Privacy Preserving Distributed Data Mining

Chris Clifton
Department of Computer Sciences

November 9, 2001

Data mining technology has emerged as a means for identifying patterns and trends from large quantities of data. Data mining and data warehousing go hand-in-hand: most tools operate on a principal of gathering all data into a central site, then running an algorithm against that data (Figure 1). There are a number of applications that are infeasible under such a methodology, leading to a need for distributed data mining. The obvious solution of a "virtual" data warehouse – heterogeneous access to all the data – is not always possible. The problem is not simply that the data is distributed, but that it must be distributed. There are several situations where this arises:

1. Connectivity. Transmitting large quantities of data to a central site may be infeasible.
2. Heterogeneity of sources. Is it easier to combine results than combine sources?
3. Privacy of sources. Organizations may be willing to share data mining results, but not data.

This research will concentrate on issue 3: obtaining data mining results that are valid across a distributed data set, with limited willingness to share data between sites. We propose to perform local operations on each site that produce intermediate data that can be used to obtain the results, without revealing the private information at each site.

There are many variants of this problem, depending on how the data is distributed, what type of data mining we wish to do, and what restrictions are placed on sharing of information. Some problems are quite tractable, others are more difficult. For example, if we are trying to learn association rules with support and confidence thresholds, a common data mining problem, there is a simple distributed solution that provides a degree of privacy to the individual sites. An example association rule could be:

> *Received Flu shot* and $age > 50$ implies *hospital admission*, where at least 5% of insured meet all the criteria (support), and at least 30% of those meeting the *flu shot* and *age* criteria actually require hospitalization (confidence).

There are algorithms to efficiently find all association rules with a minimum level of support. We can easily extend this to the distributed case using the following lemma: If a rule has $support > k\%$

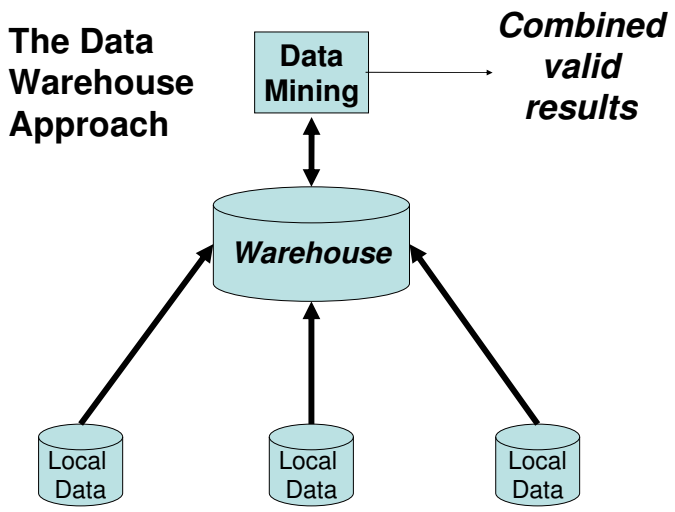

Figure 1: Data Warehouse approach to Distributed Data Mining

globally, it must have *support* $> k\%$ on at least one of the individual sites.[1] A distributed algorithm for this would work as follows: Request that each site send all rules with support at least k. For each rule returned, request that all sites send the count of items they have that support the rule, and the total count of all items at the site. From this, we can compute the global support of each rule, and (from the lemma) be certain that all rules with support at least k have been found. An example of how this works is shown in Figure 2.

This is straightforward, but as we vary the problem the challenge becomes more difficult. What if we want to protect not only the individual items at each site, but also how much each site supports a given rule? The above method reveals this information. Another variant where this approach fails is when the data is partitioned vertically: a single item may have part of it's information at one site, and part at another. We are building a research program that will address a broad spectrum of data mining and privacy issues.

We propose to use Purdue Research Foundation funding to address the problem of association rule discovery, where data is vertically partitioned, and privacy means preventing others from learning the value of "private" attribute values for each entity.

Related Work

Why is there research to be done here? What happens if we run existing data mining tools at each site independently, then combine the results? This will not generally give globally valid results.

[1]For proof of this, assume all the data is together, and divide data items into those that support the rule and those that don't. Now try to partition data such that no site has *support* $> k\%$. For each supporting item sent to a site, at least $1/k$ non-supporting items need to be send to that site. It can be seen that we will run out of non-supporting items before assigning all the supporting items to sites.

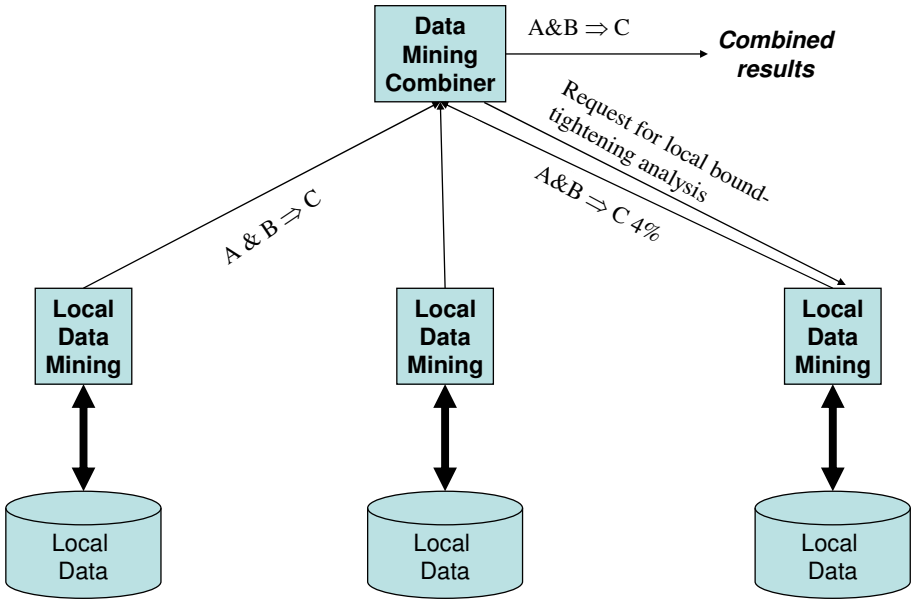

Figure 2: Example of computing association rules from individual site results.

Situations 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 overweighted 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.

Data mining algorithms that partition the data into subsets have been developed[SON95]. In particular, work in parallel data mining that may be relevant [Zak99, KC00]. Although the goal of parallelizing data mining algorithms is performance, the communication cost between nodes is an issue. Parallel data mining algorithms may serve as a starting point for portions of this research.

Algorithms have been proposed for distributed data mining. Cheung et al. proposed a method for horizontally partitioned data[CNFF96], this is basically the approach outlined in the Figure 2. Distributed classification has also been addressed. A meta-learning approach has been developed that uses classifiers trained at individual to develop a global classifier [Cha96, Cha97, PCS00]. This *could* protect the individual entities, but it remains to be shown that the individual classifiers do not release private information. Recent work has addressed classification in vertically partitioned

data [CSK01], and situations where the distribution is itself interesting with respect to what is learned [WBH01]. However, none of this work addresses *privacy* concerns.

There has been research considering how much information can be inferred, calculated or revealed from the data made available through data mining algorithms, and how to minimize the leakage of information [LP00, AS00]. However, this has been restricted to classification. The problem has been treated with an "all or nothing" approach. We desire quantification of the security of the process. Corporations may not require absolute zero knowledge protocols (that leak no information at all) as long as they can keep the information shared within strict (though possibly adjustable) bounds.

There has been work in cooperative computation between entities that mutually distrust one another. This computation may be of any sort: scientific, data processing or even secret sharing. Secure two party computation was first investigated by Yao [Yao86] and was later generalized to multiparty computation. The seminal paper by Goldreich proves that there exists a secure solution for *any* functionality[GMW87]. The approach used is as follows: the function F to be computed is first represented as a combinatorial circuit, and then the parties run a short protocol for every gate in the circuit. Every participant gets corresponding shares of the input wires and the output wires for every gate. This approach, though appealing in its generality and simplicity, means that the size of the protocol depends on the size of the circuit, which depends on the size of the input. This is highly inefficient for large inputs, as in data mining. Although this shows that secure solutions exist, achieving *efficient* secure solutions for distributed data mining is still open.

Significance

There are a number of scenarios where these issues arise. A few (as possible "challenge problems") are:

- Identifying public health problem outbreaks (e.g., epidemics, biological warfare instances). There are many data collectors (insurance companies, HMOs, public health agencies). Individual privacy concerns will limit the willingness of the data custodians to share data, even with government agencies such as the Centers for Disease Control. Can we accomplish the desired results while still preserving privacy of individual entities?

- Collaborative corporations or entities. Ford and Firestone shared a problem with a jointly produced product: Ford Explorers with Firestone tires. Ford and Firestone may have been able to use association rule techniques to discover problems earlier. This would have required extensive data sharing. Factors such as trade secrets and agreements with other manufacturers stand in the way of the necessary sharing. Could we obtain the same results, while still

preserving the secrecy of each side's data?

Government entities face similar problems, such as limitations on sharing between law enforcement, intelligence agencies, and tax collection.

- Multi-national corporations. An individual country's legal system may prevent sharing of customer data between a subsidiary and it's parent.

These examples each define a different problem, or even set of problems. Although we plan to address only the problem faced in the second example above with this funding, we expect this to seed a larger research effort that will address a wide variety of distributed data mining problems. The problems can be characterized by the following three parameters:

1. What is the desired data mining outcome? Do we want to *cluster* the data, as in the disease outbreak example? Are we looking for *association rules* identifying relationships among the attributes? There are several such data mining tasks, and each poses a new set of challenges.

2. How is the data distributed? Is each entity found only at a single site (as with medical insurance records)? Or do different sites contain different types of data (Ford on vehicles, Firestone on tires)?

3. What are the privacy requirements? If the concern is solely that values associated with an individual entity not be released (e.g., "personally identifiable information") we will develop techniques that provably protect such information. In other cases, the notion of "sensitive" may not be known in advance. This would lead to human vetting of the intermediate results before sharing, requiring that individual site results be:

   - compact;
   - human-understandable; and
   - complete (i.e., the method must not require numerous rounds of communication, each requiring human vetting.)

Sometimes it may be difficult (or impossible) to develop an exact solution that meets the privacy constraints. In data mining an approximate solution is often sufficient. The goal, then, is to obtain a solution with *bounded* error, or difference from the "data warehouse" solution.

These solutions can be applied to problems other than privacy. Mining heterogeneous databases is one such problem – combining results may be easier than combining sources. For example, one database has plane_type $\in$ (fixed_wing, rotorcraft), and a model_number field. Another uses plane_type to mean the model number. If both databases produce a rule "MD80 and Total_Time>50,000 $\rightarrow$ jack_screw replacement", even though the field for MD80 is different in the

two databases, it would be easy to manually discover that model_number ≡ plane_type (at least in the context of this rule).

Research Plan

This research will have several steps, culminating in the production of a prototype system demonstrating privacy-preserving distributed data mining. The initial phase will formally define the problem, and develop an efficient algorithm to solve the problem. Note that this may be an iterative process: As we develop solutions, we may find that we can better solve alternative definitions of the problem. These alternatives will be measured against the practical impact: Are we defining the problem in a way that solves real-world problems?

Working with Jaideep Vaidya (nominated for support under this grant) we have developed a method that enables discovery of association rules in vertically partitioned data, while still preserving privacy of individual items. This method requires $O(n)$ communication cost (where n is the number of items in the database), but only two rounds of communication. This method is outlined in Appendix A.

We are looking for alternative approaches that give cost related to the support of the rule, rather than the size of the database – this is likely to provide a more practical algorithm. We expect to develop solutions to several variants of the problem over the course of the project.

Once a solution is discovered, the next step is to formally prove the solution preserves the privacy of the data items. The method we describe works by adding random values to what is transmitted, giving n shared equations in more than n unknowns. We are able to recover the desired results from the equations. However, since there are multiple solution to the equations, recovery of the original n unknown data items is prevented. Although this is convincing, it remains to be formally proven that no recovery of the data items is possible.

Beyond this, we must develop a complete association rule mining algorithm. The method in Appendix A computes a critical figure needed for association rule mining in a secure way. We will incorporate this in a complete algorithm, and demonstrate that the full algorithm still preserves the privacy of individual data items.

Another issue to be resolved is the complexity of the algorithm. In distributed computing, we measure the number of messages, size of data transmitted, and number of communication rounds required by an algorithm. In database and data mining, complexity is generally thought of in terms of the number of data items accessed, and the memory size required. We will evaluate both sets of cost metrics. Many data mining techniques have intractable worst-case performance, but good performance in practical cases. Rather than traditional worst-case analysis, we will analyze our algorithms in comparison with existing centralized data mining algorithms (e.g., "3 messages for

each computational round in the a-priori algorithm[AS94]".)

To truly validate the efficacy and practicality of the method, we will implement and test the algorithms. This will require development of a prototype system to enable testing on a distributed platform. Computing facilities to support this testing already exist within the Indiana Center for Database Systems.

A Scalar Product Algorithm for Privacy-Preserving Association Rule Mining

Problem Definition

We consider mining of boolean association rules. The presence or absence of an attribute is represented as a 1 or 0 respectively. Thus items in the database (transactions) look like strings of 0 and 1. The entire database can be represented as a matrix of {0,1}.

To find out if a particular itemset is frequent, count the number of records where the values for all attributes in the itemset are 1. This translates to a simple mathematical problem, given the following definitions:

Let the total number of attributes be $l + m$, where A has l attributes, and B has the remaining m attributes. Thus A has the values for the attributes A_1 through A_l, and B has the values for the m attributes, B_1 through B_m.

k is the support threshold required.

n is the total number of transaction/records.

Transactions/records are a sequence of $l + m$ 1s or 0s.

Let the $\vec{X}$ and $\vec{Y}$ represent the vectors of processed data values at each site, A and B respectively. We describe how to compute the $\vec{X}$ and $\vec{Y}$ vectors later.

The scalar (dot) product of two vectors $\vec{X}$ and $\vec{Y}$ of cardinality n is defined as

$$\vec{X}.\vec{Y} = \sum_{i=1}^{n} x_i * y_i$$

To compute the frequency of a 2-itemset where one of the attributes is known to A while the other is known to B, the vectors $\vec{X}$ and $\vec{Y}$ are same as the attribute vectors with A and B. Computing whether an itemset is frequent translates to checking if the number of transactions in which all attributes present in the itemset are present is greater than the support threshold, k.

Since we represent absence or presence of an attribute as 0 and 1, this translates to computing $\sum_{i=1}^{n} x_i * y_i$ where the x_i and y_i are the attribute values. *Thus, this translates to calculating the scalar dot product of the 2 attributes.*

We present an efficient way to do this securely when both of the parties possess one of the attributes and wish to limit the information revealed in the section on the component algorithm.

This was a description of the protocol for a 2-itemset. The generalization of this protocol to a w-itemset is straightforward.

To find association rules, then, go through the following steps:

1. L_1 = {large 1-itemsets}
2. for (k=2; $L_{k-1} \neq \phi$; k++)
3. C_k = apriori-gen(L_{k-1});
4. for all candidates $c \in C_k$ do begin

5. if all the attributes in c are entirely at A or B, that party independently calculates if the itemset is frequent
6. else
7. let A have l of the attributes and B have the remaining m attributes
8. construct $\vec{X}$ on A's side and $\vec{Y}$ on B's side where $\vec{X} = \prod_{i=1}^{l} \vec{A}_i$ and $\vec{Y} = \prod_{i=1}^{m} \vec{B}_i$
9. compute $c.count = \vec{X}.\vec{Y} = \sum_{i=1}^{n} x_i * y_i$
10. $L_k = L_k \cup c | c.count \geq minsup$
11. end
12. end
13. Answer $= \cup_k L_k$

The component algorithm

The scalar product of two vectors $\vec{X} = (x_1, \cdots, x_n)$ and $\vec{Y} = (y_1, \cdots, y_n)$ is defined as $\vec{X}.\vec{Y} = \sum_{i=1}^{n} x_i * y_i$. Computation of the scalar product forms a large part of our protocol. Thus, it is important to have an efficient (low communication cost) algorithm that computes the scalar product. We now present such an algorithm.[2]

A generates $n/2$ randoms $R_1 \ldots R_{n/2}$

A now computes the following n values:

$$\begin{aligned}
&\langle x_1 + R_1 \rangle \\
&\langle x_2 + R_2 \rangle \\
&\vdots \\
&\langle x_{n/2} + R_{n/2} \rangle \\
&\langle x_{n/2+1} + a_{1,1} * R_1 + a_{1,2} * R_2 + \cdots + a_{1,n/2} * R_{n/2} \rangle \\
&\langle x_{n/2+2} + a_{2,1} * R_1 + a_{2,2} * R_2 + \cdots + a_{2,n/2} * R_{n/2} \rangle \\
&\vdots \\
&\langle x_n + a_{n/2,1} * R_1 + a_{n/2,2} * R_2 + \cdots + a_{n/2,n/2} * R_{n/2} \rangle
\end{aligned}$$

All the $(\frac{n}{2})^2$ a values, are known to both A and B. The only constraint is that the $a_{i,j}$ should be coefficients for a set of linear *independent* equations.

A sends all n values to B. B multiplies each value he gets with the corresponding y value, and adds all of them up to get a sum, S.

B now sends to A, S and the following $n/2$ values:

$$\begin{aligned}
&\langle y_1 + a_{1,1} * y_{n/2+1} + a_{2,1} * y_{n/2+2} + \cdots + a_{n/2,1} * y_n \rangle \\
&\langle y_2 + a_{1,2} * y_{n/2+1} + a_{2,2} * y_{n/2+2} + \cdots + a_{n/2,2} * y_n \rangle \\
&\vdots \\
&\langle y_{n/2} + a_{1,n/2} * y_{n/2+1} + a_{2,n/2} * y_{n/2+2} + \cdots + a_{n/2,n/2} * y_n \rangle
\end{aligned}$$

A can write S as follows:

$$S =$$

[2] We assume that n is even for simplifying the protocol. The protocol can easily be generalized to the case where n is odd as well.

$$
\begin{aligned}
&[y_1 * \{x_1 + (R_1)\} \\
&+y_2 * \{x_2 + (R_2)\} \\
&\vdots \\
&+y_{n/2} * \{x_{n/2} + (R_{n/2})\} \\
&+y_{n/2+1} * \{x_{n/2+1} + (a_{1,1} * R_1 + a_{1,2} * R_2 + \cdots + a_{1,n/2} * R_{n/2})\} \\
&+y_{n/2+2} * \{x_{n/2+2} + (a_{2,1} * R_1 + a_{2,2} * R_2 + \cdots + a_{2,n/2} * R_{n/2})\} \\
&\vdots \\
&+y_n * \{x_n + (a_{n/2,1} * R_1 + a_{n/2,2} * R_2 + \cdots + a_{n/2,n/2} * R_{n/2})\}\Big]
\end{aligned}
$$

Simplifying the equation further, and grouping the $x_i * y_i$ terms, we get:

$$
\begin{aligned}
S = \\
&(x_1 * y_1 + x_2 * y_2 + \cdots + x_n * y_n) \\
&+(R_1 * y_1 + R_2 * y_2 + \cdots + R_{n/2} * y_{n/2}) \\
&+(a_{1,1} * R_1 * y_{n/2+1} + a_{1,2} * R_2 * y_{n/2+1} + \cdots + a_{1,n/2} * R_{n/2} * y_{n/2+1}) \\
&+(a_{2,1} * R_1 * y_{n/2+2} + a_{2,2} * R_2 * y_{n/2+2} + \cdots + a_{2,n/2} * R_{n/2} * y_{n/2+2}) \\
&\vdots \\
&+(a_{n/2,1} * R_1 * y_n + a_{n/2,2} * R_2 * y_n + \cdots + a_{n/2,n/2} * R_{n/2} * y_n)
\end{aligned}
$$

The first line of the R.H.S. can be succinctly written as $\sum_{i=1}^{n} x_i * y_i$, the desired final result. In the rest, we can group all multiplicative components vertically, and rearrange the equation to factor out all the R_i values, to get:

$$
\begin{aligned}
S = \\
&\sum_{i=1}^{n} x_i * y_i \\
&+R_1 * (y_1 + a_{1,1} * y_{n/2+1} + a_{2,1} * y_{n/2+2} + \cdots + a_{n/2,1} * y_n) \\
&+R_2 * (y_2 + a_{1,2} * y_{n/2+1} + a_{2,2} * y_{n/2+2} + \cdots + a_{n/2,2} * y_n) \\
&\vdots \\
&+R_{n/2} * (y_{n/2} + a_{1,n/2} * y_{n/2+1} + a_{2,n/2} * y_{n/2+2} + \cdots + a_{n/2,n/2} * y_n)
\end{aligned}
$$

Now, to get the desired result (viz. $\sum_{i=1}^{n} x_i * y_i$), A needs to only peel off the remaining baggage. A already knows the $n/2$ R_i values. Recall that B sent him $n/2$ other values earlier that are exactly the same as the coefficients of the $n/2$ R_i values. Thus A can easily multiply the $n/2$ values he got from B with the corresponding R_i, add it all up, and subtract the total from S to get the desired result. In this way, A and B are able to cooperatively get the total number of transactions containing both attributes without revealing any details about any particular transaction.

References

[AS94] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*,

Santiago, Chile, September 12-15 1994. VLDB.

[AS00] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proceedings of the 1997 ACM SIGMOD Conference on Management of Data*, Dallas, TX, May 14-19 2000. ACM.

[Cha96] Philip Chan. *An Extensible Meta-Learning Approach for Scalable and Accurate Inductive Learning*. PhD thesis, Department of Computer Science, Columbia University, New York, NY, 1996. (Technical Report CUCS-044-96).

[Cha97] Philip Chan. On the accuracy of meta-learning for scalable data mining. *Journal of Intelligent Information Systems*, 8:5–28, 1997.

[CNFF96] David Wai-Lok Cheung, Vincent Ng, Ada Wai-Chee Fu, and Yongjian Fu. Efficient mining of association rules in distributed databases. *Transactions on Knowledge and Data Engineering*, 8(6):911–922, December 1996.

[CSK01] Rong Chen, Krishnamoorthy Sivakumar, and Hillol Kargupta. Distributed web mining using bayesian networks from multiple data streams. In *The 2001 IEEE International Conference on Data Mining*. IEEE, November 29 - December 2 2001.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game - a completeness theorem for protocols with honest majority. In *19th ACM Symposium on the Theory of Computing*, pages 218–229, 1987.

[KC00] H. Kargupta and P. Chan, editors. *Advances in Distributed and Parallel Knowledge Discovery*. AAAI/MIT Press, 2000.

[LP00] Yehuda Lindell and Benny Pinkas. Privacy preserving data mining. In *Advances in Cryptology – CRYPTO 2000*, pages 36–54. Springer-Verlag, August 20-24 2000.

[PCS00] Andreas Prodromidis, Philip Chan, and Salvatore Stolfo. *Meta-learning in distributed data mining systems: Issues and approaches*, chapter 3. AAAI/MIT Press, 2000.

[SON95] Ashoka Savasere, Edward Omiecinski, and Shamkant B. Navathe. An efficient algorithm for mining association rules in large databases. In *Proceedings of 21th International Conference on Very Large Data Bases*, pages 432–444. VLDB, September 11-15 1995.

[WBH01] Rüdiger Wirth, Michael Borth, and Jochen Hipp. When distribution is part of the semantics: A new problem class for distributed knowledge discovery. In *Ubiquitous Data Mining for Mobile and Distributed Environments workshop associated with the Joint 12th European Conference on Machine Learning (ECML'01) and 5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'01)*, Freiburg, Germany, September3-7 2001.

[Yao86] Andrew C. Yao. How to generate and exchange secrets. In *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, pages 162–167. IEEE, 1986.

[Zak99] Mohammed J. Zaki. Parallel and distributed association mining: A survey. *IEEE Concurrency, special issue on Parallel Mechanisms for Data Mining*, 7(4):14–25, December 1999.