Privacy-Enhancing Distributed Higher-Order ARM

Aleksandar Nikolov
Rutgers University
anikolov@cs.rutgers.edu

Shenzhi Li
Rutgers University
lshenzhi@dimacs.rutgers.edu

William M. Pottenger
Rutgers University
drwmp@cs.rutgers.edu

Abstract

Traditional association rule mining algorithms assume that data instances are independent and identically distributed (IID) [29]. In statistical relational learning (SRL), however, relationships between instances can be leveraged to improve performance of learning algorithms [2]. Higher-order association rule mining is an example of a SRL approach that does not make the IID assumption, but instead discovers itemsets that cross record boundaries [21]. Empirical analysis shows that higher-order methods perform especially well on small datasets as they are able to capture the variability of the underlying data distribution more readily than traditional methods [11]. In a distributed environment, however the discovery of higher-order itemsets reveals significant information about the nature of disparate data sources [21]. Preserving privacy in a setting in which data instances are treated as nodes in a graph rather than independent entities is an open problem in privacy research that has only recently received attention in the data mining community [24]. In this paper we propose a novel privacy-enhancing distributed higher-order ARM algorithm, PE-DiHO ARM. PE-DiHO ARM discovers itemsets from distributed data with a hybrid (non-horizontal, non-vertical) distribution while significantly limiting the amount of private data that is revealed. To demonstrate the validity of the approach we compare it to a non-privacy enhancing higher-order ARM algorithm [21] in an evaluation framework based on supervised learning [23]. Experimental results confirm that privacy can be significantly enhanced during the computation of higher-order itemsets in a distributed environment without significantly impacting performance. In future work we plan to apply these techniques to data provided by our law enforcement partners.

1 Introduction

There are numerous situations in which not all data can be stored in a central site due to storage constraints or privacy concerns. This is an especially important problem for the law enforcement domain where there are numerous legal restrictions on the sharing of information, while at the same time records for the same individual or legal entity may exist in the databases of multiple jurisdictions. Recent work in distributed data mining [4] [36] and privacy-enhancing data mining [1] [8] [18] [19] [22] [31] [32] [33] has addressed mining distributed data of this nature. Naturally, the solutions proposed so far also have limitations. Existing approaches, for example, deal with either horizontally or vertically fragmented data and assume the existence of a global schema. In a real-world application, on the other hand, often no global schema is available. In addition, because different sites maintain different schemas for storing data, it is often impossible to guarantee purely vertical or horizontal fragmentation. This is especially the case when dealing with information extracted from textual data sources in the form of named entities stored in database records.

Another limitation in current solutions is the need for approaches to enhancing privacy for data mining algorithms which do not make the assumption that data instances are independent and identically distributed (IID). This issue is related to the problem of enhancing privacy in statistical relational learning, which exploits similarity links between data instances [12]. It is also related to enhancing privacy when the input data is modeled as a graph – a problem which has only recently attracted the attention of researchers [24]. While traditional association rule mining algorithms assume there are no correlations between instances [29], in contrast a relational algorithm exploits links between instances. This allows a dataset to be treated as a graph in which data instances are nodes and links are edges. In [21] for example, Li et al. present DiHO ARM, a relational distributed association rule mining algorithm which exploits not only information within records but also links between records. We refer to these links as *higher-order* because they form the basis for itemsets that cross record boundaries. Recent work in our lab shows that such higher-order information captures the variability of the underlying data more quickly than equivalent methods which make the IID assumption. For this reason, higher-order algorithms are especially useful for small datasets [11].

Indeed, manual higher-order approaches are standard operating procedure in real-world law enforcement operations. In 2003, for example, the DEA and the Royal Canadian Mounted Police announced the arrests of over 65 individuals in ten cities throughout the United States and

Canada in an international methamphetamine investigation [39]. The arrests were the result of an 18-month international investigation using manual higher-order association techniques that linked documents through addresses, phone numbers, etc. In effect, individuals who appear in different documents can be linked as partners in crime.

Motivated by the evident utility of higher-order algorithms for law enforcement, the authors are involved in an ongoing collaboration with several law enforcement agencies, including the Bethlehem, PA Police Department (BPD), the Public Safety Department of the Port Authority of New York and New Jersey and the Richmond, VA Police Department. In conjunction with the BPD, we have collected hundreds of records from a high-profile murder case. We are using the collection to establish a secure, anonymized, annotated ground truth data repository for use in evaluating different algorithms using higher order links. We are also working closely with the Port Authority Public Safety Department to provide advanced data and visual analytics capabilities in support of the twin mission of fighting crime and preventing terrorism.

The DiHO ARM algorithm, however, is not private and thus is not applicable in situations where privacy restrictions apply, and, in particular, in the law enforcement domain. Preserving privacy with higher-order association rules is inherently more difficult than traditional privacy-enhancing association rule mining. Intuitively, in the higher-order case the algorithm leverages more information from the dataset [2], and, hence, more information needs to be privately exchanged between the parties in the computation. Also, because records cannot be treated independently, many of the computations become more complex.

In this article we present PE-DiHO ARM, a privacy-enhancing algorithm for mining higher-order itemsets in a distributed environment. The algorithm is privacy-enhancing in the sense that it significantly reduces the amount of private data revealed during the computation. Like the original DiHO ARM algorithm [21], it can deal with data which is neither horizontally nor vertically distributed. Another contribution of PE-DiHO ARM is a method for privacy-enhancing construction of a graph from distributed data. We also address the problem of privacy-enhancing path generation when the total number of nodes in the distributed graph is unknown and the path length is limited.

The article is organized as follows. In the following section we describe related work in the field of privacy-enhancing data mining. In Section 3 we present our approach to enhancing the privacy of higher-order itemset mining in a distributed environment, including an example run as well as an explanation of our methodology of evaluation. Section 4 presents experimental evidence for the validity of PE-DiHO ARM, followed by Conclusions and Future Work in Section 5.

2 Related Work

There has been a considerable amount of work on privacy-enhancing techniques for data mining. Research in this field naturally falls into two categories – data perturbation approaches and secure multi-party computation approaches. There is some very interesting recent work in the latter area dealing with graph data. We consider this in Section 2.2.1 below.

2.1 Perturbation Approaches. The data perturbation perspective assumes that not even the data mining algorithm is allowed access to the private data. The general method used in data perturbation algorithms follows three basic steps:

1. Private data is randomized, so that individual data values cannot be easily estimated.
2. The original distribution of the private data is reconstructed using information about the randomization technique used.
3. Analysis is performed on the reconstructed distribution.

It is important to note that only the distribution of the original data is reconstructed. The actual data values remain inaccessible. It should also be mentioned that data mining methods which use perturbation are not exact, i.e. they give only approximations of the rules that would be discovered from the original data.

In [1], Agrawal and Srikant propose a data perturbation approach to decision tree learning with numeric data. The original private values are perturbed by adding a random number from some distribution. In [20] the utility of additive noise for privacy applications is questioned and it is pointed out that the noise can be easily filtered out. Hence, [28] and [25] propose stronger methods based on multiplication by a random matrix. Nevertheless, techniques such as independent component analysis can still make these perturbation methods vulnerable.

In [8], Evfimievski et al. further develop the randomization method of that proposed by Agrawal et al. [1] and generalize it to apply to categorical data and association rule mining in particular. They propose a mathematical model of a privacy breach as the probability of discovering a property of the original data based on the randomized data. Then they define a class of randomization operators which apply the same randomization algorithms to each transaction in the database being mined and do not use item-specific information.

Data perturbation approaches have the advantage of usually being more efficient than cryptographic techniques. However, the privacy guarantees they offer are not as strict [33].

2.2 Secure multiparty computation.

The secure multi-party computation perspective on enhancing privacy assumes a distributed environment in which a number of parties want to perform a data mining task on their combined data but no party wants to compromise its privacy. Secure multi-party computation approaches are not concerned with hiding data from the data mining algorithm. These methods are exact: they usually derive precisely the same models of the data a non-privacy preserving algorithm would produce.

Secure multi-party computation itself is a cryptographic technique which allows multiple parties to compute a function without revealing anything but the final output. While there are general techniques to achieve this [14] [34], their computational and communication complexity for large inputs is usually prohibitive for practical use in data mining applications. Therefore, specific techniques tailored to data mining and data analysis algorithms are needed.

2.2.1 Privacy-preserving analysis of graph data.

Some very recent work that is highly relevant to our research seeks to apply the secure multi-party computation paradigm to graph data. Treating data instances as nodes in a graph inherently violates the IID assumption and poses new challenges for privacy researchers. These are the same challenges that we face with higher-order information.

There has been some recent work addressing the problem of preventing re-identification of nodes or edges in an anonymized social network [15], [37]. This work, however, is orthogonal to the problems we address.

In [3] Brickell and Shmatikov propose privacy-preserving algorithms based on secure multiparty computation for several classical graph problems: all-pairs shortest distance, all-pairs shortest path, and single source shortest distance. Their algorithms are designed to work with two parties who share the set of nodes and the set of edges of a graph, but associate different weights with the edges. The protocols do not extend to a setting in which nodes and/or edges are partitioned among sites participating in the computation.

An online version of the popular link analysis algorithm HITS is developed in [6]. One goal of the method is to address link analysis in a setting in which the graph structure is implicit in a set of documents and not explicitly defined. The authors also propose a privacy-enhancing version of their algorithm, based on evaluating dot products using homomorphic encryptions. He et al. follow a similar approach in [16]. They reduce the problem of link discovery to finding the transitive closure of a distributed graph by raising the distributed adjacency matrix to an appropriate power. The lengths of paths are ignored. Both [16] and [6] assume that the set of nodes of a directed graph is partitioned among k sites and each site knows the total number n of nodes in the graph and all edges to and from its local nodes. Without these assumptions, a matrix-based approach is not feasible. Note, however, that the total number of nodes can give significant information about other users' data when the number of participating sites is small.

While in [6] the authors recognize both the usefulness of analyzing a graph constructed from document data and the need for doing this in a privacy-enhancing way, they do not address the issue of securely detecting links between documents belonging to two different sites. To the best of our knowledge, to-date the problem of privacy-enhancing construction of a graph from distributed data has not been addressed in the literature.

2.2.2 Traditional privacy-preserving data mining.

The study of privately building and analyzing graph models of data is still in its infancy. However, it can benefit from substantial prior work in preserving privacy with traditional data mining algorithms using secure multi-party computation. We present some of the results in this field that are relevant to our work.

In [18], Kantarcioglu et al. propose a secure multi-party computation approach to association rule mining from horizontally distributed data. The authors assume parties in the computation follow the semi-honest model [13]. In this model, each party is unwilling to share its own data, but agrees to follow a common communication protocol. Furthermore, each party is free to use what it sees during execution of the protocol to compromise security. The protocol proposed in [18] requires every party to encrypt its itemsets using a commutative encryption algorithm. Itemsets are then all aggregated in two phases: first at two sites and then at one site, in order to minimize the disclosure of private information about the number of common itemsets between sites. Then for each itemset each site locally computes the difference between the support of the itemset and the minimum support threshold. Finally, large itemsets are determined using Yao's general secure two-party evaluation [34].

A different method is necessary for privacy-enhancing distributed association rule mining in vertically partitioned data. One such method is offered by Vaidya and Clifton in [32]. They consider the two-party Boolean case and model each attribute, or column in the distributed database, as a binary vector; then, the support of an itemset is computed as the dot product of the vectors corresponding to each item in the itemset. In this model the problem of finding all frequent itemsets becomes equivalent to the problem of securely computing a dot product of two vectors. To solve this problem, the authors propose a secure scalar product protocol with linear communication cost.

The secure multi-party computation perspective addresses the problem of mining in a distributed environment. However, the methods discussed above reveal a common trend: they deal with either horizontal or vertical distribution of the data. While these two scenarios are useful theoretically, they are often not representative of real-world datasets. When data comes from heterogeneous sources, it is very likely that there will be no global schema available. Integration of data in this case is an open research problem [27]. Related to this, sites will neither have records containing the same set of items nor will they have values for the same set of attributes.

We term distribution of data which is neither horizontal nor vertical *hybrid distribution*. More precisely, in a hybrid distribution each site has information about a subset of the rows and a subset of the columns of the full dataset. Current secure multi-party computation methods do not address the problem of hybrid distribution of the data. Therefore, new methods for privacy-enhancing mining of hybrid data need to be developed.

2.2.3 Privacy with higher-order information. Within the context of statistical relational learning, another issue noted in the Introduction that needs to be addressed is the problem of enhancing privacy when dealing with higher-order associations between items in a dataset – associations that cross record boundaries [21]. (As noted previously the use of higher-order associations is based on the assumption that the data instances are not IID.) The higher-order association rule mining algorithm presented in [21] (DiHO ARM) has been developed quite recently and to the best of our knowledge, to-date no privacy-preserving or privacy-enhancing solutions for higher-order ARM have been proposed. DiHO ARM is based on building a graph model of the input data, and enhancing its privacy is related both to the more traditional privacy-preserving data mining methods as well as to the relatively new field of private graph data analysis discussed above. A privacy enhancing version of DiHO ARM necessarily must address the issues of securely detecting edges between records belonging to different sites as well as private enumeration of distributed paths with a limited maximal path length.

3 Approach

3.1 Homomorphic encryption schemes. One of the most important cryptographic primitives utilized in our approach to privacy-enhancing distributed higher-order itemset mining is homomorphic encryption. Homomorphic encryption schemes are public-key systems with the following properties:

1. There exists an addition operation $\oplus$, such that $Enc(A) \oplus Enc(B) = Enc(A+B)$;
2. There exists a multiplication-by-constant operation $\otimes$, such that $c \otimes Enc(A) = Enc(cA)$, where *c* is a constant.

Both the addition and the multiplication operations are efficient and work without knowledge of the private key. A number of homomorphic cryptosystems exist, including Pallier [29] and ElGamal [10]. ElGamal supports the additional property of rerandomization: given an encryption of a plaintext, a different encryption can be computed which decrypts to the same plaintext.

3.2 Latent higher-order itemset mining. Our approach to privacy-enhancing distributed higher-order itemset mining is based on the DiHO ARM LHOIM (Latent Higher Order Itemset Mining) algorithm proposed in [21]. This algorithm is based on the following definitions:

DEFINITION 1. *Two items a and b are related by co-occurrence in record r if both* $a \in r$ *and* $b \in r$. *The relation is denoted* $a \sim^{r} b$.

DEFINITION 2. *Two items are* n^{th}*-order associated if* $a \sim^{r_1} i_1 \sim^{r_2} i_2 \sim^{r_3} \ldots \sim^{r_{n-1}} i_{n-1} \sim^{r_n} b$, $i \neq j \Leftrightarrow r_i \neq r_j$.

DEFINITION 3. *A latent itemset is a set of items I such that for any two items a and b (* $a, b \in I$ *) a and b are* n^{th}*-order related for some* $n \geq 1$.

DEFINITION 4. *A link group is a sequence of records* $\{r_1, r_2, \ldots, r_n\}$, *s.t. each pair of consecutive records in the sequence share at least one item. It is denoted as* $r_1 \sim^{I_1} r_2 \sim^{I_2} \ldots \sim^{I_{n-2}} r_{n-1} \sim^{I_{n-1}} r_n$, *where* I_j *is the set of common items in* r_j *and* r_{j+1}. *A link group can also be interpreted as a set of latent itemsets generated by the same record sequence. The size of the link group is* $\prod_{i=1}^{n-1} |I_i|$. *The order of the link group is the number of records, n.*

The LHOIM algorithm creates a higher-order input space of link groups in two steps, which are also followed by our privacy-enhancing version of the algorithm:

1. A graph of records with common items is constructed.
2. All non-cyclical paths in the graph up to a certain length are enumerated. Each path defines a link group.

The link group space thus generated can be used as input to traditional ARM algorithms. We now proceed to describe how abstract graph generation and path enumeration can be performed in a privacy-enhancing manner.

Abstract Graph Generation. The key insight in this step of the algorithm is that the problem of constructing a graph of higher-order links between records in a distributed database is reducible to the problem of securely finding set intersections. If each record in the database is defined as a set of items $r = \{i_1, i_2, \ldots, i_n\}$, then, by definition, two records are connected by a second-order link iff their set intersection is non-empty: $r_j \sim r_k \Leftrightarrow r_j \cap r_k \neq \varnothing$. The algorithm which is presented in Figure 1 is based on an extension of the private set intersection protocol of Freedman et al. [9]. Unlike [9], our algorithm uses asynchronous communication to compute pairwise set intersections between an arbitrary number of sites.

```
Input: $DB_i$; Output: Abstract graph;
1.  Construct local graph;
2.  for each $r_j$ in $DB_i$
3.      $P_j$ := ComputePolynomial($r_j$);
4.      for dest := myrank..N
5.          ISend Enc($P_j$) to dest;
6.  done := 0;
7.  while done < $N$-1
8.  Recv msg from any src;
9.  Switch(msg.type)
10. case COEFFS:
11.     for each $r_k$ in $DB_{src}$
12.         for each $i_l$ in $r_k$
13.           $T_l$ = Evaluate polynomial;
14.           ISend $T$ to src;
15. case RESULTS:
16.     write to adjacency list;
17.     done++;
18. case EVALS:
19.     Decrypt and count 0’s;
20.     Write to adjacency list;
21.     ISend results;
22.     done++;
```

Figure 1. Abstract graph generation

An important precondition of this algorithm is that all items on all sites need to be uniquely and consistently encoded as integers. This makes the numerical manipulations as well as the encryptions possible. A cryptographic hash function, such as those discussed in [7], can be used to map strings to integers in a way that appears random. In the random oracle model, such hash functions do not compromise security [7] [36]. Also, for a large enough domain size (1024 bits is sufficient for practical purposes) the probability of collisions is exponentially small, which makes the hash function a good approximation of a perfect hash function [7].

The algorithm in Figure 1 runs on each participating site. We assume that the database is distributed among *N* parties. As noted in the Introduction, the distribution of the data can be hybrid. The share of the database owned by party *i* is denoted as DB_i. While communicating with site *j*, site *i* can assume one of two roles – it can be either Alice or Bob. We note that the same site *i* can act as Alice in relation to site *j* and as Bob in relation to some other site *k* . The output of the algorithm does not depend on the exact assignment of roles. For the sake of simplicity, in the pseudocode in Figure 1 the site with higher rank always acts as Bob. However, more sophisticated schemes can be used to better balance the roles a site takes.

In step 3, Alice computes a polynomial for each of its records via interpolation; the roots of the polynomial are equal to the integers to which items in the record are mapped. In the loop in step 4, Alice sends homomorphic encryptions of the coefficients of the polynomial to all sites which will act as Bob in relation to her.

Note that the degree of P_j is equal to the number of items in Alice’s record. In order not to reveal this information to Bob, Alice can expand all polynomials it sends to a uniform degree. In order to do this, Alice can multiply P_j with polynomials of the first degree which do not have integer roots, e.g., $3x+2$. Each multiplication by a linear component would increase the degree of P_j by one. This transformation does not introduce any new *integer* roots to P_j. Because Bob’s items are mapped to integer values, the outcome of the algorithm would not be affected by the expansion, because none of the newly introduced roots can coincide with values of items in Bob’s records.

The actions of Bob upon receiving the coefficients of the polynomials from Alice are in lines 10-14. Using the homomorphic encryption scheme, Bob can evaluate P_j with the integer values of the items in its own records. The resultant values are each multiplied with a random number, so that Alice cannot recover the original value and solve the equation to discover Bob’s items. To hide the size of his records, Bob can pad the vector of evaluations $\boldsymbol{T}$ with non-zero values encrypted with Alice’s public key.

Upon receiving the evaluations of P_j from Bob, Alice can decrypt them, and, for each of her records, discover if there is a higher order link from it to a record belonging to Bob. This part of the algorithm is found in lines 18-22. In lines 15-17, Bob receives the results from Alice and saves them.

During the execution of the protocol, each site learns the foreign records to which its own records are linked. Sites also learn the size of the intersection between any pair of records, which is equal to the number of 0’s in $\boldsymbol{T}$. In extreme cases this allows one site to learn the contents of some of the other site’s records. More precisely, suppose that record r_1 links to record r_2 and $i \in r_1$. Then,

$$\Pr(i \in r_2 \mid |r_1|, |r_1 \cap r_2|) = \frac{|r_1 \cap r_2|}{|r_1|}.$$

In one case, if record r_1 contains a single item *i*, it follows that r_2 also contains *i*. In another case, if the size of the intersection between the records is equal to the size of r_1, then the owner of r_1 will know that r_2 is a superset of r_1. In both cases the probability in the equation above becomes one (1). The rest of the information exchanged between Alice and Bob does not compromise privacy: by the semantic security of encryption, the coefficients Bob receives can be simulated as random numbers. Also, the non-zero values decrypted by Alice in ***T*** appear random because they are either multiplied by a random number or are random padding values.

After the completion of the algorithm, each site's graph is augmented with weighted (by intersection size) links to records belonging to foreign sites. Each link points to an ID number of a record on some other site. The algorithm essentially creates distributed adjacency lists on each site. For this reason, we term the graph created by the algorithm a *distributed graph* and the paths in this graph which contain vertices from more than one site *distributed paths*. All other paths are considered *local paths*. Let us also term the maximal local subpath of a distributed path a *local section* of the distributed path.

3.4 Security enhancement. Since, as noted, disclosing the size of the intersection between two records can lead to significant privacy breaches, we propose an enhancement to our algorithm which leaks only the existence of an edge. Let, as before, Bob be the site that receives an encryption of the polynomial constructed by Alice. In the original algorithm in Figure 1, Bob sends back a vector ***T*** of encrypted evaluations of the polynomial, padded with encrypted non-zero values. If, however, some of the padding values are zeroes, Alice would not know the exact size of the intersection. If Alice counts k zeroes upon decryption, and Bob has inserted l zero values, the two sites can use a Yao protocol to securely find if $k < l$. A positive answer implies the existence of an edge. Comparing the two numbers is simply an instance of the millionaire problem [35]. Efficient solutions exist for this problem with communication complexity logarithmic in the domain size of k and l [17]. Note that this method is more efficient than the more general solution in [9], which requires added communication linear in the size of the two records. These savings are helpful as the datasets involved in the computation can be very large in size.

Bob can choose the number of zeros to be added from some probability distribution. Furthermore, Bob can estimate the expected proportion of items shared between any two records from its local graph; let this proportion be $\boldsymbol{\varepsilon}$. If the expected proportion of zeros inserted in ***T*** by Bob is $(1\text{-}2\boldsymbol{\varepsilon})/2$, the total expected number of zeros counted by Alice would be half the number of elements in ***T***, assuming a certain regularity of the data. Under these assumptions, the number of zeroes can be simulated only using the length of the vector ***T***, which can be treated as a global parameter. Even if the regularity assumption is not entirely met, the amount of private information revealed is still significantly reduced, while adding communication cost only logarithmic in the size of the records[1].

3.5 Path Enumeration. Our approach is based on a distributed depth-first enumeration of paths [26]. Each site performs a depth-first enumeration in the same way it would with a non-distributed graph. Each local path is assigned a numerical ID. When the enumeration reaches a foreign node, the enumerating site takes the ID of the path that was discovered just before reaching the foreign node, encrypts it, and sends it, along with the foreign node information, to the owner of the foreign node. Effectively, a site transfers responsibility for the enumeration of the rest of the path to another site.

When a site receives information for one of its nodes, it starts a depth-first enumeration from this node. If a foreign node is reached again, the site follows the same procedure as before, adding the encrypted ID of its own local subpath to the list it received. The site assigns a numerical ID to each path it discovers and broadcasts the ID along with the list of encrypted sub-path ID's it received. The site announcing the numerical ID should also make sure each ID is unique[2]. Each party in the computation can decrypt the ID's which the party itself has encrypted, and only them. As a result, each site will know which of its own vertices are part of a distributed path, recognized by a unique ID number.

```
Input: G; MaxL, MaxSP
Output: Paths in G

forall vertex i and foreign calls to i
  Insert i in Path;
  Output();
  while 0 < Path.length
    if ∃ adjacent vertices not in Path
        AND Path.length <= MaxL
      v := adjacent vertex not in Path;
      if v is not local AND
          SubPaths.length < MaxSP then
        Add Enc(Path.ID) to SubPaths;
        Send {SubPaths,v} to v.owner;
      else
        Insert v in Path;
        Output();
    else
      Set Path.last to the next vertex
adjacent to its parent or remove
Path.last;

function Output()
  if SubPaths ≠∅ then
    Add Path to SubPaths;
    Broadcast {SubPaths, GlobalID};
  else
    Save Path with local ID;
```

Figure 2. Path Enumeration

[1] As of the time of writing this security enhancement was still being implemented in the PE-DiHO ARM framework.

[2] One way to handle this is to query a central site which returns a global numerical ID.

The algorithm in Figure 2 takes three inputs: the distributed graph *G* produced by the algorithm in Figure 1 and the two parameters: *MaxL*, which is the maximum number of vertices for a local path or a local subpath of a distributed path, and *MaxSP*, which is the maximum number of local sections of a distributed path. These two parameters are necessary to limit the time complexity of the algorithm.

The algorithm will enumerate all local paths of length less than or equal to *MaxL* and distributed paths not longer than $MaxL \times MaxSP$. However, not all distributed paths up to that length will be enumerated. For example, if *MaxL* = *MaxSP* = 2, distributed paths can reach length four, but some paths of length four can reside on three or four different sites and will not be enumerated. The problem can be easily solved by using one global maximum path length that is decreased each time a vertex is added and is sent to the foreign party together with the list of subpaths. However, doing this would reveal additional information – for paths which consist of two sections, the two sites which share the path will each know how long the path is on the other site.

For shorter maximal path lengths, all paths can be discovered by running the algorithm multiple times with different values for *MaxL* and *MaxSP*. For example, two runs can generate all paths of length two. One run with *MaxL*=2 and *MaxSP*=1 will generate all local paths, and a second run with *MaxL*=1 and *MaxSP*=2 will generate all distributed paths. We can do the same with paths of length three, but the situation is more complicated. The four runs needed in this case are:

Site Alice	Site Bob
(1) MaxL=3; MaxSP=1	MaxL=3; MaxSP=1
(2) MaxL=1; MaxSP=2	MaxL=2; MaxSP=2
(3) MaxL=2; MaxSP=2	MaxL=1; MaxSP=2
(4) MaxL=1; MaxSP=3	MaxL=1; MaxSP=3

Figure 2. Path Enumeration

In this scheme, run (1) generates all local paths; run (2) generates paths with one vertex on Alice and up to two on Bob; run (3) generates paths with one vertex on Bob and up to two on Alice; run (4) generates paths with three sections, each with one vertex. However it appears that there is no general way to do this for an arbitrary maximal path length. In particular, when the maximal path length is five and we have two sites, there can be paths which have three sections: a one-vertex section on site Alice, a two-vertex section on site Alice, and a two-vertex section on site Bob. In order to enumerate such paths, we need to set *MaxSP*=3 and *MaxL*=2 on both sites. With these parameters we will go beyond the maximal path length of five and will enumerate some paths of length six.

Therefore, for longer maximum path lengths and more nodes, the basic trade off we face is: a) enumerate all paths up to a certain length in one run but reveal additional private information; b) have stronger privacy guarantees, but enumerate a subset of all paths of length up to $MaxL \times MaxSP$.

Except for the local and distributed paths in *G*, each site participating in the computation also learns how many distributed paths include each of the party's local vertices. Also, each site knows how many local sections a distributed path has and to which sites they belong. If the encryptions of the subpath ID's are rerandomized, sites cannot tell if two distributed paths share a local section on a foreign site. These disclosures are unavoidable when the output is presented as a set of distributed paths.

3.6 Example Run. We will now illustrate our approach with a simple example. Let's assume the following distributed database: site Alice has two records – record 1 with items ACD and record 2 with items ACDF; site Bob has one record – record 1 with items AE. Alice and Bob map item A to 10, C to 20, D to 30, E to 40, and F to 50.

Alice and Bob construct the local parts of their graphs shown in Figure 3. Nodes in the figure represent records, and edges represent second-order links. Each edge is labeled with its weight, which is equal to the size of the intersection between the records it joins.

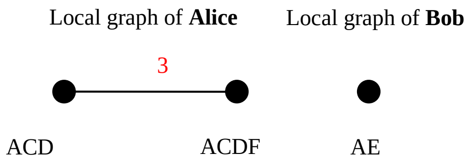

Figure 3. Local graphs

For record 1, Alice computes the polynomial:

$$P(i) = (i-10)(i-20)(i-30) = i^3 - 60i^2 + 1100i - 6000$$

Alice sends the encryptions of the coefficients 1, -60, 1100, and -6000 to B. Bob in turn computes:

$$\bar{T} = < Enc_A(P(10)) \bullet r_1, Enc_A(P(40)) \bullet r_2 > .$$

Here r1 and r2 are random numbers. Alice decrypts each of the components of vector *T* and gets one zero and other, random numbers. Thus, Alice concludes that record 1 is linked to a record in Bob with 1 common item. After repeating the same steps for record 2 in Alice, we get the final distributed graph shown in Figure 4.

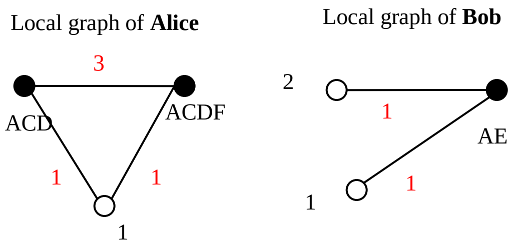

Figure 4. Distributed graph

Site Alice starts the depth-first search at ACD, and lists path ACD with ID 101 and path ACD-ACDF with ID 102. Independently, Bob also lists its only single-node local path with ID 201.

In the next step, Alice reaches a node in Bob and sends the message {(Enc$_A$(102)), 3} to Bob. Bob starts a depth-first search from AE. It first finds the single-node path AE, generates the number 301 as an ID for the distributed path, and broadcasts {(Enc$_A$(102)), 301}. Now Alice knows that its part in the distributed path with ID 301 (ACD-ACDF-AE) is its own path with ID 102 (ACD-ACDF). Then Bob adds Enc$_B$(201) to the subpath ID list. In the next step Bob finds the foreign node with record ID 1, and sends {(Enc$_A$(102), Enc$_B$(201)), 1} to the owner of the node, site Alice. There are no more unprocessed adjacent vertices and Alice does not enumerate a global path. The rest of the enumeration proceeds analogously.

3.7 Empirical Evaluation Methodology. In order to empirically demonstrate the validity of the privacy-enhancing approach, we compared PE-DiHO ARM to the non-privacy-enhancing DiHO ARM LHOIM algorithm [21]. DiHO ARM by default models second-order links. Textual data from four subsets of the 18828 version of the 20 Newsgroups dataset were used for the evaluation. The 20 Newsgroups dataset is a popular benchmark for text classification algorithms and contains approximately 20,000 newsgroup documents. The four subsets of the dataset which we used in the evaluation are COMP (containing five classes: comp.graphics, comp.window.xm comp.sys.mac.harware, comp.os.ms-windows.misc, comp.sys.ibm.pc.harware), SCIENCE (four classes: sci.crypt, sci.electronics, sci.med, sci.space), POLITICS (three classes: talk.politics.mideast, talk.politics.guns, talk.politics.misc), and RELIGION (three classes: alt.atheism, talk.religion.misc, soc.religion.christian). Each subset comprises 2000-3000 documents. Eight random samples were taken from each of the four subsets; each sample comprised 30% of the subset, except for the samples from SCIENCE which were taken at 25% due to its larger size. A total of 32 samples were processed using the non-privacy enhancing DiHO ARM algorithm compared to PE-DiHO ARM on 3, 8, and 32 nodes.

The raw textual data was preprocessed and formatted into XML documents. Linked records were merged using the two algorithms. Both sets of linked records were used as input to the CBA algorithm [23]. CBA is a supervised learning algorithm based on association rule mining, which considers the class of a data instance as a special record item. Using an adaptation of the classical Apriori algorithm, CBA generates all association rules which contain only the class item in the consequent. A heuristic approach is used to build a classifier from the association rules in CBA. We used this supervised learning approach with labeled data in order to enable us to compare PE-DiHO ARM with non-privacy-enhanced DiHO ARM based on error rate. This allows us to explore privacy-enhancing approaches that do not necessarily produce the same set of itemsets that our baseline algorithm discovers.

Error measurements for the results obtained from CBA were used for a statistical comparison. The standard t-test was used to evaluate whether the mean error rate for each of the four newsgroup subsets was the same for both algorithms, with at least 95% confidence.

4 Experimental Results

The experiments described in section 3.7 were executed with the following setup. Each dataset was randomly and equally partitioned among the participating sites (3, 8, and 32 sites were used). For the path enumeration step, two runs were made for each dataset: one with parameters *MaxL*=2, *MaxSP*=1; another with parameters *MaxL*=1, *MaxSP*=2. The generated paths were then used as input to the CBA supervised learning algorithm, with support set to 0.01%, confidence set to 100%, and without rule pruning. The average error rates and the standard deviation for each of the four subsets of the 20 Newsgroups dataset for the non-privacy-enhancing (Non-PE) algorithm and for each configuration of the privacy-enhancing (PE) algorithm are summarized in Table 1. The p-value computed using a Student's t-distribution with eight degrees of freedom is also shown.

The t-test p-values in Table 1 clearly show that with high confidence the distributions of error rates for PE-DiHO ARM and non-privacy-enhancing DiHO ARM are not statistically significantly different. Thus we conclude that the models built by PE-DiHO ARM are virtually identical to the models built by non-privacy-enhanced DiHO ARM. This provides evidence that the underlying input is also identical, and in fact in these experiments this is indeed the case. A comparison of the actual higher-order itemsets discovered by PE-DiHO ARM with those discovered by non-privacy-enhancing DiHO ARM reveals that the two sets contain precisely the same itemsets. Further experiments are needed for links of order three and greater, as well as for additional datasets such as WebKB and Citeseer. In fact these experiments are underway.

Table 1. Experimental Results (32 runs total)

	Non-PE	**PE 3 Nodes**	**PE 8 Nodes**	**PE 32 Nodes**	**p-value (PE vs. non-PE)**
COMP	34.6% ±2.6	34.6% ±2.6	34.6% ±2.6	34.6% ±2.6	1.00
POL	18.9% ±3.2	19.2% ±3.6	18.9% ±3.2	18.9% ±3.2	0.95
REL	23.17% ±1.3	23.17% ±1.3	23.14% ±1.3	23.14% ±1.3	0.98
SCI	15.9% ±0.8	15.9% ±0.8	15.9% ±0.8	15.9% ±0.8	1.00

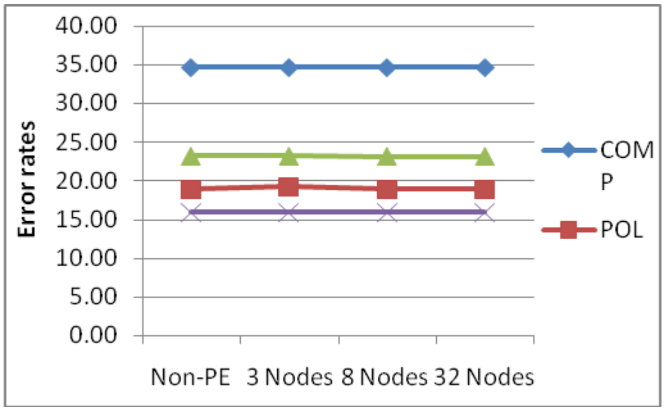

Figure 5. Non-PE vs. PE on 3, 8 and 32 Nodes

5 Conclusion and Future Work

This article has provided an in-depth analysis of several outstanding issues in privacy-enhanced distributed association rule mining based on higher-order itemsets. Typically both supervised and unsupervised data mining algorithms assume that instances are independent and identically distributed (IID). In the field of statistical relational learning (SRL), however, the IID assumption is not made because valuable correlations between instances can be leveraged to improve model performance [2]. However, the use of SRL requires deeper knowledge of datasets, making privacy issues more complex.

In this article we have presented a privacy-enhancing higher-order association rule mining algorithm that operates on hybrid fragmented data which is neither vertically nor horizontally distributed. We have also addressed the problem of privately constructing a graph from a distributed set of data instances and we have presented an algorithm which can privately enumerate paths in the graph. We demonstrate that although our algorithm significantly enhances privacy, it maintains the same performance as its non-privacy-enhancing counterpart. In so doing we address several outstanding issues in distributed association rule mining, especially when the non-IID assumption is made in SRL.

In future work we plan to evaluate our approach on higher-order links of length three and greater, as well as to conduct experiments on additional data from our law enforcement partners including sets that have hybrid, horizontal and/or vertical fragmentation. We also plan to investigate how our algorithms can be made secure against malicious adversaries.

As mentioned above, with our privacy-enhancing approach it is not possible in the general case to enumerate all paths up to a maximal length. For this reason, we will investigate how leveraging a subset of paths affects the performance of PE-DiHO ARM. Another problem, which is also an open question for the non-privacy-enhancing DiHO ARM algorithms, is formulating an optimal support metric for higher-order itemsets. Counting frequent higher-order itemsets in a privacy-enhancing way is tied to the definition of such a metric.

6 Acknowledgements

The authors wish to thank Rutgers University and the National Science Foundation. This material is based upon work partially supported by the National Science Foundation under Grant Numbers 0703698 and 0712139. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or Rutgers University.

We are also grateful for the help of co-workers, family members and friends. Co-author W. M. Pottenger also gratefully acknowledges the continuing help of his Lord and Savior, Yeshua the Messiah (Jesus the Christ) in his life and work.

7 References

1. R. Agrawal, and R. Srikant, Privacy preserving data mining, Proceedings of the ACM SIGMOD Conference on Management of Data, Dallas, USA, 2006, pp. 439 – 450.
2. R. Angelova, and G. Weikum, Graph-based Text Classification: Learn From Your Neighbors, Proceedings of SIGIR, Seattle, USA, 2006.
3. J. Brickell, V. Shmatikov, Privacy-Preserving Graph Algorithms in the Semi-Honest Model, ASIACRYPT, 2005.
4. D. W. Cheung, J. Han, VAT. Ng, AWE. Fu, and Y. Fu, A Fast Distributed Algorithm for Mining Association Rules, Proceedings Parallel and Distributed Information Systems, IEEE CS Press, 1996, pp. 31 – 42.
5. Du, Wenliang, Zhan, and Zhijun, Building decision tree classifier on private data, Proceedings of the IEEE ICDM Workshop on Privacy, Security and Data Mining, 2002
6. Y. Duan, J. Wang, M. Kam, and J. Canny, A secure online algorithm for link analysis on weighted graph. In Proceedings of the Workshop on Link Analysis, Counterterrorism and Security, SIAM Data Mining Conference, 2005, pages 71 – 81.
7. A. Evfimievski, Privacy Preserving Information Sharing, doctoral dissertation, Cornell University, August 2004.
8. A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, Privacy preserving mining of association rules, Proceedings of the 8th

ACM SIGKDD International Conference on Knowledge Discovery in Databases and Data Mining, Edmonton, Canada, 2002, pp. 217 – 228.

9. M. Freedman, K. Nissim, and B. Pinkas, Efficient private matching and set intersection, Advances in Cryptology (EUROCRYPT'04), Springer-Verlag, 2004, pp 1 – 19.
10. T.E. Gamal, A public key cryptosystem and a signature scheme based on discrete logarithms, CRYPTO, 1984, pp. 10 – 18.
11. M. Ganiz and W. M. Pottenger, Higher Order Naïve Bayes: A Novel Bayesian Classifier for Textual Data, SIAM International Conference on Data Mining, 2008 (Submitted for review)
12. L. Getoor, and C.P. Diehl, Link Mining: A Survey, SIGKDD Explorations 72, 2005, pp. 3 – 12.
13. O. Goldreich, Secure multi-party computation, working draft, 1998.
14. O. Goldreich, S. Micali, and A. Wigderson, How to play any mental game - a completeness theorem for protocols with honest majority, 19th ACM Symposium on the Theory of Computing, 1987, pp 218 – 229.
15. M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis, Resisting structural re-identification in anonymized social networks, In VLDB, 2008.
16. X. He, B. Shafiq, J. Vaidya, and N. Adam, Privacy-preserving link discovery, In Proceedings of the 2008 ACM Symposium on Applied Computing, 2008, 909-915.
17. I. Ioannidis and A. Grama, An efficient protocol for Yao's millionaires'problem, in Hawaii International Conference on System Sciences (HICSS-36), Jan. 6-9 2003.
18. M.Kantarcioglou, and C. Clifton, Privacy-preserving distributed mining of association rules on horizontally partitioned data, Proceedings of the ACM SIGMOD Workshop on Research Isuues in Data Mining and Knowledge Discovery, 2002, pp. 24 – 31.
19. M. Kantarcoglu, and J. Vaidya, Privacy preserving naïve Bayes classifier for horizontally partitioned data, IEEE ICDM Workshop on Privacy Preserving Data Mining, 2003.
20. H. Kargupta, S. Datta, Q. Wang, Krishnamoorthy Sivakumar, On the privacy preserving properties of random data perturbation techniques, ICDM, pp 99 – 106, 2003.
21. S. Li, C.D. Janneck, A.P. Belapurkar, M. Ganiz, X. Yang, M. Dilsizian, T. Wu, J.M Bright, and W.M Pottenger, Mining Higher-Order Association Rules from Distributed Named Entity Databases, Intelligence and Security Informatics 23 – 24, IEEE, 2007, pp. 236 – 243.
22. Y. Lindell, and B. Pinkas, Privacy-preserving data mining, Advanced in Cryptology – CRYPTO 2000, Springer-Verlag, 2000, pp 36 – 54. Springer-Verlag.
23. B. Liu, Wynne Hsu, and Yiming Ma, Integrating Classification and Association Rule Mining, Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98, Plenary Presentation), New York, USA, 1998.
24. K. Liu, K. Das, T. Grandison, and H. Kargupta, Privacy-Preserving Data Analysis on Graphs and Social Networks. In Next Generation Data Mining. Edited by Hillol Kargupta, Jiawei Han, Philip Yu, Rajeev Motwani, and Vipin Kumar, CRC Press, 2008.
25. K. Liu, H. Kargupta, J. Ryan, Random Projection-Based Multiplicative Data Perturbation for Privacy Preserving Distributed Data Mining, IEEE Transactions on Knowledge and Data Engineering, 18(1), 92–106, 2006.
26. V. Menon, W. M. Pottenger, and S. Li, BGP Event Classification and Anomaly Detection, Proceedings of the 2008 SIAM International Conference on Data Mining (Submitted).
27. V. Morocho, F. Saltor, and L. Perez-Vidal, Schema Integration on Federated Spatial DB across Ontologies, Proceedings of the 5th International Workshop on Engineering Federated Information Systems, EFIS, IOS Press, Coventry, UK, Jul 2003, pp. 63-72.
28. S. R. M. Oliveira and O. R. Zaiane, Privacy preserving clustering by data transformation, in Proceedings of the 18th Brazilian Symposium on Databases, Manaus, Amazonas, Brazil, October 2003, pp. 304–318.
29. P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, Advances in Cryptology EUROCRYPT '99, Springer-Verlag, 1999, pp. 223 – 238.
30. B. Taskar, P. Abbeel, and D. Koller, Discriminative Probabilistic Models for Relational Data, Proceedings of Uncertainty in Artificial Intelligence conference UAI02, Edmonton, Canada, 2002.
31. J. Vaidya, and C. Clifton, Privacy preserving naïve Bayes classifier on vertically partitioned data, 2004 SIAM International Conference on Data Mining, 2004.
32. J. Vaidya, and C. Clifton, Privacy preserving association rule mining in vertically partitioned data, The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 639 – 644.
33. R. Wright, and Z. Yang, Privacy-Preserving Bayesian Network Structure Computation on Distributed Heterogeneous Data, Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, USA, August 2004.
34. A.C. Yao, How to generate and exchange secrets, Proceedings of the 27th IEEE Symposium on Foundations of Computer Science, IEEE 1986, pp. 162 – 167.
35. A.C. Yao, Protocols for secure computation, Proc. 23rd IEEE Symposium on Foundations of Computer Science (FOCS), 1982, pp 160 - 164.
36. M. Zaki, Parallel and Distributed Association Mining: A Survey, IEEE Concurrency, Oct.~Dec. 1999, pp. 14 – 25.
37. E. Zheleva and L. Getoor, Preserving the privacy of sensitive relationships in graph data, In PinKDD Workshop, 2007.
38. S. Zhong, Z. Yang, R. N. Wright, Privacy-Enhancing k-Anonymization of Customer Data, Proceedings of the ACM SIGACT SIGMOD SIGART Symposyum on Principles of Database Systems, 2005, pp. 139-147.
39. Over 65 Arrested in International Methamphetamine Investigation, URL: http://www.usdoj.gov/dea/pubs/pressrel/pr041503.html, Accessed 12/28/2008.