D-HOTM: Distributed Higher Order Text Mining

Shenzhi Li, Aditya P. Belapurkar, Christopher D. Janneck
Murat Can Ganiz, William M. Pottenger, Tianhao Wu

Lehigh University Department of Computer Science and Engineering
19 Memorial Drive West, Bethlehem, PA 18015, USA

{shl3, apb204, cdj2, mug3, billp and tiw2}@lehigh.edu

Abstract. We present D-HOTM, a framework for Distributed Higher Order Text Mining based on named entities extracted from textual data that are stored in distributed relational databases. Unlike existing algorithms, D-HOTM requires neither full knowledge of the global schema nor that the distribution of data be horizontal or vertical. D-HOTM discovers rules based on higher-order associations between distributed database records containing the extracted entities. A theoretical framework for reasoning about record linkage is provided to support the discovery of higher-order associations. In order to handle errors in record linkage, the traditional evaluation metrics employed in ARM are extended. The implementation of D-HOTM is based on the TMI [29] and tested on a cluster at the National Center for Supercomputing Applications (NCSA). Results on a dataset simulating an important DEA methamphetamine case demonstrate the relevance of D-HOTM in law enforcement and homeland defense.

Keywords Association Rule Mining (ARM), Data Mining, Text Mining, Distributed Association Rule Mining, Distributed Higher-Order Association Rule Mining (DiHO ARM), Distributed Higher-Order Text Mining (D-HOTM).

1 Introduction

With the spread of information technology and subsequent accumulation of data, data mining is becoming a necessary data analysis tool with a variety of applications. Among the different approaches to data mining, association rule mining (ARM), is one of the most popular. ARM generates rules based on item co-occurrence statistics. Co-occurrence, also called 1st-order association, captures the fact that two or more items appear in the same context. Orders of association higher than 1st-order are termed higher-order associations. Higher-order association refers to association among items that come from different contexts. The higher-order associations are formed by linking different contexts through common item(s). For example, if one customer buys {milk, eggs}, and another buys {bread, eggs}, then {milk, bread} is a higher-order association linked through "eggs".

Higher-order associations are employed in a number of real world applications including law enforcement and homeland defense. For example, methamphetamine use is the number one drug problem in 60% of US counties and children are often the victims due to the social nature of the use of this drug – parents often are both abusers, which endangers the health of the entire family [30]. The United States Drug Enforcement Administration (DEA) has conducted several operations to investigate the

entire methamphetamine trafficking process. In 2003, the DEA made arrests based on the use of manual higher-order association techniques that linked distributed documents through addresses, phone numbers, etc.

Currently, there are no ARM algorithms capable of mining distributed higher-order associations of this nature. Existing ARM algorithms for mining distributed data are capable of mining only data that is either horizontally or vertically fragmented [6] [27] [29]. In addition, they assume that data/schema integration problems have been solved [12]. Absent the ability to reason about record linkage, distributed ARM algorithms are incapable of identifying higher-order associations. Similarly, existing algorithms capable of mining higher-order associations are incapable of mining distributed data.

This paper proposes a novel distributed higher-order association rule mining (DiHO ARM) framework that (1) provides a theoretical basis for reasoning about higher-order record linkage as well as about metrics for evaluation in the presence of errors in record linkage; (2) is able to discover propositional rules based on higher-order associations between records linked by common items; (3) in the absence of knowledge of the complete global schema, enables mining of distributed data in a hybrid form that is neither horizontally nor vertically fragmented.

The paper is organized as follows: in section 2 we discuss background and related work. In section 3 we present the D-HOTM algorithm design and a theoretical framework for reasoning about record linkage. In Section 4 we discuss the implementation of the system. We present results in section 5, and close with conclusions and future work in section 6.

2 Related Work

As noted in the Introduction, traditional ARM algorithms only identify 1st-order associations, i.e., co-occurrence in the same context. On the other hand, higher-order association occurs between different contexts, linking contexts through items such as the value of an attribute in a database. There are two types of ARM algorithms that identify certain higher-order associations: sequential pattern mining and multi-relational ARM. Sequential pattern mining is a data mining approach that discovers frequent subsequences as patterns in a sequence database. The sequential pattern mining algorithm was introduced by Agrawal and others in [1] and [4]. In later work Mannila et al. introduce an efficient solution to the discovery of frequent patterns in a sequence database [24]. Chan et al. [10] study the use of wavelets in time-series matching and Faloutsos et al. [16]and Keogh et al. [21] propose indexing methods for fast sequence matching using R* trees, the Discrete Fourier Transform and the Discrete Wavelet Transform. Toroslu et al. introduce the problem of mining cyclically repeated patterns [28]. Han et al. introduce the concept of partial periodic patterns and propose a data structure called the Max Subpattern Tree for finding partial periodic patterns in a time series [19]. To accommodate the phenomenon that the system behavior may change over time, a flexible model of asynchronous periodic patterns is proposed in [32]. In [33], instead of frequently occurring periodic patterns, statistically significant patterns are mined. Aref et al. extend Han's work by introducing algorithms for incremental, online and merge mining of partial periodic patterns [5]. Bettini et al. propose an algorithm to discover temporal patterns in time sequences [7].

Multi-relational ARM is a type of ARM algorithm designed specifically to mine rules across tables in a single database [14]. In fact, multi-relational data mining in general (not limited to ARM) is an emerging research area that enables the analysis of complex, structured types of data such as sequences in genome analysis. Similarly, there is a wealth of recent work concerned with enhancing existing data mining approaches to employ relational logic. WARMR, for example, is a multi-relational enhancement of Apriori presented by Dehaspe and Raedt [14]. Although WARMR provides a sound theoretical basis for multi-relational ARM, it does not seriously address the efficiency of computation. In fact the runtime performance of WARMR depends heavily on the implementation of θ-subsumption, and because θ-subsumption is NP-complete, performance is poor. In addition, the model sacrifices the perspicuity of a propositional representation. In summary, existing higher-order ARM algorithms are neither capable of dealing with distributed data (particularly in the absence of knowledge of the complete schema) nor do they efficiently support 3rd and higher order record linkage.

More recently, as the need to mine patterns across distributed databases has emerged, distributed ARM algorithms have been developed. Existing distributed ARM algorithms are based on a kernel that employs either Apriori or a similar ARM algorithm based on data-parallelism [3]. Fast Distributed Mining (FDM) is based on count distribution [11]. 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. [6] 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. Noting that FDM does not scale well as the number of sites grow, Schuster and Wolff [27] propose the Distributed Decision Miner algorithm based on sampling techniques. Otey et al. [26] propose an incremental frequent itemset mining algorithm in a distributed environment which focuses on efficiently generating itemsets when the data is updated.

It is noteworthy that all of the distributed ARM algorithms we surveyed assume that the databases are horizontally distributed. This limits the applicability of these algorithms. Thus no existing distributed ARM algorithms are capable of identifying higher-order associations, while both existing distributed and higher-order ARM algorithms are unsuitable for use in a distributed environment in which the complete global schema is unknown, data is fragmented in a hybrid non-vertical, non-horizontal form, and errors occur in record linkage. In the following section we introduce D-HOTM, a Distributed Higher Order Text Mining algorithm that discovers higher-order associations in a complex distributed environment of this nature.

3 D-HOTM and DiHO ARM

In this section we present D-HOTM, which discovers rules based on higher-order associations between entities extracted from textual data based on our novel Distributed Higher Order Association Rule Mining algorithm, DiHO ARM. The outline of D-HOTM is depicted in figure 1. The first step is entity extraction based on our prior work in [31]. The approach uses a covering algorithm, which is implemented using Reduced Regular Expressions (RREs) that form rules to extract named entities from narrative textual data. After applying the entity extraction algorithm to unstructured

textual data, the items (i.e., entities) extracted populate databases local to each site that in turn become input to DiHO ARM. Each row in a given local database represents an object, which is for example a particular individual mentioned in an investigative report. In addition to the items identifying the object such as a person's name or SSN, each row also contains other items known to exist in the source document. 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, and in the case where an object is not a report but a person, different distributed sites may refer to the same object multiple times. On the other hand, the data is not vertically fragmented either, because there is no one-to-one mapping connecting 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*.

(1) Entity extraction
(2) Select linkage items
(3) Assign a globally unique ID to each record/object
(4) Identify linkable records using Apriori on global IDs
(5) Exchange information about linkable records
(6) On each site, apply the Apriori algorithm locally

Figure 1: D-HOTM

DiHO ARM comprises steps 2 through 6 in Figure 1. In step 2, the set of linkage items is selected. One requirement for this set is that the item (or combination of items) must uniquely identify objects. To exemplify the difference between an *object* and a *record*, consider the following example: given documents such as research papers, the combination of the two attributes *title* and *authors* might be selected as the linkage items because in general these two attributes together form a unique identifier for each document. On the other hand, given documents such as police investigative reports, SSN might be selected as the linkage item because such reports are written about individuals, and individuals are uniquely identified by SSN. In the former case, *title* and *authors* together uniquely represent research paper objects, whereas in the latter case, SSN uniquely represents person objects. Of course, in the latter case there are also investigative report objects, but the point is that in the latter case linkage is done using person objects, not report objects. To distinguish the usage of *object* vs. *record*, in what follows the linkage items refer to *objects*, while a *record* consists of the collection of both linkage and non-linkage items (i.e., entities) extracted from a given document (e.g., a police report). Naturally this implies that one or more of the items in a given record uniquely represent the object used for record linkage[1].

In step 3, a globally unique ID is assigned to each object and record, respectively. Step 4 discovers linkable records using the item(s) selected in step 2. For example, if a given suspect appears in a burglary record in Detroit, and the same suspect also appears in a mugging case in Philadelphia, and the SSN is chosen as the linkage item, then those two distributed records are considered linkable. In a practical sense, linking these two records might reveal new information to the investigating police officer.

[1] In [23] we develop a technique for handling errors in matching and the impact on support/confidence.

After determining which distributed records are linkable, entities are exchanged and records merged. Continuing with the same example, entities extracted from the record in Detroit are sent to Philadelphia, and vice versa. The two distributed records 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.

To further illustrate our algorithm, we provide 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 linkage item (perhaps identified by name or SSN), and the various facts such as modus operandi that surround different crimes become the fragmented non-linkage items. 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 incidents[2]. 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 DiHO ARM 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. We use rows to represent records while columns contain the entities extracted from the reports. For example, the entities "Allen" and "Gun" were extracted from the first report on site 1. Tables 1 and 2 represent two databases at different (i.e., distributed) sites.

Table 1. Relational Database on Site 1

Record	Name	Non-linkage items
1	Allen	Gun
2	Jack	Knife
3	Carol	Knife
4	Diana	Hands
5	John	Gun
6	Jack, Diana	Knife

Table 2. Relational Database on Site 2

Record	Name	Non-linkage items
7	Allen	Robbery
8	Jake	Robbery
9	Carol	Mugging
10	Bill	Burglary
11	John	Kidnapping

In step 2 of the D-HOTM algorithm in figure 1, suppose that the suspect's name is the linkage item selected for linking records. Let us further suppose that each investigative record has been assigned a unique numerical ID as shown. In this case the link-

[2] Because we are all creatures of habit, some good, some bad.

age items used to link records are {Allen, Jack, Carol, Diana, John, Bill, Jake}. As evidenced by this example, we assume that the schema for linkage items is known, in this case the "Name" attribute. Let the threshold for support be one.

The next step is to discover linkable records, and the resolution of object identifiers becomes the first task. Suppose we are using the edit distance function for the resolution of object identifiers, and the edit distance threshold is set to two. The function will reveal that "Jack" and "Jake" actually represent the same object. Let's use "Jack" as the global unique identifier; then, record eight becomes: {8, Jack, Robbery}. Now the unique global object identifiers are {Allen, Jack, Carol, Diana, John, Bill}. Based on these identifiers and the modified databases, the algorithm depicted in figure 2 discovers linkable records. Table 3 portrays the itemsets generated using Apriori on the linkage items.

Table 3. Itemsets on Global Object Ids (OIDs)

Itemsets on Global OIDs	Global OID List
Allen	1, 7
Jack	2, 6, 8
Carol	3, 9
Diana	4, 6
John	5, 11
Bill	10
Jack, Diana	6

In what follows we develop a theoretical framework for discovering linkable records per step 4 of Figure 1. Let L be the set of all linkage items. Let D^* be the set of records derived from D in which each record contains only those items that are linkage items.

Definition 1: Given *n* records $r_1, r_2, \ldots, r_n$ where $r_i \in D^*$, let M be the list of items $(a_1, a_2, \ldots, a_{n-1})$ such that $a_i \in r_i \cap r_{i+1}$. Then we say records r_1 and r_n are *n^{th}-order* linkable through M, denoted as $r_1 \sim^{a_1} r_2 \sim^{a_2} \ldots r_{n-1} \sim^{a_{n-1}} r_n$. We term M a *viable path*. For example: $r_0 \sim^a r_2 \sim^b r_3$ is a 3rd-order link between r_0 and r_3 with viable path (a, b). Also, $r_0 \sim^d r_0 \sim^a r_2 \sim^b r_3$ and $r_0 \sim^a r_1 \sim^a r_2 \sim^b r_3$ are also higher-order links between r_0 and r_3. A viable path with no repeated records or items is termed a *minimum viable path.*

Definition 2: Let the records supporting item *a* be denoted as G_a, termed a *group on a*. For a given minimal viable path $(a_1, a_2, \ldots, a_n)$, all the higher-order links satisfying the path can be written as $G_{a_1} \sim^{a_1} G_{a_1 a_2} \sim^{a_2} \cdots G_{a_{i-1} a_i} \sim^{a_i} G_{a_i a_{i+1}} \cdots G_{a_n}$. We term the set of such higher-order links a *higher-order link cluster*. A link cluster whose groups all meet the support threshold is termed a *frequent link cluster*.

Definition 3: Given all minimal viable paths corresponding to the minimal links between two records, if the minimal viable path of a given minimal link is not a super-sequence of any other minimal viable path, we term such a minimal viable path the *shortest viable path*, and the corresponding minimal link the *shortest link*. The algorithm in Figure 2 discovers the shortest higher-order links for a support threshold greater than or equal to one.

Input: D^*, L, *min_sup*, level
Output: higher-order links

Foreach item x in L, generate G_x
Broadcast and get global information on G_x
Remove item x in L if x.sup<min_sup or x.sup=1
If (level==2) exit

Generate frequent 2-itemsets and G_{xy}
Foreach frequent 2-itemset *xy*
Generate link cluster G_x-G_{xy} $\sim^x$ G_{xy} $\sim^y$ G_y-G_{xy}
Generate 3rd-order shortest links $r_0 \sim^x r_1 \sim^y r_2$
If (level=3) exit
Generate maximal k-itemsets (M_1, M_2, …, M_n) where k≥2
For any pair (M_i, M_j) where $M_i \cap M_j \neq \phi$
Foreach item a∈(M_i-M_j), b∈(M_j-M_i), c∈($M_j \cap M_i$)
If $ab \not\subset M_t$ and $ab \neq M_t$, $1 \leq t \leq n$
Generate cluster G_a-G_{ac} $\sim^a$ G_{ac} $\sim^c$ G_{bc} $\sim^b$ G_b-G_{bc}
Generate 4th-order shortest links $r_0 \sim^a r_1 \sim^c r_2 \sim^b r_3$
Discard the link if
i) b∈r_0 and a∈r_3 or
ii) r_1=r_2.

Figure 2: Using Apriori to generate up to 4th-order links

First, 2nd-order links are generated from all the 1-itemsets. Since only one record supports {Bill}, and we do not allow the same record to appear more than once in minimal higher-order links, no 2nd-order links are generated for itemset {Bill}.

In this example, only one 2-itemset exists, {Jack, Diana}, which means that this is the only itemset capable of generating 3rd-order link clusters. 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*, the groups are higher-order linked as follows: G_{Jack}-$G_{Jack,Diana}$ $\sim^{Jack}$ $G_{Jack,Diana}$ $\sim^{Diana}$ G_{Diana}-$G_{Jack,Diana.}$ Using the GIDLists from Table 3 to represent the groups on these items, we have {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 and Clusters

Itemsets	Higher-order link cluster	New Records
Allen	$1 \sim^{Allen} 7$	{1, 7}
Jack	$2 \sim^{Jack} 6$; $6 \sim^{Jack} 8$; $2 \sim^{Jack} 8$	{2, 6, 8}
Carol	$3 \sim^{Carol} 9$	{3, 9}
Diana	$4 \sim^{Diana} 6$	{4, 6}
John	$5 \sim^{John} 11$	{5, 11}
Jack, Diana	{2, 8} $\sim^{Jack}$ 6 $\sim^{Diana}$ 4	{2, 6, 4}, {8, 6, 4}

This completes step 4 of the DiHO ARM algorithm in Figure 1. Next, step 5 of Figure 1 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. The merging process can be completed in different ways. For example, given that the records {2, 6, 8} are 2^{nd}-order linkable to each other through *Jack*, we could generate three newly merged records based on the three 2^{nd}-order links. Alternatively, we could generate just one record by merging the three records together. The choice of method is an open issue, and in this example we choose the latter method. In addition, for the 3^{rd}-order link cluster {2, 8} $\sim^{Jack}$ 6 $\sim^{Diana}$ 4, one way to generate new records is by merging only the records at the start and end such as 2 and 4 or 8 and 4. An alternative is to merge all the records in the path, which will result in the new records {2, 6, 4} and {8, 6, 4}. As noted, the method of merging records is an open issue. At this point, however, we have chosen the latter method because it preserves more information about record linkage. This will naturally result in the discovery of additional association rules, but at the same time we will not lose any information. Based on the results in Table 4, after merging we obtain the new records:

{1, 7} {2, 6, 8} {3, 9} {4, 6} {5, 11} {2, 6, 4} {8, 6, 4}

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

Document	Name	Non-linkage items
1,7	Allen	Gun, Robbery
2,6,8	Jack, Diana	Knife, Robbery
3,9	Carol	Knife, Mugging
4,6	Jack, Diana	Hands, Knife
5,11	John	Gun, Kidnapping
2,6,4	Jack, Diana	Knife, Hands
8,6,4	Jack, Diana	Knife, Hands, Robbery

In step 6 of Figure 1, 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 as well as the usual first-order associations will be included in the resulting rules generated by Apriori. For example, "Diana $\Rightarrow$ Robbery" is a rule generated based on higher-order associations discovered by the DiHO ARM algorithm. This rule is obtained based on the evidence that (1) Jack and Diana are involved in some crime together; (2) Jack committed a robbery; (3) Diana might have been involved in that robbery too. This kind of higher-order rule cannot be discovered by existing distributed or higher-order association rule mining algorithms.

Before proceeding to discuss our progress in the design, implementation and testing of D-HOTM, it is important to note that although the sites in this example have identical databases in step 6 of Figure 1, this is not a requirement of the algorithm. In fact, due to various data availability constraints, it is likely that different sites will combine the new, merged records with existing local records that were not shared initially, resulting in rule sets that differ on a site-by-site basis.

4 Implementation

Based on the theoretical framework and algorithm outlined in this article we have designed, developed, implemented and conducted preliminary testing of a software system named that incorporates DiHO ARM with an entity extraction phase and data analysis capabilities in D-HOTM.

The D-HOTM system is based on the Text Mining Infrastructure (TMI) developed by the parallel and distributed text mining lab at Lehigh University [20]. Originally designed for single-processor applications, in its most recent release (version 1.3), the TMI now includes classes that can be utilized in a parallel or distributed environment based on OpenMP or MPI, now fully supporting distributed data mining.

The D-HOTM system is comprised of three top-level components. The first, used by the Investigator, is the Controls component. This serves as the front-end of the system, accepting input directly from the user to define the parameters of the D-HOTM mining job. These parameters are passed to the core of the system, the D-HOTM component itself. This component performs the distributed mine based on the DiHO ARM algorithm, contacting any necessary databases and sites, utilizing the Global Justice XML Data Model (GJXDM) for transmission of data. The system employs Borgelt and Kruse's Apriori code for local association rule mining [20]. Once the mining process is complete and results are stored locally on each node, they are passed to the Analysis component. This component provides various abilities to sort, organize, filter, and visualize the mining results.

The D-HOTM system comprises over 106,000 lines of C++ and C code. The GNU C/C++ compiler version 3.2.2 was used to compile and link the code under Red Hat Linux 9.0. In the following section, we discuss preliminary experimental results including execution time of the complete process of generating higher-order association rules on a distributed computational cluster per the D-HOTM algorithm depicted in Figure 1.

5 Experiments

We conducted experiments to evaluate D-HOTM on the National Center for Supercomputing Applications (NCSA) Tungsten Supercluster (Xeon Linux). Tungsten is composed of Intel 3.06GHz Xeon DP processor-based systems running Red Hat 9.0, with Myrinet 2000 interconnects, an I/O subcluster with more than 120 terabytes of DataDirect storage. Tungsten provides Intel 8.0 icc and GNU gcc 3.2.2 for compilation, the Load Sharing Facility (LSF) batch system for job control and ChaMPIon/Pro for the MPI runtime environment.

The test data is a simulated set for the methamphetamine trafficking process discussed in the introduction. The data consists of many categories, including names, companies and times, of which only phone numbers and addresses were used as linkage items (the rest were extracted with no particular semantics).

We tested the system on the cluster at NCSA scaling up to about 4,000 records on each of up to 128 processes on 64 nodes. The input was the aforementioned methamphetamine case data. The output was both 1st and higher-order rules. The correctness of program execution was determined by comparing the rules produced by each MPI process. In order to validate the system, the input data was designed such that each

local MPI process produced identical rules. Thus as noted, the algorithm was validated simply by comparing the resulting local rules.

Table 6: Processing Times with 4,096 Records per Node

Processes (Nodes)	Total Records	User (sec)	System (sec)	CPU (sec)
2 (1)	8,192	0.64	0.14	0.78
4 (2)	16,384	0.53	0.14	0.67
8 (4)	32,768	1.1	0.39	1.49
16 (8)	65,536	1.59	0.39	1.98
32 (16)	131,072	3.64	1.0	4.64
64 (32)	262,144	5.14	1.13	6.27
128 (64)	524,288	14.33	3.1	17.43

The user, system, and elapsed wall clock execution times of the system are depicted in Table 6. The 1st and higher-order rules generated by each MPI process were identical. The DiHO ARM algorithm successfully discovered the following higher-order link from the example in the introduction:

$$\textit{meth lab} \sim^{\textit{address/245 4th St, Chicago, IL}} \textit{Jason Carter} \sim^{\text{phone/}\textit{9052319000}} \textit{Reu Robots}$$

The algorithm also generated association mining rules based on this link and others like it, including (but not limited to) the following:

$$\textit{meth lab} \Rightarrow \textit{Reu Robots} \quad \text{and} \quad \textit{Jason Carter} \Rightarrow \textit{Reu Robots}$$

Although these results are preliminary in nature, they serve to validate the theoretical framework and the DiHO ARM algorithm.

6 Conclusions and Future Work

We have embarked on an ambitious program of research and development that addresses significant challenges in distributed data management faced by organizations such as law enforcement agencies and healthcare providers. We have identified critical assumptions made in existing association rule mining algorithms that prevent them from scaling to complex distributed environments in which the complete global schema is unknown, data is fragmented in a hybrid non-vertical, non-horizontal form, and errors occur in record linkage. We developed a theoretical framework to reason about record linkage, and a theoretical framework for evaluation metrics based on linkage matching errors[3]. We also designed, implemented and tested a distributed higher-order association rule mining algorithm, DiHO ARM, which discovers propositional rules based on higher-order associations in a distributed environment.

In our future work we plan to address both theoretical and practical issues in areas such as the utility of higher-order associations as well as record linkage, evaluation metrics and issues in efficiency of execution. Second, our current framework for reasoning about record linkage needs to be expanded in several ways. Third, metrics are needed to provide a measure of the strength or importance of higher-order links and link clusters. Finally, since both false positive and false negative mismatches are possible in the linkage item/object ID mapping process in DiHO ARM, additional theo-

[3] See [23] for details on the theoretical framework for evaluation metrics.

retical work is needed to develop suitable metrics for evaluating the utility of the resulting rules.

Acknowledgements

The authors wish to thank Lehigh University, the Pennsylvania State Police, the Lockheed-Martin Corporation, the City of Bethlehem Police Department, the National Science Foundation and the National Institute of Justice, US Department of Justice. This work was supported in part by NSF grant number 0534276 and NIJ grant numbers 2003-IJ-CX-K003, 2005-93045-PA-IJ and 2005-93046-PA-IJ. 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 National Science Foundation, the Pennsylvania State Police or the Lockheed Martin Corporation. We are also grateful for the help of other co-workers, family members and friends. Co-authors Shenzhi Li, Christopher D. Janneck, Tianhao Wu and William M. Pottenger also gratefully acknowledge the continuing help of their Lord and Savior, Yeshua the Messiah (Jesus the Christ) in our lives and work. Amen.

References

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207–216, Washington, D.C., 26–28 1993.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. Advances in knowledge discovery and data mining, pages 307–328, 1996.

[3] R. Agrawal and J. C. Shafer. Parallel mining of association rules. IEEE Trans. On Knowledge and Data Engineering, 8:962–969, 1996.

[4] R. Agrawal and R. Srikant. Mining sequential patterns. In P. S. Yu and A. S. P. Chen, editors, Eleventh International Conference on Data Engineering, pages 3–14, Taipei, Taiwan, 1995. IEEE Computer Society Press.

[5] W. G. Aref, M.G. Elfeky and A.K. Elmagarmid. Incremental, Online and Merge Mining of Partial Periodic Patterns in Time-Series Databases. IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 3, pp. 332-342, 2004.

[6] M. Z. Ashrafi, D. Taniar, and K. Smith. ODAM: An optimized distributed association rule mining algorithm. *IEEE Distributed Systems Online*, 05(3), March 2004.

[7] C. Bettini, X.S. Wang, S. Jajodia and J. Lin. Discovering frequent event patterns with multiple granularities in time sequences. *Knowledge and Data Engineering, IEEE Transactions on*, Volume: 10 Issue: 2, Mar/Apr. Page(s): 222 –237, 1998.

[8] C. Borgelt and R. Kruse. Induction of Association Rules: Apriori Implementation. In *14th Conf. on Computational Statistics*, 2002.

[9] D. Boyd. 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, September 2004.

[10] K. Chan and A. Fu. Efficient Time-Series Matching by Wavelets. In Proc. of 1999 Int. Conf. on Data Engineering, Sydney, Australia, March, 1999.

[11] Cheung, Han, Ng, Fu, and Fu. A fast distributed algorithm for mining association rules. In PDIS: International Conference on Parallel and Distributed Information Systems. IEEE Computer Society Technical Committee on Data Engineering, and ACM SIGMOD, 1996.

[12] C. Clifton, M. Kantarcioğlu, A. Doan, G. Schadow, J. Vaidya, A. Elmagarmid, and D. Suciu. Privacy-preserving data integration and sharing. In DMKD '04: Proceedings of the

9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, pages 19–26, New York, NY, USA, 2004. ACM Press.
[13] Dean M. and Schreiber G. OWL Web Ontology Language Reference. Editors, W3C Recommendation, 10 February 2004. [Online Article]. Retrieved Nov. 25, 2005 from the World Wide Web: http://www.w3.org/TR/2004/REC-owl-ref-20040210/.
[14] L. Dehaspe and L. D. Raedt. Mining association rules in multiple relations. In ILP '97: Proceedings of the 7th International Workshop on Inductive Logic Programming, pages 125–132, London, UK, 1997. Springer-Verlag.
[15] M. Elfeky, V. Verykios and A. Elmagarmid. TAILOR: A Record Linkage Toolbox. In Proc. of *the 18th Int. Conf. on Data Engineering*, 2002.
[16] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast Subsequence Matching in Time-Series Databases. In Proc. of the 1994 ACM SIGMOD Int. Conf. on Management of Data, Minneapolis, Minnesota, May, 1994.
[17] M. R. Genesereth, A. M. Keller, and O. M. Duschka. Infomaster: an information integration system. In ACM SIGMOD Conference, pages 539–542, 1997.
[18] GJXDM. Global Justice XML Data Model. [Online Article]. Retrieved Nov. 17, 2005 from the World Wide Web: http://www.it.ojp.gov/gjxdm
[19] J. Han, G. Dong, and Y. Yin. Efficient mining partial periodic patterns in time series database. Proc. ICDE, 106-115, 1999.
[20] L.E. Holzman, T.A. Fisher, L.M. Galitsky, A. Kontostathis and W. M. Pottenger. A Software Infrastructure for Research in Textual Data Mining. *The International Journal on Artificial Intelligence Tools*, volume 14, number 4, pages 829-849. 2004.
[21] E. Keogh, K. Chakrabarti, M. Pazzani and S. Mehrotra. Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. Springer-Verlag, Knowledge and Information Systems, p. 263–286, 2001.
[22] V.I. Levenstein. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Sov. Phys. Dokl. 10:707-710, 1966.
[23] S. Li, T. Wu and W. M. Pottenger. Distributed Higher Order Association Rule Mining Using Information Extracted from Textual Data. *SIGKDD Explorations*, Volume 7, Issue 1, June, 2005.
[24] H. Mannila, H. Toivonen, and A.I. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, vol. 1, no. 3, 259-289, 1997.
[25] USDOJ News. Over 65 Arrested in International Methamphetamine Investigation. 2003. http://www.usdoj.gov/dea/pubs/pressrel/pr041503.html
[26] M.E. Otey, S. Parthasarathy, W. Chao, A. Veloso and W. Meira. Parallel and distributed methods for incremental frequent itemset mining. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 34, No. 6, pp. 2439-2450, 2004.
[27] A. Schuster and R. Wolff. Communication-Efficient Distributed Mining of Association Rules. *Proc. ACM SIGMOD Int'l Conf. Management of Data*, ACM Press, 2001, pp. 473-484.
[28] H. Toroslu ana M. Kantarcioglu. Mining Cyclically Repeated Patterns. Springer Lecture Notes in Computer Science 2114, p. 83, 2001.
[29] J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 639–644, New York, NY, USA, 2002. ACM Press.
[30] B. Wilson. Meth crisis 'disastrous' for US. Citizen, Oct 2005.
[31] T. Wu and W.M. Pottenger. A Semi-Supervised Active Learning Algorithm for Information Extraction from Textual Data. *JASIST*, volume 56, number 3, pages 258-271, 2005.
[32] J. Yang, W. Wang and P. Yu. Mining asynchronous periodic patterns in time series data. *Proc. SIGKDD*, 275-279, 2000.
[33] J. Yang, W. Wang and P. Yu. Mining surprising periodic patterns. *Proc. SIGKDD*, 2001.