

Distributed Higher Order Association Rule Mining Using Information Extracted from Textual Data

Shenzhi Li

Lehigh University
19 Memorial Dr W,
Bethlehem, PA 18015, USA
1-610-758-3737

shl3@lehigh.edu

Tianhao Wu

Lehigh University
19 Memorial Dr W,
Bethlehem, PA 18015, USA
1-610-758-3737

tiw2@lehigh.edu

William M. Pottenger

Lehigh University
19 Memorial Drive West,
Bethlehem, PA 18015, USA
1-610-758-3454

billp@lehigh.edu

ABSTRACT

The burgeoning amount of textual data in distributed sources combined with the obstacles involved in creating and maintaining central repositories motivates the need for effective distributed information extraction and mining techniques. Recently, as the need to mine patterns across distributed databases has grown, Distributed Association Rule Mining (D-ARM) algorithms have been developed. These algorithms, however, assume that the databases are either horizontally or vertically distributed. In the special case of databases populated from information extracted from textual data, existing D-ARM algorithms cannot discover rules based on higher-order associations between items in distributed textual documents that are neither vertically nor horizontally distributed, but rather a hybrid of the two. In this article we present D-HOTM, a framework for Distributed Higher Order Text Mining. D-HOTM is a hybrid approach that combines information extraction and distributed data mining. We employ a novel information extraction technique to extract meaningful entities from unstructured text in a distributed environment. The information extracted is stored in local databases and a mapping function is applied to identify globally unique keys. Based on the extracted information, a novel distributed association rule mining algorithm is applied to discover higher-order associations between items (i.e., entities) in records fragmented across the distributed databases using the keys. Unlike existing algorithms, D-HOTM requires neither knowledge of a global schema nor that the distribution of data be horizontal or vertical. Evaluation methods are proposed to incorporate the performance of the mapping function into the traditional support metric used in ARM evaluation. An example application of the algorithm on distributed law enforcement data demonstrates the relevance of D-HOTM in the fight against terrorism.

Keywords

Distributed data mining, distributed association rule mining, knowledge discovery, artificial intelligence, machine learning, data mining, association rule mining, text mining, terrorism

1. INTRODUCTION

The burgeoning amount of textual data in distributed sources combined with the obstacles involved in creating and maintaining central repositories motivates the need for effective distributed information extraction and mining techniques. One example of this is in the criminal justice domain. For instance, there are more than 1,260 police jurisdictions in the Commonwealth of Pennsylvania alone. As was made strikingly clear in the aftermath of the terrorist attack on September 11, different kinds of records on a given individual may exist in different databases – a type of data fragmentation. In fact, the United States Department of Homeland Security (DHS) recognizes that the proliferation of databases and

schemas involving fragmented data poses a challenge to information sharing. In response, the DHS is promulgating a “System of Systems” approach that acknowledges the infeasibility of creating a single massive centralized database [5]. Indeed, the picture that is emerging in the DHS is basically a three-tier structure with some overlap between tiers: local databases, state databases and federal databases. The state collects information from local jurisdictions to form a state-level centralized database, and likewise for the federal government. However, due to the sheer volume of data, constraints on system interoperability as well as legal restrictions on data sharing, not all information is passed from local jurisdictions to the state-level. Likewise, the federal level captures only a modicum of the data available in state-level databases. In essence, the resulting data sharing structure is pyramidal in nature. The more centralized the database, the less information that is shared. Given this reality, the DHS as noted has acknowledged that it is simply not feasible to keep an all-in-one federal database. As a result, the DHS is promoting a “System of Systems” approach that is based initially on the creation of standards for interoperability and communication in areas where standards are currently lacking. Indeed, efforts are underway to establish standards in database schema integration (e.g., OWL [7], GJXDM [11], etc.). Nonetheless, even with the widespread acceptance of such standards, the ability to integrate schemas automatically is still an open research issue [10, 8, 15, 16].

A related issue is the fact that current algorithms for mining distributed data are capable of mining data (whether vertically or horizontally fragmented) only when the global schema across all databases is known [4, 6, 18, 14, 9]. In the case of information extracted from distributed textual data, however, no preexisting global schema¹ is available. This is due to the fact that the entities extracted may differ between textual documents at the same or different locations. As a result, a fixed global schema cannot be assumed and current algorithms that rely on the existence of such a schema cannot be employed. For the same reason, this problem would not be solved even if fixed OWL or GJXDM-based schema mappings were available – in short, schemas of textual entities are highly fluid in nature.

As noted, distributed Association Rule Mining (ARM) algorithms mine association rules from data in homogeneous databases². It is our contention that the restriction to homogeneous database sources is unnecessary, and that useful rules can be mined from diverse databases with different local schemas as long as records can be linked via, for example, a unique key such as SSN. Many interesting applications emerge if one considers this approach, which we term *higher-order* distributed association rule mining. *Higher-order* implies that rules may be inferred between items that do not occur in the same local database schema. In other words, rules can be inferred based on items (entities) that may never occur

¹ I.e., global dictionary of extracted entities

² Homogeneous means either vertically or horizontally fragmented data.

together in any record in any of the distributed databases being mined independent of knowledge of a global schema.

In this article, we propose a distributed higher-order text mining framework that requires neither the knowledge of the global schema nor schema integration as a precursor to mining rules. The framework, termed D-HOTM, extracts entities and discovers rules based on higher-order associations between entities in records linked by a common key. D-HOTM has two components: entity extraction and distributed association rule mining. The entity extraction is based on information extraction rules learned using a semi-supervised active learning algorithm detailed in [22]. The rules learned are applied to automatically extract entities from textual data that describe, for example, criminal modus operandi. The entities extracted are stored in local relational databases, which are mined using the D-HOTM distributed association rule mining algorithm described in Section 3.

The article is organized as follows. Section 2 summarizes the related work in parallel and distributed ARM. Section 3 describes our D-HOTM framework, and is followed in Section 4 by a discussion of issues in calculating support raised by the fragmented nature of distributed textual entity data. We draw conclusions and discuss future work in section 5.

2. RELATED WORK

Association rule mining (ARM) discovers associations between items [1, 2]. Given two distinct sets of items, X and Y , we say Y is associated with X if the appearance of X implies the appearance of Y in the same context. ARM outputs a list of association rules of the format $X \Rightarrow Y$, where $X \Rightarrow Y$ has a predetermined support and confidence. Many ARM algorithms are based on the well-known Apriori [1] algorithm. In Apriori, rules are generated from itemsets, which in turn are formed by grouping items that co-occur in instances of data. The prototypical application of ARM is market-basket analysis in which items that are frequently purchased together are identified in order to aid in stores' layout of items.

Parallelism is an ideal way to scale ARM to large databases. There are two major approaches for using multiple processors: parallel ARM algorithms, in which all processors access shared memory, and distributed ARM algorithms, in which each processor accesses its own private memory and communication is accomplished via message passing. Most parallel and distributed ARM algorithms are based on a kernel that employs the Apriori algorithm [1].

Parallelism in both shared-memory and distributed memory ARM algorithms can be categorized as *data-parallelism* or *task-parallelism* [3, 23]. Data-parallelism logically partitions the database among processors. Each processor works on its local partition using the same computational model. Count distribution (CD) is a simple data-parallelism algorithm. Each processor generates the local candidate itemsets independently based on the local partition. Then the global counts are computed by sharing (or broadcasting) the local counts, and the global frequent candidate itemsets are generated. Data-parallelism exchanges only the counts among processors, which minimizes the communication cost. As a result, it seems ideal for use in a distributed environment.

In task-parallelism, each processor performs different computations independently, yet all have access to the entire dataset. For example, the computation of candidate itemsets of different sizes might be distributed among processors in a parallel loop across itemsets. In this case, each processor generates global counts independently. As noted, this requires that each processor have access to the entire dataset. In a distributed environment, this can be accomplished by performing an initial 'copy-in' operation of the dataset, albeit often at great cost.

Distributed ARM algorithms discover rules from distributed databases. Fast Distributed Mining (FDM) is based on count distribution [6]. The advantage of FDM over CD is that it reduces the communication cost by sending the local frequent candidate itemsets to a polling site instead of broadcasting. Also based on CD, Ashrafi, et al. [4] propose the Optimized Distributed Association Mining (ODAM) algorithm which both reduces the size of the average transaction and reduces the number of message exchanges in order to achieve better performance. The transactions are reduced by deleting the non-frequent items from the itemsets and merging several transactions with the same itemsets into one record. As for the message exchange, instead of using broadcast as with CD or polling sites like FDM, ODAM just sends all local information to one site, called the receiver. The receiver then calculates the global frequent itemsets and sends them back.

It is noteworthy that all of the parallel and distributed ARM algorithms discussed here assume that the databases are horizontally distributed. This limits the applicability of these algorithms. To address this issue, distributed mining of vertically fragmented data has received a growing amount of attention, especially in the context of privacy preserving data mining. For example, Vaidya and Clifton [18] propose a privacy preserving association rule mining algorithm for vertically distributed data. The authors use a vector to represent each (vertically fragmented) record in which the attributes in the schema are distributed amongst the different local sites. In essence, it is a method for mining association rules when the (global) schema is distributed amongst multiple sites. As noted in the introduction, however, this approach requires a priori knowledge of the complete (global) schema by all sites, and is thus unsuitable for mining rules from distributed data for which local schemas differ and the global schema is unknown. In fact, although these algorithms deal with fragmented data, they make the following three assumptions: (1) the databases are vertically fragmented; (2) there is a global key to unambiguously link subsets of items, and no additional techniques are needed to identify which subsets should be linked; (3) the global database schema is known. In the following section, we present the D-HOTM framework which performs entity extraction and distributed higher-order association rule mining independent not only of these three constraints, but also independent of the aforementioned constraint that data be horizontally distributed.

3. D-HOTM Framework

In this section, we present our Distributed Higher-Order Text Mining framework, which discovers rules based on higher-order associations between entities extracted from textual data.

3.1 Entity Extraction

The first step in D-HOTM is to extract linguistic features, or entities, from textual documents. For example, law enforcement agencies generate numerous reports, many of them in narrative (unstructured) textual form. Much valuable information is contained in these reports. Unfortunately many agencies do not utilize these descriptive reports – they are generally filed away either in hardcopy form (e.g., printed or typed), or in outdated electronic formats. Information extraction techniques can however be employed to automatically identify and extract data from such descriptions and store it in fielded, relational form in databases. Once stored in relational form, the extracted information is useful in a variety of everyday applications such as search, retrieval and data mining.

We have developed an algorithm that learns rules and extracts entities from unstructured textual data sources such as criminal modus operandi, physical descriptions of suspects, etc. Our algorithm discovers sequences of words and/or part-of-speech tags that, for a given entity, have high frequency in the labeled instances of the training data (true set) and low frequency in the unlabeled instances (false set). The formal definition of the class of rules discovered by our algorithm is given in [22], and each rule is termed a *reduced regular expression* (RRE). Our algorithm first discovers the most common element³ of an RRE, termed the *root* of the RRE. The algorithm then extends the RRE in “AND”, “GAP”, and “Start/End” learning phases (Figure 1 from [22]).

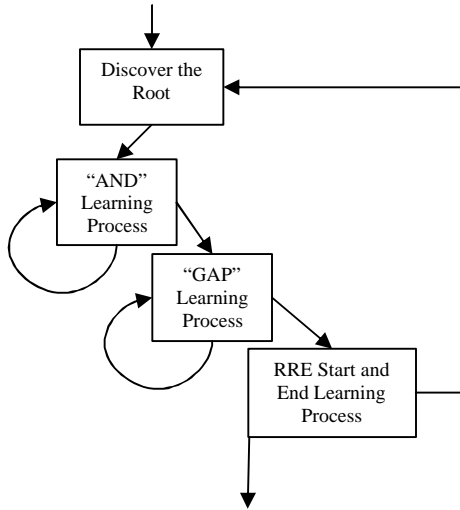


Figure 1: RRE Discovery Process (from [22])

Our approach employs a covering algorithm. After an RRE is generated for a subset of the true set for a given entity, the algorithm removes all segments covered by the RRE from the true set. The remaining segments become a new true set and the steps in Figure 1 repeat. The learning process ends when the number of segments left in the true set is less than or equal to a user-defined threshold [22]. In each iteration of Figure 1, a single RRE is generated. This RRE is considered a sub-rule of the current entity. After all RREs have been discovered for the current entity (i.e., all instances labeled with the entity are covered), the system uses the “OR” operator to combine the sub-rule RREs into a single rule that is also an RRE. Our results demonstrate that our algorithm achieves excellent performance on features important in law enforcement and the fight against terrorism [22].

3.2 Distributed Higher-Order ARM

After applying the entity extraction algorithm to unstructured textual data, a list of linguistic features is extracted that becomes input to our distributed higher-order (DiHO) ARM algorithm. We assume that each column in a given local database represents an object, which is for example an investigative report about a crime committed by a particular individual. In addition, each row in a given local database represents a single item known to exist in the object (document), which is an entity extracted using our semi-supervised active learning algorithm. The field value associated with each item is either zero or one depending on whether a given

entity is present or absent in the given object. It is clear that the distributed data cannot be horizontally fragmented because there is no guarantee that every site will include the same set of items (e.g., entities), and in the case where an object is not a document but a person, different distributed sites may also refer to the same object multiple times (e.g., investigative reports about different crimes committed by the same individual). On the other hand, the data is not vertically distributed either, because there is no one-to-one mapping connecting the distributed records in the distributed databases. In addition, the (local) ‘schema’ for each individual document varies, and no clean division of all objects’ items into identical sets can be made as required for vertically fragmented data. As a result, the data is neither vertically nor horizontally fragmented, but is present in a form we term a *hybrid fragmentation*.

The skeleton of our DiHO ARM algorithm is depicted in Figure 2.

- (1) Select linkage items
- (2) Assign a globally unique ID to each record
- (3) Identify linkable records using Apriori on global IDs
- (4) Exchange information about linkable records
- (5) For each site
- (6) Apply the Apriori algorithm locally

Figure 2: The DiHO ARM Algorithm

In step 1, the set of items used for linking records is selected. One requirement for this set is that the item (or combination of items) must uniquely identify objects. For example, given documents such as research papers, the combination of the two attributes *title* and *authors* might be selected as the set used for record linkage because in general these two attributes together form a unique identifier for each document. Given documents such as police investigative reports, SSN can be used for linkage because reports are written about individuals, and individuals are uniquely identified by SSN⁴.

In step 2, a globally unique ID is assigned to each record. This step is discussed in more detail in subsection 3.2.1, and an example is given in section 3.3.

Step 3 discovers linkable records using the item(s) selected in step 1. For example, if a given suspect appears in a burglary police report in Detroit, and the same suspect also appears in a mugging case in Philadelphia, and the SSN is chosen as the linking item, then those two distributed reports are considered linkable. In a practical sense, linking these two reports might reveal new information to the investigating police officer. This step is discussed in detail in subsections 3.2.2 and 3.2.3. A more extensive example is given in section 3.3.

After determining which distributed records are linkable, entities are exchanged and records merged. Continuing with the same example, entities extracted from the investigative report in Detroit are sent to Philadelphia, and vice versa. The two distributed reports about the same suspect are then treated as a new record which is stored in each local database. The final step is to apply a traditional association rule mining algorithm locally at each site to obtain the final association rules.

In the following subsections, we take a closer look at step 2, the resolution of object identifiers, and step 3, the detection of linkable records.

³ A word, part-of-speech tag or a punctuation mark.

⁴ In subsection 3.2.1 we address the issue of approximate record linkage.

3.2.1 Resolution of Object Identifiers

Our DiHO ARM algorithm requires that the items selected in step 1 of Figure 2 uniquely identify objects. In other words, the item (or combination of items) should be globally unique and consistent. Unique means different objects have different identifiers while consistent implies that the records about the same object have the same identifiers. To understand the need for uniqueness, consider for example the online paper databases Axiom (Compendex®, INSPEC®) and Citeseer. In many cases these two databases will have the same object (a reference to a scholarly research article) with different items. To illustrate, consider this simple example. Suppose the Citeseer database has a table: {ID, title, year, citenum} where ID is a unique ID assigned to the article (e.g., a DOI), title is the title of the publication, year is the year of publication, and citenum is the number of citations to the article. Likewise, suppose that the Axiom database has a table: {ID, title, year, code} where the first three fields are similar in meaning to those in Citeseer, and the fourth contains one of the classification codes assigned to the article by abstract and indexing personnel. Suppose an information scientist wishes to know the citation rate of articles containing certain Axiom classification codes. If we assume that the two databases use the same global ID for articles, then in DiHO ARM each site will generate the local frequent 1-itemset with the associated frequencies and exchange itemsets and frequencies. The 2-itemset {code citenum} can then be calculated using the global ID to match records in the two different databases – a higher-order association between code and citenum via the global ID.

A problem arises, however, when there is no guaranteed unique global ID for objects. In this particular example, in the absence of a DOI or URN, there may be no failsafe method to resolve object identity. It is reasonable to expect this kind of situation to arise fairly often, and steps must be taken to deal with it when it does. First, in the absence of a guaranteed globally unique ID, a decision must be made to approximate the ID. In this particular case, author names can be concatenated with the title to form a reasonable approximation of a globally unique ID. As a general solution we employ an edit-distance algorithm to match IDs formed in this way. Other special-purpose matching algorithms can be employed as well, such as that described in [19] for matching potentially falsified criminal suspect IDs. This does not guarantee, however, that for a given user-defined threshold, either the edit-distance or other special-purpose matching algorithms will predict matches correctly – both false positive and false negative mismatches are possible. Such mismatches will have an impact on both the support and confidence of the final rules. We deal with this issue further in section 4.

3.2.2 Detection of Linkable Objects

The key step in enabling the discovery of higher-order associations is the detection of linkable objects. There are two cases to consider: case one, in which the item(s) selected to link records in step 1 of Figure 2 are also used to form the globally unique ID in step 2. For example, in the aforementioned Axiom and Citeseer example, each local record corresponds to a single document object. The combination of the *title* and *author* attributes serve both to link records from different sites and to uniquely identify objects. Thus, by comparing the global identifier (e.g., using an edit-distance function), linkable objects can be discovered. It is worth noting that in this case, only 2nd-order associations between items (i.e., two items linked by a third) will be discovered during subsequent association rule mining.

The second case that needs to be considered in detecting linkable objects is more complex. In this case, object identifiers (such as DOIs for investigative reports) differ from the object identifiers used for linking. For example, suppose each document is an investigative report that contains criminal suspect information such as SSN and *modus operandi*. Higher-order associations can be discovered by linking two reports through the SSN item. In this case, the linking item, SSN, uniquely identifies individuals, not reports. This is different from the Citeseer/Axiom example in which the linking items were identical to the unique global identifier. The point in this example is that the item used to link reports (SSN) identifies an object (a criminal suspect) that is different than the investigative report object itself. In addition, in this case a single investigative report may contain multiple SSNs, and higher-order associations will not be limited only to 2nd-order. In what follows we lay the theoretical framework for the detection of linkable objects given these two scenarios.

3.2.3 Theoretical Framework for the Detection of Linkable Objects

In order to address the issues regarding the detection of linkable objects, it is first necessary to lay a theoretical foundation for reasoning about record linkage. As noted in section 2, the itemsets generated by Apriori are composed of items which co-occur in instances of data (i.e., in records). For example, for a given document record, *title* and *author* have particular values, and in that record these items are said to co-occur. Since the items co-occur in the same record, the co-occurrence is termed 1st-order, and is a direct link. Indirect links, on the other hand, involve more than one record and make use of a particular item to link records. For example, two different document records could be linked through a common author. Such links are termed *higher-order*. The DiHO ARM algorithm is designed to detect higher-order links between records. In the following, we first give a formal definition for such higher-order links. In the context of detecting linkable objects, we then prove that the maximum frequent itemsets generated using Apriori on the subset of items used to link records is sufficient to identify all linkable objects.

Definition 1: Given two records R_i and R_j , let $T = R_i \cap R_j$, and $T \neq \emptyset$. For any item $a \in T$, we say R_i and R_j are 2nd-order linkable through a , or a_2 -linkable, denoted as $R_i \sim^a R_j$.

Theorem 1: The relation between records that are a_2 -linkable is reflexive, symmetric and transitive.

Proof: (1) Assume item $a \in R_1$. Then $R_1 \sim^a R_1$ by Definition 1. Thus the relation is reflexive. (2) Assume R_1 and R_2 are 2nd-order linkable, then $R_1 \sim^a R_2$. Per Definition 1, $R_1 \sim^a R_2 \Leftrightarrow$ for some item a , $a \in R_1$ and $a \in R_2$. But this implies $R_2 \sim^a R_1$. Thus, the relation is symmetric. (3) Given that $R_1 \sim^a R_2$ and $R_2 \sim^a R_3$, then for some item a , $a \in R_1$, $a \in R_2$, and $a \in R_3$. Per Definition 1, R_1 and R_3 are a_2 -linkable, i.e., $R_1 \sim^a R_3$. Thus, the relation is transitive.

Without loss of generality, we simplify the discussion in what follows by dealing only with transitive links between records (i.e., we ignore reflexive and symmetric links). Furthermore, as will become evident, we allow each record to occur at most once in a given transitive path linking records.

Definition 2: Given n distinct records $R_1, R_2, \dots, R_i, \dots, R_n$ and $T_i = R_i \cap R_{i+1}$ with $T_i \neq \emptyset$, let L be a list of items $L = (a_1, a_2, \dots, a_i, \dots, a_{n-1})$ such that $a_i \in T_i$. Then we say records R_1 and R_n are $(k+1)$ th-order linkable through L , where k is the number of distinct

items in L . The $(k+1)$ th-order link is denoted $R_1 \sim^{a_1} R_2 \sim^{a_2} R_3 \sim^{a_3} R_4 \dots R_{i-1} \sim^{a(i-1)} R_i \dots R_{n-1} \sim^{a(n-1)} R_n$, or in short, $R_1 \sim^L R_n$. We term L a *viable path*.

For example: $R_1 \sim^a R_2 \sim^b R_3$ is a 3rd-order link because R_1 and R_3 are linked through two distinct items, a and b , and the viable path between R_1 and R_3 is (a, b) . On the other hand, $R_1 \sim^a R_2 \sim^a R_3$ is only a 2nd-order link because R_1 and R_3 are linked through only a single item a , and the viable path in this case is (a, a) .

Theorem 2: For any higher-order link between two records, there must exist at least one link which does not have repeated items in the viable path, or repeated records in the link.

Proof: (1) Suppose we have a higher-order link which has two occurrences of an item b in the viable path as follows:

$$R_1 \sim^{a_1} R_2 \sim^{a_2} \dots R_i \sim^b R_{i+1} \sim^{a(i+1)} \dots R_j \sim^b R_{j+1} \sim^{a(j+1)} \dots R_n$$

Per Definition 1, we have $b \in R_i$, $b \in R_{i+1}$, $b \in R_j$, $b \in R_{j+1}$, and clearly, $R_i \sim^b R_{j+1}$. Thus, the above higher-order link becomes a new link which has no repeated item in the viable path:

$$R_1 \sim^{a_1} R_2 \sim^{a_2} \dots R_i \sim^b R_{j+1} \sim^{a(j+1)} \dots R_n$$

(2) Suppose we have a higher-order link where some record R_i is the same as R_j . Then, the general form of the higher-order link is: $R_1 \sim^{a_1} \dots R_{i-1} \sim^{a(i-1)} R_i \sim^{a_i} R_{i+1} \sim^{a(i+1)} \dots R_{j-1} \sim^{a(j-1)} R_j \sim^{a_j} R_{j+1} \sim^{a(j+1)} \dots R_n$. Per Definition 1, $a_{i-1} \in R_{i-1}$, $a_i \in R_i$, $a_j \in R_j$, and $a_{j+1} \in R_{j+1}$; thus, the higher-order link can be rewritten as $R_1 \sim^{a_1} \dots R_{i-1} \sim^{a(i-1)} R_i \sim^{a_j} R_{j+1} \dots R_n$. In other words, no repeated records occur in the higher-order link.

When a given higher-order link has no repeated records or repeated items in the viable path, we term it a *minimal higher-order link*, and the corresponding viable path is termed a *minimal viable path*.

Let the records supporting item a be denoted as R_a , termed a *group on a* . From theorem 1, the records in R_a are a_2 -linkable to each other. For a given minimal viable path (a_1, a_2, \dots, a_n) , the corresponding minimal higher-order links can be written as $R_{a_1 a_2} \sim^{a_1} R_{a_1 a_2} \sim^{a_2} R_{a_2 a_3} \sim^{a_3} \dots R_{a(n-1) a_n} \sim^{a_n} R_{a_n a(n-1) a_n}$ for $R_{a_1} \neq R_{a_1 a_2} \neq \dots \neq R_{a_n}$. The notation R_{a-ab} , for example, means the group on item a minus the group on the pair of items ab . For instance, given a minimal viable path (a, b) , $R_{a-ab} \sim^a R_{ab} \sim^b R_{b-ab}$ represents all the higher-order links which are derivable from the viable path (a, b) . We term the set of such higher-order links a *higher-order link cluster*. To simplify, the higher order link cluster for a given minimal viable path (a_1, a_2, \dots, a_n) is denoted $R_{a_1} \sim^{a_1} R_{a_2} \sim^{a_2} \dots R_{a_n}$ where $\cap R_{a_i} = \emptyset$.

Definition 3: The length of a higher-order link cluster is defined as the number of groups in the cluster. The distance between two groups is defined as the shortest of all possible higher-order links.

For example, given the following two higher-order link clusters:

$$R_a \sim^a R_{ab} \sim^b R_{bc} \sim^c R_c$$

$$R_a \sim^a R_{ac} \sim^c R_c$$

The length of the first cluster is four while the length of the second is three. Suppose these two link clusters contain the only links between R_a and R_c , then from the link cluster $R_a \sim^a R_{ac} \sim^c R_c$ we see that the distance between R_a and R_c is three.

Theorem 4: Given a frequent k -itemset X generated by Apriori for $k \geq 2$, for any pair of items which are members of X , the distance between the groups on those items is at most three.

Proof: Suppose we have two groups R_b and R_c , where items $b, c \in X$. Per the Apriori algorithm, bc is a frequent itemset also.

We will thus have $R_{b-bc} \sim^b R_{bc} \sim^c R_{c-bc}$ if $R_b \neq R_{bc} \neq R_c$, and the distance between the group on b and the group on c is three. If $R_b \neq R_{bc} = R_c$, then we will have $R_{b-bc} \sim^b R_{bc} \Rightarrow R_{b-bc} \sim^b R_c$ with a distance of two between groups. The same case applies when $R_b = R_{bc} \neq R_c$. If $R_b = R_{bc} = R_c$, the distance between R_b and R_c is only one. Thus, we conclude that the distance between the group on b and the group on c is at most three.

Theorem 5: Given a frequent k -itemset A and j -itemset B ($j, k \geq 2$) for which A is not a subset of B and B is not a subset of A , suppose there exists at least one item which is a member of A and B . Furthermore, suppose there are items $a \in A$ and $b \in B$, then the distance between groups R_a and R_b is at most four.

Proof: (1) If item $a \in B$, then we have $a \in B$ and $b \in B$, and by theorem 4, the distance between groups R_a and R_b is at most three; (2) Let T be a set of items $T = \{t \mid t \in A \text{ and } t \in B\}$. Suppose $a \in A - T$, $b \in B - T$ and $c \in T$, then by theorem 4, the distance between groups R_a and R_c is at most three, i.e., $R_{a-ac} \sim^a R_{ac} \sim^c R_{c-ac}$. The distance between groups R_b and R_c is at most three also, i.e., $R_{b-bc} \sim^b R_{bc} \sim^c R_{c-bc}$. Thus, groups R_a and R_b can be linked through the viable path (a, c, b) , yielding the higher-order link cluster $R_{a-ac} \sim^a R_{ac} \sim^c R_{c-b} \sim^b R_{b-cb}$ with distance at most four.

As noted in the introduction to this subsection, our goal is to prove that the maximum frequent itemsets generated using Apriori on the subset of items used to link records is sufficient to identify all linkable objects. Clearly, given any itemset A that is a subset of some itemset B , any pair of items in A will appear in B , thus any higher-order link cluster generated for a given pair of items in A will be the same as the cluster generated for the same items in B . As a result, we may now conclude based on theorem 4 that the 3rd-order linkable objects can be generated using only the maximum frequent itemsets discovered by Apriori. Thus Apriori is applied in step 3 of the DiHO ARM algorithm depicted in Figure 2 to identify linkable records using the subset of items selected to form the global IDs. (Apriori will be used a second time (in step 6 of Figure 2) to compute the final higher-order association rules at each distributed site as well.) In addition, links of even higher-order may be generated via connections between the maximum frequent itemsets discovered by Apriori in step 3. It seems intuitive to speculate that the higher the order of the link, the weaker the link. The question of where to stop higher-order link detection is open at this point. In step 3 of our DiHO ARM algorithm in Figure 2, we speculatively limit our higher-order links to 4th-order. The algorithm for completing step 3 to identify linkable records is shown below in Figure 3. An example application of this algorithm is given in the following section.

```

Discover_Linkable_Records(level)
  For items selected in step 1 of Figure 2
    Count locally, add global IDs (GIDs) into local GIDList
    Broadcast; receive the GIDs and merge into GIDList
    If the |GIDList| < min_sup, remove the item
  If level == 2, exit // Per Definition 1
  Generate frequent itemsets using Apriori
  For each maximum frequent itemset
    Generate the 3rd-order link clusters // Per Theorem 4
  If level = 3, exit
  For any maximum frequent k-itemset A and j-itemset B
    If A and B have one or more common items
      Generate the 4th-order link clusters // Per Theorem 5

```

Figure 3: Generate 4th-order links

3.3 An Example Application of DiHO ARM

To further illustrate our algorithm, in this section we give a simple example. Consider a situation in the law enforcement domain where multiple investigative reports from different jurisdictions detail different crimes committed by the same person. In this case, the criminal is the primary key (perhaps identified by name or SSN), and the various facts such as modus operandi that surround different crimes become the fragmented data items associated with the key. Let’s suppose that our goal is to learn association rules that link the type of crime committed by an individual with some aspect of the modus operandi used in committing the crime (e.g., the type of weapon used). This kind of association rule can be very useful in narrowing the list of possible suspects to question about new criminal incidents⁵. However, as noted earlier, we have no guarantee in this case that both the crime type and weapon used will be recorded in a given investigator’s record of an incident. This can result, for example, from incomplete (or inaccurate) testimony from witnesses. Thus D-HOTM is applied to discover associations between crime type and weapon used in multiple jurisdictions’ distributed databases.

In Tables 1 and 2 below, we have depicted databases containing entities (i.e., items) extracted from 11 investigative police reports. Entities extracted from the reports are shown in the rows of the tables. For example, the entities “Allen” and “Gun” were extracted from the first report, D1, on Site 1. Tables 1 and 2 represent two databases at different (i.e., distributed) sites.

Table 1. Relational Database on Site 1

	D1	D2	D3	D4	D5	D6
Allen	1	0	0	0	0	0
Jack	0	1	0	0	0	1
Carol	0	0	1	0	0	0
Diana	0	0	0	1	0	1
John	0	0	0	0	1	0
Gun	1	0	0	0	1	0
Knife	0	1	1	0	0	1
Hands	0	0	0	1	0	0

Table 2. Relational Database on Site 2

	D7	D8	D9	D10	D11
Allen	1	0	0	0	0
Jack	0	1	0	0	0
Carol	0	0	1	0	0
Bill	0	0	0	1	0
John	0	0	0	0	1
Robbery	1	1	0	0	0
Mugging	0	0	1	0	0
Burglary	0	0	0	1	0
Kidnapping	0	0	0	0	1

In step 1 of the DiHO ARM algorithm in Figure 2, suppose that the suspect’s name is the item selected for linking records. Let us further suppose that each investigative report has been assigned a unique numerical ID as shown. In this case, however, the criminal suspect is the unique object, and the suspect’s name is used to link distributed records associated with each object. Hence, the items used to link records are {Allen, Jack, Carol, Diana, John, Bill}. Let the threshold for support be one. The next step is to discover

⁵ Because we are all creatures of habit, some good, some bad.

the frequency of the itemsets from the local records associated with these items that are being used for linking. The globally frequent itemsets are obtained by exchanging the local information. The result is depicted in Table 3.

Table 3. Local and Global ID Lists for Linkable Records

	Site 1	Site 2	GIDList (Doc IDs)
Allen	{1}	{7}	{1, 7}
Jack	{2, 6}	{8}	{2, 6, 8}
Carol	{3}	{9}	{3, 9}
Diana	{4, 6}	{}	{4, 6}
John	{5}	{11}	{5, 11}
Bill	{}	{10}	{10}
Jack, Diana	{6}	{}	{6}

Given an input level of three, step 3 in Figure 2 can be completed using the algorithm depicted in Figure 3. The first step is to generate 2nd-order links from 1-itemsets. Since only one record supports {Bill}, and we do not allow the same record to appear more than once in the minimal higher-order links, no 2nd-order links are generated for itemset {Bill}. However, the Apriori algorithm is applied using the remaining 1-itemsets to generate all the frequent k-itemsets. In this example, only one 2-itemset {Jack, Diana} is generated, which means that this is the only itemset capable of generating 3rd-order link clusters (by theorem 4). As there are only two items in the 2-itemset {Jack, Diana}, only a single higher-order link cluster exists; i.e., the link cluster between the group on *Jack* and the group on *Diana*. As the group on *Jack* is not the same as the group on *Diana*, by theorem 4 the distance between R_{Jack} and R_{Diana} is three, and the groups are higher-order linked as follows: $R_{Jack-Jack,Diana} \sim_{Jack} R_{Jack,Diana} \sim_{Diana} R_{Diana-Jack,Diana}$. Using the GIDLists to represent the groups on the items, we have $\{2,6,8\}-\{6\} \sim_{Jack} \{6\} \sim_{Diana} \{4,6\}-\{6\}$, or $\{2,8\} \sim_{Jack} \{6\} \sim_{Diana} \{4\}$. The resulting higher-order links and link clusters are portrayed in Table 4.

Table 4. Higher-Order Links

Itemset	Higher-order link clusters	Records involved
Allen	D1 \sim_{Allen} D7	{1, 7}
Jack	D2 \sim_{Jack} D6 D6 \sim_{Jack} D8 D2 \sim_{Jack} D8	{2, 6, 8}
Carol	D3 \sim_{Carol} D9	{3, 9}
Diana	D4 \sim_{Diana} D6	{4, 6}
John	D5 \sim_{John} D11	{5, 11}
Bill		{10}
Jack, Diana	{D2, D8} \sim_{Jack} D6 \sim_{Diana} D4	{2, 8, 6, 4}

This completes step 3 of the DiHO ARM algorithm in Figure 2. Next, step 4 of Figure 2 involves the exchange of the entities extracted from the linkable records, and the subsequent generation of new, merged records based on the higher-order links discovered. At this point in the computation, each site has the same global information, which is depicted in Table 5.

Table 5. Relational Database on All Sites

	D1,7	D2,4,6,8	D3,9	D5,11
Allen	1	0	0	0
Jack	0	1	0	0
Carol	0	0	1	0
Diana	0	1	0	0

John	0	0	0	1
Gun	1	0	0	1
Knife	0	1	1	0
Hands	0	1	0	0
Robbery	1	1	0	0
Mugging	0	0	1	0
Burglary	0	0	0	0
Kidnapping	0	0	0	1
Bill	0	0	0	0

In steps 5 and 6 in Figure 2, the Apriori algorithm is applied again, this time to each local database. Since higher-order associations have been implicitly included in the new, merged records, both higher-order and the usual first-order associations will be included in the resulting rules generated by Apriori. For example, “Gun \Rightarrow Robbery” and “Diana \Rightarrow Robbery” are rules generated based on higher-order associations discovered by the DiHO ARM algorithm. These rules cannot be discovered by existing distributed association rule mining algorithms.

Although this example is contrived, it is motivated by our work funded by the National Institute of Justice in modus operandi search of narrative textual police reports being conducted at Lehigh University [22]. We discuss this further in section 5.

4. METRICS FOR EVALUATION

The methods described in section 3.2.1 for resolving object identifiers lead to another challenge – evaluation. One of the most important metrics used in ARM is *support*, which is defined as the frequency of an itemset divided by the total number of objects (instances). In distributed databases, the total number of unique objects can be calculated by counting the number of unique global IDs. As noted in section 3.2.1, the function used to map local keys to a unique global identifier is not guaranteed to be 100% accurate. Different source records can be incorrectly mapped to the same global ID; likewise, records that should be mapped to a single global ID can be mapped to different IDs. The error rate of the mapping function will thus influence both the support and confidence metrics. It will not suffice to calculate support and confidence in the traditional way employed in existing ARM algorithms. The error rate of the mapping function must be considered in the calculation of these metrics. To our knowledge, no similar work has been conducted that addresses this issue. In what follows, we present a novel evaluation analysis that incorporates an error rate into support. To simplify the presentation we use upper case letters to represent sets and lower case letters to represent single elements or sizes.

In a realistic ‘real-world’ data mining scenario where databases reach terabytes in size, it is infeasible to obtain the true error rate of the mapping function. Hence, we must rely on an estimate of the error rate obtained from a sample of the data for which all errors have been manually identified. This sample data is termed a *ground truth* because, for the sample, the actual error rate is known. As a result, in the analysis that follows we estimate the true error rate as the upper bound of an assumed normally distributed observed error rate for a given confidence level⁶.

Given a sample data set $R=\{r_1, r_2, \dots, r_k\}$ where r_i is a local database record, the ground truth data set $G=\{g_1, g_2, \dots, g_n\}$ is a partition on R , where $g_i \subseteq R$, $\cap g_i = \mathcal{F}$ and each element g_i in G contains the

records which map to a single object. On the other hand, the ID mapping performed on R will result in a partition $P=\{p_1, p_2, \dots, p_n\}$, where $p_i \subseteq R$, $\cap p_i = \mathcal{F}$ and each element p_i in P contains the records which map to a single object. This second mapping might incorrectly map two different records which are in fact unique objects into a single group, or conversely fail to map the records representing the same object into a single group.

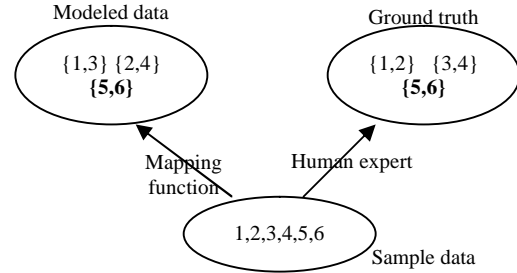


Figure 4: An Example of Mapping

Figure 4 depicts a simple example that reveals the problem context. The sample data set is $R=\{1,2,3,4,5,6\}$, where records 1 and 2 represent the same object (e.g., have the same primary key), records 3 and 4 represent another object, and records 5 and 6 represent a third object. Thus, the ground truth data set G has 3 elements which are {1,2}, {3,4} and {5,6}. Suppose the ID mapping function in this case also results in the three elements $P=\{\{1,3\},\{2,4\},\{5,6\}\}$. Clearly only one element in P is correct, {5,6}. The observed error rate is defined as the number of objects (i.e., objects in the ground truth data set) absent in the modeled data divided by the total number of objects (the size of the ground truth data set). In symbolic form the observed error rate f is:

$$f = 1 - \frac{|G \cap P|}{n}$$

We also define the observed difference degree t' as:

$$t' = \frac{n'}{n}, \quad t' > 0 \quad (1)$$

where n' is the size of the modeled data set and n is the size of the ground truth. Assuming that f is normally distributed, given a confidence level $1-c$, the probability that the normalized error rate is greater than z is:

$$\Pr \left[\frac{f - e}{\sqrt{e(1-e)/n}} > z \right] = c,$$

where e is the true population error rate. We use the upper bound of the confidence interval to estimate the true population error rate e of the data set as follows:

$$e = \frac{f + \frac{z^2}{2n} + z \sqrt{\frac{f - f^2}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}}$$

Similarly, the difference degree t is estimated in the same way. Having estimated both e and t based on sample data compared to the ground truth, the affect of the ID mapping function on support can be logically deduced as follows.

⁶ This technique is employed, for example, in the post-pruning stage of the popular C4.5 decision tree induction algorithm [20].

Given a set of real application data, the modeled data set D' can be obtained by applying the ID mapping function to the application data. The support rate for a given atom set on D' is denoted as s' .

Suppose D is the ground truth for the application data, which is unknown. To estimate the true support error rate s , we must estimate both the size of the ground truth data set as well as the number of objects supported (i.e., the support number).

Based on the definition of difference degree t , the size of the ground truth data set can be denoted as:

$$m = \frac{m'}{t}, \quad m = |D| \quad (2)$$

Consider the modeled application data D' as two parts: one part – denoted as T – containing all the correctly mapped objects; while the other part – denoted as E – is an incorrect partition over a set of records that represents the incorrectly mapped objects. Thus:

$$D' = T + E, \quad T \cap E = \mathbf{f}$$

The ground truth data can be represented in a similar way:

$$D = T + W, \quad T \cap W = \mathbf{f}$$

where W is the correct partition for the same set of records partitioned in E .

For the example shown in Figure 4, the modeled data set D' has three objects: $\{1,2\}$, $\{3,4\}$ and $\{5,6\}$, while the ground truth $D = \{\{1,3\}, \{2,4\}, \{5,6\}\}$. $T = \{\{5,6\}\}$ because the modeled data correctly group the two records 5 and 6 into one object, as in the ground truth. $E = \{\{1,3\}, \{2,4\}\}$ and $W = \{\{1,2\}, \{3,4\}\}$. Clearly E and W are two different partitions of the same four records, and $E \cap W = \mathbf{f}$.

Theorem 6: Given a set $X = \{x_1, x_2, \dots, x_u\}$ where $X \subseteq E$, $|x_i| = 1$; set $Y = \{y_1, y_2, \dots, y_v\}$ where $Y \subseteq W$ and $\cup x_i = \cup y_j$, for $\forall (y_j \in Y) |y_j| \geq 2$.

Proof. Suppose $\exists (y_j \in Y)$, $y_j = \{r : |y_j| = 1\}$. Since Y and X are two partitions of the same data set, $\exists (x_i \in X)$ where $r \in x_i$. Since $|x_i| = 1$, $x_i = \{r : |x_i| = 1\}$. We thus conclude that $x_i = y_j$, which is correctly mapped. This however violates the original assumption.

Let S' be the subset supporting a given itemset in D' (i.e., $s' = \frac{|S'|}{m'}$, $|S'| \geq 1$), and S be the supporting subset in D (i.e., $s = \frac{|S|}{m}$). Furthermore, let

$$S' = S_T + S_E, \quad S_T \subseteq T, S_E \subseteq E$$

$$S = S_T + S_W, \quad S_T \subseteq T, S_W \subseteq W$$

where S_W is the correct partition of a subset of the data while S_E is an incorrect partition. Let's consider the following cases.

Case 1. $S_E = \mathbf{f}$ and $S_T = S'$

$S_E = \mathbf{f}$ means that all records in the application data are correctly mapped by the mapping function. Since S_W partitions the same data set as S_E , $S_W = \mathbf{f}$. Thus, the true supporting data set can be derived as:

$$S = S_T + S_W = S' + \mathbf{f} = S'$$

As a result, the true support rate in this case is

$$s = \frac{|S_T|}{m} = \frac{|S'|}{m} = \frac{s' \cdot m'}{m} = t \cdot s'$$

Case 2. $S_T = \mathbf{f}$ and $S_E = S'$

$$S_E = S' \implies |S_E| = |S'| = s' \cdot m' \quad (3)$$

Assuming that the estimated error rate e is the true error rate for the ground truth data set D , we have:

$$|W| = e \times m \quad (4)$$

From equations (2) and (4), the size of E can be derived as:

$$\begin{aligned} |E| &= m' - |T| \\ &= m' - (m - |W|) \\ &= tm - (m - (1-e)m) \\ &= (t + e - 1)m, \quad t + e - 1 > 0. \end{aligned}$$

From equations (2) and (3), the size of $|S_E|$ can be derived as:

$$|S_E| = s' m' = s' t m$$

A special situation can occur in which each of the objects in S_E only contains one record. The number of records in S_E would thus equal the number of objects in S_E , which is $s' t m$. By Theorem 6, the size of each object in S_W must be at least two. The largest possible size of any object in S_W is the total number of records in S_E . This occurs when every record in S_E represents the same object. Thus, we conclude that:

$$1 \leq |S_W| \leq \lfloor s' t m / 2 \rfloor \quad (5)$$

In the general case when objects in S_E contain more than a single record, the bounds for S_W are:

$$1 \leq |S_W| \leq em \quad (6)$$

The lower bound is achieved when the records contained in all the elements in S_E represent the same object. Assuming that the remaining objects in E map to z objects in W , $em - z$ objects in W correspond to the objects in S_E . When S_E equals E , z is 0. Thus the upper bound em is achieved.

Combining equations (5) and (6), we conclude that:

$$\begin{aligned} 1 \leq |S_W| &\leq \max(em, \lfloor s' t m / 2 \rfloor) \\ \implies \frac{1}{m} \leq s &\leq \max\left(\frac{em}{m}, \frac{\lfloor s' t m / 2 \rfloor}{m}\right) \\ &\implies \begin{cases} \frac{1}{m} \leq s \leq \max\left(e, \frac{s' t}{2}\right), & \text{if } m \text{ is even} \\ \frac{1}{m} \leq s \leq \max\left(e, \frac{s' t(m-1)}{2m}\right), & \text{if } m \text{ is odd} \end{cases} \quad (7) \end{aligned}$$

Since m is large, we assume that $\frac{m-1}{m} \approx 1$, and thus equation (7) becomes:

$$s \in \left[\frac{1}{m}, \max\left(e, \frac{s' t}{2}\right) \right] \text{ for a given } |S_E| = s' t m \quad (8)$$

Case 3. Based on the previous two cases, we can now explore the true error rate in the general situation where

$$S' = S_T + S_E, \quad S_T \subseteq T, S_E \subseteq E.$$

Let $\alpha = |S_E| / |S'|$ and $1-\alpha = |S_T| / |S'|$ where $\alpha \in [0,1]$. This implies that $|S_T| = (1-\alpha) \cdot |S'| = (1-\alpha) \cdot s' \cdot m' = (1-\alpha) \cdot s' \cdot tm$ and that $|S_E| = \alpha \cdot |S'| = \alpha \cdot s' \cdot m' = \alpha \cdot s' \cdot tm = (\alpha \cdot s') \cdot tm$.

The true support rate s can thus be represented as:

$$\begin{aligned} s &= \frac{|S|}{m} = \frac{|S_T| + |S_W|}{m} \\ &= \frac{|S_T|}{m} + \frac{|S_W|}{m} \\ &= (1-\mathbf{a})s't + \frac{|S_W|}{m} \\ &= -s't \cdot \mathbf{a} + s't + \frac{|S_W|}{m}, \quad \mathbf{a} \in [0,1] \end{aligned}$$

Obviously, $s(\mathbf{a})$ is a linear function for which the upper bound is reached when $\mathbf{a}=0$ and the lower bound when $\mathbf{a}=1$. Thus:

$$s \in \left[\frac{|S_W|}{m}, s't + \frac{|S_W|}{m} \right] \quad (9)$$

From equation (8), given that $|S_E| = (\alpha \cdot s') \cdot tm$,

$$\frac{|S_W|}{m} \in \left[\frac{1}{m}, \max\left(e, \frac{\mathbf{a} \cdot s't}{2}\right) \right].$$

Equation (9) can thus be expressed as:

$$\begin{aligned} s &\in \left[\frac{1}{m}, s't + e \right] \\ \Rightarrow s &\in \left[\frac{t}{m'}, s't + e \right] \end{aligned}$$

The lower bound for the support s on the ground truth implies that there is only one object supporting the itemset. This situation occurs when the records contained in all the objects in S_E represent a single object, which results in only one object in S_W . This result serves to demonstrate the fact that the calculation of support and confidence in distributed association rule mining is non-trivial. Naturally, we speculate that the probability of this extreme lower bound occurring is not large. Nonetheless, it cannot be ruled out theoretically. One way to improve the utility of the lower bound somewhat is to substitute t/m' for $1/m$ because m' is a known value and t/m' may be larger than one. In fact, the utility of these bounds depends on the application. If the application needs demand a conservative estimate for support, the lower bound may be appropriate. For applications involving extremely large data sets, especially in distributed databases with relatively low bandwidth connections, choosing a higher support rate will be extremely helpful in reducing the size of the itemsets generated.

Although this analysis represents to the best of our knowledge the first attempt to incorporate global ID mapping errors into the evaluation of distributed ARM, clearly there is much work that remains. Specifically, we have yet to deal with the impact of errors in the ID mapping on the confidence metric, as well as the impact on the resulting rules. The approach we have taken, however, serves to blaze a trail for such future work.

5. CONCLUSIONS AND FUTURE WORK

We have presented D-HOTM, a novel distributed higher-order text mining algorithm that mines hybrid distributed data. Our D-HOTM algorithm is a first step towards tackling the difficult challenge posed by heterogeneous distributed databases that cannot be easily centralized. This is also the first work to address the complex issues surrounding the use of the traditional support metric in the context of distributed higher-order ARM. Even so, this is just the beginning of the research task at hand and much of real interest remains to be accomplished.

In no particular order, we plan to address both theoretical and practical issues in areas such as further exploration of the utility of higher-order associations as well as identifier linkage, evaluation metrics and workload balancing. Although D-HOTM applies a key resolution method to identify globally unique IDs, currently the algorithm relies on the user to identify the database fields to be used in generating IDs. This process could be partially automated given semantic mappings such as those supported by the recently released Web Ontology Language [7]. A related challenge is the need for further work in the adaptation of metrics for distributed mining of hybrid fragmented data. Our initial foray into this area serves primarily to highlight the need to address these issues in a rigorous manner within an overall framework, both theoretical and empirical, for evaluation. We have taken the first steps in creating the theoretical framework herein, and have also developed an empirical framework in the Text Mining Infrastructure (TMI), a software infrastructure for sequential, parallel and distributed textual data mining [13]. D-HOTM will be released initially in a Linux/MPI version packaged with the existing TMI at hddi.cse.lehigh.edu.

One of the more interesting open problems that has not been addressed in any previous work that we are aware of in distributed ARM deals with the efficiency of computation. D-HOTM too as presented is just a skeleton in this regard – a great deal of work is needed to deal with workload balancing as well as optimization of both computation and communication. Unbalanced workloads coupled with the requirements of synchronization will heavily affect the efficiency of the algorithm. This can be mitigated in part by the fact that the D-HOTM can be used to mine association rules from the results returned from a search (as opposed to mining rules from entire databases). For instance, in the Axiom/Citeseer example given earlier, a user may wish to find strong associations between Axiom's classification codes and the number of citations on Citeseer for a particular set of documents returned by queries. Furthermore, from a different perspective, multilevel parallelism can be introduced to improve runtime performance: e.g., at each distributed database, a parallel ARM algorithm can be executed. This also leads to the need for time and space complexity analyses based on metrics such as isoefficiency [12].

Foremost in our thoughts, however, is a plan to deploy D-HOTM in a law enforcement environment. We have done much work in preparation for such a deployment. In [22], we detail our work in information extraction from narrative textual data sources. In [21] we describe the design of a system based on the theory developed in [22] that has recently been deployed in the Bethlehem, Pennsylvania Police Department. This system uses advanced information extraction techniques to mine modus operandi data from the narrative text of police investigator's reports. The extracted data populates a relational database, thereby enabling straightforward modus operandi search.

The next phase of our research will build on this system. It involves the development and deployment of D-HOTM in the Northampton County, Pennsylvania region. Northampton County,

Pennsylvania has 32 independent police jurisdictions, none of which share modus operandi data on a systematic basis. The County has, however, recently implemented a high-bandwidth networking infrastructure that supports secure communication amongst all 32 jurisdictions. It is our plan to deploy D-HOTM in this environment, with distributed higher-order modus operandi association rule mining and search as our first application. In this way we will take a step towards realizing the “System of Systems” envisioned by the US Department of Justice.

There are many other practical examples of how distributed higher-order rules can be mined from multiple diverse databases. The common advantage of employing such technology is the promise of discovery of higher-order associations in data sources that cannot be easily centralized. Thus the area of distributed association rule mining in general, and higher-order distributed mining in particular, is an intriguing and promising area of research. Existing distributed ARM algorithms, however, assume that the distributed databases are either horizontally or vertically fragmented. In this article, we have presented a novel framework, D-HOTM, which supports mining of higher-order rules from distributed textual data in a hybrid fragmented form.

6. ACKNOWLEDGEMENTS

The authors wish to thank Lehigh University, the Pennsylvania State Police, the Lockheed-Martin Corporation, the City of Bethlehem Police Department and the National Institute of Justice, Office of Justice Programs, US Department of Justice. This work was supported in part by NIJ grant number 2003-IJ-CX-K003. Points of view in this document are those of the authors and do not necessarily represent the official position or policies of Lehigh University, the US Department of Justice, the Pennsylvania State Police or the Lockheed Martin Corporation.

We are also grateful for the help of Tim Cunningham, other co-workers, family members and friends. We also gratefully acknowledge the continuing help of our Lord and Savior, Yeshua the Messiah (Jesus the Christ) in our lives and work. Amen.

7. REFERENCES

- [1] Agrawal R., Imielinske T., and Swami A. N. Mining association rules between sets of items in large databases. In *Proc. of the 1993 ACM SIGMOD Int'l. Conference on Management of Data*, pages 207-216, Washington, D.C., June 1993.
- [2] Agrawal R., Mannila H., Srikant R., Toivonen H., and Inkeri Verkamo A. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307-328. AAAI/MIT Press, 1996.
- [3] Agrawal R. and Shafer J. C. Parallel Mining of Association Rules. *IEEE Trans. Knowledge and Data Eng.*, vol. 8, no. 6, 1996, pp. 962-969.
- [4] Ashrafi M. Z., Taniar D. and Smith K. ODAM: an Optimized Distributed Association Rule Mining Algorithm. *IEEE Distributed Systems*, vol. 5, No 3, March 2004.
- [5] Boyd D., Director of the Department of Homeland Security's new Office of Interoperability and Compatibility, in a presentation at the Technologies for Public Safety in Critical Incident Response Conference and Exposition 2004, New Orleans, LA, September.
- [6] Cheung D. W., Han J., Ng VAT., Fu AWE. and Fu Y. A Fast Distributed Algorithm for Mining Association Rules. *Proc. Parallel and Distributed Information Systems*, IEEE CS Press, 1996, pp. 31-42.
- [7] Dean M. and Schreiber G. OWL Web Ontology Language Reference. Editors, W3C Recommendation, 10 February 2004. [Online Article]. Retrieved Nov. 17, 2004 from the World Wide Web: <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>.
- [8] Draper D., Halevy A. Y., Weld D. S., The Nimble XML Data Integration System, Proceedings of the 17th International Conference on Data Engineering, p.155-160, April 02-06, 2001.
- [9] Evfimievski A., Srikant R., Agrawal R. and Gehrke J. Privacy preserving mining of association rules. SIGKDD'02, Edmonton, Alberta, Canada, 2002.
- [10] Genesereth M. R., Keller A. M., and Duschka O. M. Infomaster: An information integration system. In Proceedings of the ACM SIGMOD Conference, May 1997.
- [11] GJXDM. Global Justice XML Data Model. [Online Article]. Retrieved Nov. 17, 2004 from the World Wide Web: <http://www.it.ojp.gov/gjxdm>
- [12] Grama A., Gupta A. and Kumar V. Isoefficiency function: a scalability metric for parallel algorithms and architectures. *IEEE Parallel and Distributed Technology*, vol. 1, no. 3, pp. 12-21, 1993.
- [13] Holzman, L.E., Fisher, T.A., Galitsky, L. M., Kontostathis, A., and Pottenger, W. M. A Software Infrastructure for Research in Textual Data Mining. *The International Journal on Artificial Intelligence Tools*, volume 14, number 4, pages 829-849, 2004.
- [14] McConnell S. and Skillicorn D. B. Building predictors from vertically distributed data, Proceedings of the 2004 conference of the Centre for Advanced Studies conference on Collaborative research, p.150-162, October 04-07, 2004, Markham, Ontario, Canada
- [15] Papakonstantinou Y. and Vassalos V. Architecture and Implementation of an Xquery-based Information Integration Platform. *IEEE Data Engineering Bulletin*, vol 25, n. 1, pg 18-26, 2002.
- [16] Rahm E., Bernstein P.A. A survey of approaches to automatic schema matching. *VLDB J.* 10:4 (2001), pp. 334-350.
- [17] Schuster A. and Wolff R. Communication-Efficient Distributed Mining of Association Rules. *Proc. ACM SIGMOD Int'l conf. Management of Data*, ACM Press, 2001, pp. 473-484.
- [18] Vaidya J. and Clifton C. Privacy preserving association rule mining in vertically partitioned data, Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, July 23-26, 2002, Edmonton, Alberta, Canada.
- [19] Wang, G., Chen, H. and Atabakhsh H. Automatically Detecting Deceptive Criminal Identities. *Communications of the ACM*, 47 (3): 71-76, 2004.
- [20] Witten, I. and Frank, E. *Data Mining*. Morgan Kaufmann, New York, 2000.
- [21] Wu, T. and Pottenger, W. M. A Software System for Information Extraction in Criminal Justice Information Systems. Technical Report LU-CSE-04-009, Lehigh University, Bethlehem, PA, July 2004. [Online Article]. Retrieved from the World Wide Web: http://www3.lehigh.edu/images/userImages/jgs2/Page_3813/LU-CSE-04-009.pdf
- [22] Wu, T. and Pottenger, W. M. A Semi-Supervised Active Learning Algorithm for Information Extraction from Textual Data. *Journal of the American Society for Information Science and Technology (JASIST)*, volume 56, number 3, pages 258-271, 2004.
- [23] Zaki M. J. Parallel and Distributed Association Mining: A Survey. *IEEE Concurrency*, Oct.-Dec. 1999, pp. 14-25.

ABOUT THE AUTHORS:

Shenzhi Li is a Ph.D. student at Lehigh University conducting research in distributed association rule mining.

Tianhao Wu is a Ph.D. student at Lehigh University conducting research in reduction of knowledge engineering cost in information extraction.

William M. Pottenger, Ph.D., is an assistant professor at Lehigh University directing research in theories and algorithms for parallel and distributed text and data mining.