}$ of cardinality n, $\vec{X} = (x_1 \dots, x_n)$, $\vec{Y} = (y_1 \dots, y_n)$. The scalar product (or dot product) of $\vec{X}$and $\vec{Y}$ is defined as $\vec{X} \cdot \vec{Y} = \sum_{i=1}^{n} x_i * y_i$ If party A has the vector $\vec{X}$and party B has the vector $\vec{Y}$, securely compute the scalar product $\vec{X} \cdot \vec{Y}$.

Scalar product protocols have been proposed in the Secure Multiparty Computation literature [6], however these cryptographic solutions do not scale well to data mining problems. We give an algebraic solution that hides true values by placing them in equations masked with random values. The knowledge disclosed by these equations only allows computation of private values if one side learns a substantial number of the private values from an outside source. (A different algebraic technique has recently been proposed [11], however it requires at least twice the bitwise communication cost of the method presented here.)

We assume without loss of generality that n is even.

D. Cryptographic Method for Computing Boolean Dot Product

This part presents a purely cryptographic primitive for computing the dot product for Boolean vectors. To be precise, we show how to compute the number of 1s in the logical AND vector of several Boolean vectors.

The entire protocol is quite efficient. P_1 Broadcasts the key E to all other parties. Each party also sends the entire (encrypted) vector to the next party once. P_k Finally sends the encrypted sum back to P_1. Thus the total communication cost is

$$k - 1) * \text{key size} + (k * n + 1) * \text{encrypted msg size} = O(k_n)$$

bits, and $k - 1 + k = 2k - 1$ messages (assuming the entire vector can be sent off as a single message. In terms of computation, every party has to perform n encryptions (one for each bit in its vector), P_k has to perform n multiplications and finally P_1 has to perform 1 decryption to get the final result. Thus, there are a total of k_n encryptions and 1 decryption.

4. EXPERIMENTAL RESULTS

The algorithms developed and build a framework in which privacy preserving data mining can be demonstrated. As a part of the experimental validation, the experimental results on two problems – decision tree classification and association rule mining is done by using Weka database.

A. Weka

To demonstrate real practicality, here implemented the methods as part of an existing and widely used Data Mining toolkit. Weka [91], developed at the University of Waikato in New Zealand, is a collection of machine learning algorithms. Apart from providing algorithms, it is a general implementation framework, along with support classes and documentation. It is extensible and convenient for prototyping purposes. However, the Weka system is a centralized system meant to be used at a single site. It extended the Weka core classes "Instance and Instances" to provide support for distributed instances. A distributed instance consists of only the key identifier and the site identifiers for the sites that together contain the whole instance.

B. Decision Tree Classification

The general model of privacy preserving distributed classification is as follows. The user initiates a request to build a classifier and then request(s) classification of an instance whenever required. The process of building the classifier needs to be co-ordinated so that the data sites locally construct enough state to enable them to jointly satisfy a classification request. To this end, every centralized classification class must be extended with a distributed class that provides the same functionality, however the implementation of these functions/ messages is in a distributed manner.

Increase in the number of parties causes a quadratic expansion in the amount of time required. One of the most important factors affecting the computation time of the protocol is the size of the tree built. Simpler trees are much faster to build. A good thing to note is that once the classification tree is built, classifying an instance takes very little time. Thus, if the

(much more expensive) protocol to build the tree has already been executed, it is an easy (and much less computationally intensive) task to classify any given instance.

The current implementation is multi-threaded and does exploit parallelism to the extent possible. Readily available hardware for encryption or implementation in more highly optimized languages than Java would result in significant improvement. This prototype is meant as a demonstration of the viability and correctness of the protocol.

Table 1: Computation and communication cost of encryption

Number of items Encrypted	Key Size(sec)			Transfer Time(sec)
	256	512	1024	
1k	< 0.0001	5	29	0.0027
10k	10	47	286	0.007
100k	90	467	2827	0.04
1M	900	4660	28762	0.41

C. Association Rule Mining

Using the data generated in the prior table we can easily estimate the extra cost incurred by privacy while doing association rule mining in a particular situation (characterized by the number of transactions, attributes and parties). Table 5.4 estimates the computation cost assuming that the encryption key size is 512 bits.

Both assume that attributes can have at most 100k transactions. We give a worst case scenario estimate assuming that all the attributes are frequent 1-itemsets 128 and also encrypting and communicating the entire attribute. In practice, the cost would be much lower (at least an order of magnitude), since all attributes may not be frequent and even the frequent attributes are present in only a fraction of the total number of transactions.

The cost for other values of key size and communication bandwidth can be easily extrapolated using the data provided above. It is clear from this data that the computation cost greatly exceeds the communication cost. Computation cost can be drastically reduced by optimizing the code (we used the generic variant of GNU gmp), or through widely-available special-purpose encryption hardware. Note that the cost described here is the additional cost of assuring privacy. We still need to compute the association rules at each site. Overall, though expensive, the process is much faster than obtaining necessary approval to release data, assuming such approval could be obtained.

Table 2: Worst-case added computation to achieve privacy

Number of Attributes	Number of Sites				
	2	3	5	10	20
10	9342s	14010s	23350s	46700s	-
50	16hr	20hr	33hr	66hr	132hr
100	28hr	40hr	66hr	132hr	262hr
200	54hr	80hr	132hr	264hr	524hr

5. CONCLUSION

The privacy-preserving data mining over vertically partitioned data is both feasible and practical. Privacy/Security concerns have become an enduring part of society and commerce. It is increasingly necessary to ensure that useful computation does not violate legal/commercial norms for the safety of personal data. The paper demonstrates that Privacy and Data Mining are not inherently in conflict. The major contribution has been to develop solutions for representatives of all of the major data mining tasks: classification, clustering, association rule mining and outlier detection. Some of the tools developed are interesting in and of themselves. They are definitely applicable even beyond the scope of data mining. Here the development of privacy preserving solutions for optimization problems (such as linear programming) by utilizing some of the underlying techniques developed. In the future, some other algorithm can be incorporated for practical problems.

6. REFERENCES

[1] Nabil R. Adam and John C. Wortmann. Security-control methods for statistical databases: A comparative study. ACM Computing Surveys, 21(4):515–556, December 1989.

[2] Dakshi Agrawal and Charu C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In Proceedings of the Twenti-eth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pages 247–255, Santa Barbara, CA, May 21-23 2001. ACM.

[3] Rakesh Agrawal, Alexandre Evfimievski, and Ramakrishnan Srikant. Information sharing across private databases. In Proceedings of ACM SIGMOD Inter-national Conference on Management of Data, San Diego, CA, June 9-12 2003.

[4] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In Proceedings of the 2000 ACM SIGMOD Conference Management of Data, pages 439–450, Dallas, TX, May 14-19 2000. ACM.

[5] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. Verykios. Disclosure limitation of sensitive rules. In Knowledge and Data Engineering Exchange Workshop (KDEX'99), pages 25–32, Chicago, IL, November 8 1999.

[6] Mikhail J. Atallah and Wenliang Du. Secure multi-party computational geometry. In Seventh International Workshop on Algorithms and Data Structures (WADS 2001), Providence, RI, August 8-10 2001.

[7] Philip Chan. An Extensible Meta-Learning Approach for Scalable and Accurate Inductive Learning. PhD paper, Department of Computer Science, Columbia University, New York, NY, 1996.

[8] Philip Chan. On the accuracy of meta-learning for scalable data mining. Journal of Intelligent Information Systems, 8:5–28, 1997.

[9] Wenliang Du. A Study of Several Specific Secure Two-party Computation Prob- lems. PhD paper, Purdue University, West Lafayette, Indiana, 2001.

[10] Alexandre Evfimievski, Ramakrishnan Srikant, Rakesh Agrawal, and Johannes Gehrke. Privacy preserving mining of association rules. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 217–228, Edmonton, Alberta, Canada, July 23-26 2002.

[11] Ioannis Ioannidis, Ananth Grama, and Mikhail Atallah. A secure protocol for computing dot-products in clustered and distributed environments. In The 2002 International Conference on Parallel Processing, Vancouver, British Columbia, August 18-21 2002.

[12] Richard A. Moore, Jr. Controlled data-swapping techniques for masking public use microdata sets. Statistical Research Division Report Series RR 96-04, U.S. Bureau of the Census, Washington, DC., 1996.

[13] Stanley R. M. Oliveira and Osmar R. Zaiane. Foundations for an access control model for privacy preservation in multi-relational association rule mining. In Chris Clifton and Vladimir Estivill-Castro, editors, IEEE ICDM Workshop on Privacy, Security and Data Mining, volume 14 of Conferences in Research and Practice in Information Technology, pages 19–26, Maebashi City, Japan, 2002. ACS.

[14] Stanley R. M. Oliveira and Osmar R. Zaiane. Privacy preserving frequent itemset mining. In Chris Clifton and Vladimir Estivill-Castro, editors, IEEE ICDM Workshop on Privacy, Security and Data Mining, volume 14 of Conferences in Research and Practice in Information Technology, pages 43–54, Maebashi City, Japan, 2002. ACS.

[15] Stanley R. M. Oliveira and Osmar R. Za¨ıane. Privacy preserving clustering by data transformation. In Proceedings of the Eighteenth Brazilian Symposium on Databases, pages 304–318, Manaus, Amazonas, Brazil, October 6-10 2003.

[16] Stanley R. M. Oliveira and Osmar R. Za¨ıane. Protecting sensitive knowledge by data sanitization. In Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03), Melbourne, FL, November 19-22 2003.

[17] Huseyin Polat andWenliang Du. Privacy-preserving collaborative filtering using randomized perturbation techniques. In Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03), pages 625–628, Melbourne, FL, November 19-22 2003.

[18] Andreas Prodromidis, Philip Chan, and Salvatore Stolfo. Advances in Dis- tributed and Parallel Knowledge Discovery, chapter 3: Meta-learning in distributed data mining systems: Issues and approaches. AAAI/MIT Press, September 2000.

[19] Shariq J. Rizvi and Jayant R. Haritsa. Maintaining data privacy in association rule mining. In Proceedings of Twenty-eighth International Conference on Very Large Data Bases, pages 682–693, Hong Kong, August 20-23 2002. VLDB.

[20] Y¨ucel Saygin, Vassilios S. Verykios, and Chris Clifton. Using unknowns to prevent discovery of association rules. SIGMOD Record, 30(4):45–54, December 2001.

[21] Ian H. Witten and Eibe Frank. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, October 1999.