Mining Higher-Order Association Rules from Distributed Named Entity Databases

Shenzhi Li
Lehigh University
shl3@lehigh.edu

Christopher D. Janneck
Lehigh University
cdj2@lehigh.edu

Aditya P. Belapurkar
Lehigh University
apb204@lehigh.edu

Murat Ganiz
Lehigh University
mug3@lehigh.edu

Xiaoning Yang
Lehigh University
xiy204@lehigh.edu

Mark Dilsizian
Lehigh University
mjd204@lehigh.edu

Tianhao Wu
Lehigh University
tiw2@lehigh.edu

John M. Bright
Lehigh University
jmbi@lehigh.edu

William M. Pottenger
Rutgers University
billp@dimacs.rutgers.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. 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. In this paper, two approaches to the definition and discovery of higher order itemsets are presented. **The implementation of D-HOTM is based on the TMI [20] and tested on a cluster at the National Center for Supercomputing Applications (NCSA). Results on a real-world dataset from the Richmond, VA police department demonstrate the performance and relevance of D-HOTM in law enforcement and homeland defense.**

I. 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 [32]. The United States Drug Enforcement Administration (DEA) has conducted several operations to investigate the entire methamphetamine trafficking process. In 2003, the DEA and the Royal Canadian Mounted Police announced the arrests of over 65 individuals in ten cities throughout the United States and Canada in an international methamphetamine investigation [26]. The arrests were the result of an 18-month international investigation using manual higher-order association techniques that linked distributed documents through addresses, phone numbers, etc.

Figure 1 depicts an example of the discovery of higher-order associations in methamphetamine trafficking. The three records are distributed in different databases. The underscored named entity in record 1 on site 1 reveals the address of a broker involved in selling precursor chemicals to a meth lab in LA. This same address is extracted from record 2 on site 2, linking to a second named entity – the phone number – of a suspect broker named Jason Carton. This same phone number is extracted from record 3 on site 3, revealing the link to a Canadian chemical company that produces pseudoephedrine,

a precursor chemical used in meth production. Using the address and phone number, Reu Robots, the supplier of pseudoephedrine, can be linked to Jason Carton, a chemical broker, who in turn is linked to the producer in LA. Linking the three records through the address and the phone number results in the rule "*meth lab ⇒ Reu Robots*" based on the higher-order association {meth lab, Jason Carton, Reu Robots}. This is precisely the kind of information that investigators need. No existing ARM algorithms are capable of producing rules of this nature in a distributed environment.

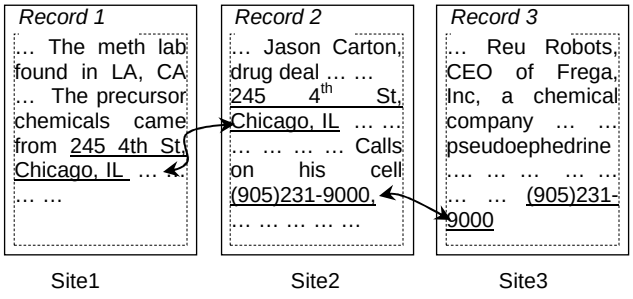

Figure 1 . An Example of Higher-Order Association

To identify rules based on higher-order associations in a distributed environment, another challenge must be considered also – data fragmentation. 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 a distributed environment. 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. 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 [9]. Indeed, efforts are underway to establish standards in schema integration (e.g., OWL [13], GJXDM [18]). Nonetheless, even should there be widespread acceptance of such standards, the ability to integrate schemas automatically is still an open research issue [17].

Currently, there are no ARM algorithms capable of mining distributed higher-order associations. Existing ARM algorithms for mining distributed data are capable of mining only data that is either horizontally or vertically fragmented [11][28][31]. 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 textual mining (D-HOTM) framework that (1) provides a theoretical basis for higher order itemsets generation and evaluation; (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 two approaches to discover higher order itemsets based on different definitions. The D-HOTM framework design and implementation is discussed in section 4. We present results in section 5, and close with conclusions and future work in section 6.

II. 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 [25]. 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 [29]. 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 [34]. In [35], 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 pro-vides 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 count distribution 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 [28] propose the Distributed Decision Miner algorithm based on sampling techniques. Otey et. al. [27] 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 the concept of latent itemsets which capture the higher order association among items. The support calculation for latent itemsets is also proposed.

III. Approach

The first step in higher order association rule mining is to discover higher order itemsets. From higher order itemsets, higher order association rules can be generated. In this section, we present two different approaches to the discovery of higher order itemsets. First, however, we must provide a theoretical framework on which to base higher order itemset discovery. We begin with two different definitions in order to explore the space of higher order itemsets: latent higher order itemsets and explicit higher order itemsets. In the following, we introduce definitions and the approaches based on them.

A. Latent Higher Order Itemset Mining (LHOIM)

Latent higher order itemsets are formed by linking different contexts through common item(s), referred to as *linkage items*. As in our prior work with Latent Semantic Indexing [22], we leverage the latent information in higher order connections, and thus refer to higher order itemsets as *latent.* In the following, we will first precisely define *higher order association*, and then present our approach to the generation and evaluation of latent itemsets.

If item *a* and item *b* from different transactions can be associated across *n* distinct records, then items *a* and *b* are n^{th}-order associated, denoted as $a \sim^{r_1} i_i \sim^{r_2} i_2 \sim \ldots \sim^{r_{n-1}} i_{n-1} \sim^{r_n} b$ where ~ represents the co-occurrence relation and *i* is termed a linkage item. The order of a higher-order association is determined by the number of distinct records *n*. This definition allows each record to occur at most once in a given transitive link. Otherwise, cyclical links are possible such as $a \sim^{r_1} i_i \sim^{r_1} i_2 \sim^{r_1} a$, which allows an item to be linked to itself at any order. This constraint is also necessary to be consistent with the original ARM framework. For example, given a higher-order association $a \sim^{r_1} i_i \sim^{r_2} i_2 \sim^{r_1} b$, based on definition 1, *a* and *b* are 2nd-order associated because there are two distinct records in the link. This conflicts however with the fact that *a* and *b* actually are 1st-order associated since they both come from r_1. And higher-order links with repeated records can always be shortened into a higher-order link per definition 1.

Latent itemsets are itemsets in which item pairs may be associated by orders of one or higher. For example, the itemset *abk*, formed from the higher order link $a \sim^{r_1} b \sim^{r_2} c \sim \ldots \sim^{r_{n-1}} f \sim^{r_n} k$, is *latent higher order* associated: *ab* is 1st-order associated, *bk* is $n\text{-}1^{th}$-order associated, and *ak* is n^{th}-order associated. Due to these associations, *abk* is a latent itemset.

Considering that for a given higher order link $a \sim^{r_1} b \sim^{r_2} c \sim \ldots \sim^{r_{n-1}} f \sim^{r_n} k$, many other higher order links share the same record sequence. The number of such links is $\prod_{i=1}^{n-1} (| r_i \cap r_{i+1} |)$. The latent itemsets generated from all these links actually are the same as the subsets generated from the union of the records. Thus, instead of dealing with a bunch of higher order links, we choose to use the record sequence, referred as *link group*, to accomplish the same goal.

A *link group* is a group of higher order links between records which have the same record sequence. Similar to Definition 1, we define the higher order links between records r_1 and r_n as: $r_1 \sim^{a_1} r_2 \sim^{a_2} \ldots r_{n-1} \sim^{a_{n-1}} r_n$, where a_i is a linkage item. And the link group is written as $r_1 \sim^{I_1} r_2 \sim^{I_2} \ldots r_{n-1} \sim^{I_{n-1}} r_n$ where $I_j = r_j \cap r_{j+1}$. To simplify this notation, henceforth, we use $r_1 \sim r_2 \sim \ldots r_{n-1} \sim r_n$ to represent a link group. Clearly, the latent itemsets generated from the higher order link $a \sim^{r_1} b \sim^{r_2} c \sim \ldots \sim^{r_{n-1}} f \sim^{r_n} k$ could also be discovered from the link group $r_1 \sim r_2 \sim \ldots r_{n-1} \sim r_n$. The *size* of a link group is defined as the product of the sizes of I_j. For example, given a 3rd-order link group L_g:

To generate the context of latent itemsets in link groups, each record could be mapped to a node, and edges mapped to common shared items, then the problem of finding all link groups reduces to finding all simple paths between two vertices in a graph. Finding simple paths between two vertices is solved using a backtracking technique. The latent itemsets are then generated from the merged records of the link groups.

Suppose latent itemset *A* is generated from link group $r_1 \sim r_2 \sim \ldots r_{n-1} \sim r_n$ with size *s*. Based on a straightforward application of Apriori, the support of *A* would be the frequency of occurrence in the link group, i.e., *s*. This approach presents two challenges: it results in very large support values, and it ignores the effect of the order. To address these issues, in what follows a metric is presented to calculate support for latent itemsets which leverages both the size of the link group and the order.

Let *L(A)* be the set of link groups which contain the latent itemset *A*. We define the *support* of a latent itemset *A* as:

$$\sum_{l \in L(A)} \frac{\log_{10}(l.size + 1)}{l.order} \quad (1)$$

The idea behind this *global* support is simply to account for both the number of higher order links supporting a given latent itemset as well as the order of the itemset. As order grows, intuition suggests that support ought to decrease – thus the denominator *l.order*. This reflects the assumption that the longer the link between records, the weaker the itemset association. In contrast, intuition also suggests that the more link groups that contain a given itemset, the stronger the support should be. These two intuitions are just that – certainly, extensive experimentation is required to ascertain the utility of this definition of support. Nonetheless, our preliminary results are quite encouraging **Error! Reference source not found.**. The challenge arises when one considers the exponential growth of the link groups' sizes given that order grows linearly. Again, intuition suggests that both of these factors are equally important. Thus, in order to constrain *l.size* to grow linearly with order, the $\log_{10}$ is taken. Also, one is added to *l.size* in the numerator to ensure that the argument to $\log_{10}$ is non-zero.

Based on the framework discussed above, an algorithm to discover latent itemsets in presented in what follows.

Latent Itemset Mining
Input: *D, L, max_order, minsup*
Output: latent itemsets
1. Form adjacency list
2. Generate connected sub-graphs
3. for each G
4. Enumpath(G, *max_order*)
5. for each n^{th}-order linkgroup *lg:* $r_1 \sim r_2 \sim \ldots r_{n-1} \sim r_n$
6. $\boldsymbol{r} = \cup \boldsymbol{r_i}$; *r.order = n; r.size = lg.size*
7. add *r* to R
8. L_1 = {supported 1-itemsets};
9. for (k=2; L_{k-1}!=null; k++)
10. C_k = apriori-gen(L_{k-1});
11. for each *r* do
12. C_t = subset(C_k, r)
13. for all candidates c$\in C_t$
14. c.count+=log10(r.*size*+1)/r.order
15. L_k = { c$\in C_k$ | c.sup $\geq$ minsup }
16. Result = $\cup L_k$

Figure 2: Latent Itemset Mining Algorithm

The first step is to form an undirected graph from the input records based on the user's choice of entities. This graph is then split into disjoint subgraphs, which is input to Enumpath in step 4. Enumpath employs the algorithm in [30] to find all simple paths (i.e., link groups) between two vertices. The worst case time complexity of this step is $O(|V||E|)$ for a given path where V is the set of vertices and E the edges in G. Steps 5 to step 7 in Figure 2 generate merged records from each link group. Steps 8 to 16 discover the frequent latent itemsets. The latent itemsets which meet the support threshold become the frequent latent k-itemsets used to generate the latent k+1-itemsets. Except for the support calculation, this level-wise process is similar to that of Apriori.

B. Explicit Higher Order Itemset Mining

Latent itemsets implicitly include itemsets of different orders. Explicit itemsets, on the other hand, maintain clear boundaries between itemsets of different orders. An n^{th}-order explicit higher order itemset is an itemset for which each pair of items is n^{th}-order associated. For example, if *abc* is a 3rd-order itemset, then there must exist at least three 3rd-order associations between a and b, b and c and a and c respectively. The context of explicit k-itemsets is defined as a k-recordset where there exists at least one n^{th}-order link group between each record pair. Thus, an n^{th}-order explicit higher order itemset $i_1 i_2 \ldots$supported by an n^{th}-order recordset $r_1 r_2 \ldots r_n$ contains no two items from the same record.

Similar to link groups, the size of a recordset is calculated by taking the product of the sizes of each link group. It is important to note that a given recordset might be composed of different link groups. This may occur for n[th]-order recordsets when n is greater than two. In this case, given j instances of n[th]-order k-recordset *rs*, its size is defined as: $size_{n\_k}(rs) = \sum_{u=1}^{j}(\prod_{v=1}^{k(k-1)/2} \lg_v .size)$. Given the sizes for all recordsets, the support of a k-itemset is defined as in equation (2) below as:

$$\sup_k(is) = \sum_{t=1}^{max\_order} \frac{\log_{10}\sqrt{\sum size_{n\_k}(rs)+1}}{t}$$

This metric is similar to Equation 1 – the global support is calculated by adding the local support at each level. The local support is also designed based on the intuition that the size of the recordsets should be of same importance as the order. To constrain $size_{n-k}(rs)$ to grow linearly with order, first the square root of $size_{n-k}(rs)$ is taken and then the $\log_{10}$. The square root accounts for the $O(n^2)$ growth of number of edges in a recordset as order grows; the $\log_{10}$ accounts for the exponential growth of $size_{n-k}(rs)$. As before, one is added to $size_{n-k}(rs)$ in the numerator to ensure that the argument to $\log_{10}$ is non-zero.

Based on the framework discussed above, an algorithm to discover explicit higher order itemsets is presented in Figure 3. EHOIM is structured in an order-first level-wise manner. Level-wise means that the size of k-itemsets increases in each iteration (as is the case for Apriori), while order-first means that at each level, itemsets are generated across all orders. The EHOIM algorithm is presented in Figure 3. In addition to the notation used in LHOIM, EHOIM uses RS_k to represent the set of *k*-recordsets. Each member has three fields: recordset, order and size. RS_{n_k} is used for the set of n[th]-order k-recordsets; each member has two fields: recordset and size. Similarly, we use IS_k for the set of k-itemsets where each member has the global support of the corresponding itemset. IS_{n_k} is used for the set of n[th]-order *k*-itemsets, where each itemset has its own local support.

Explicit Higher Order Itemset Mining
Input: *D, L, maxorder, minsup*
Output: higher-order itemsets

1. Form Adjacency List
2. For each pair of vertices (*x,y*) in *G*
3. Enumpath(*G, x, y, maxorder*)
4. For each nth-order association group *l*:
   $r_1 \sim r_2 \sim \cdots r_{n-1} \sim r_n$
5. *rs*=(r_1, r_n), $rs.size = \prod_{i=i}^{n-1}(|r_i \cap r_{i+1}|)$,
   rs.order = *n*,
6. if RS_2(*rs*, *rs*.order) is valid
7. RS_2(*rs*, *rs*.order).size += *rs*.size
8. else RS_2(*rs*,order)=size
9. For (k = 3; $RS_{k-1} \neq \phi$; k++)
10. For (n = 2; n < *maxorder*; n++)
11. RS_{n_k} = Gen_RS(RS_{n_k-1});
12. For each recordset *rs*$\in RS_{n\_k}$
13. Enum_IS(*rs*, n);
14. For each itemset is where |*is*|=k
15. IS_k(is).sup= $\sum_{t=2}^{max\_order} \log_{10}\sqrt{IS_{n\_k}(is).sup+1}/u$
16. Answer=Answer$\cup${is|IS_k(is).sup>=*minsup*}

Figure 3: EHOIM Algorithm

The first three steps of EHOIM are the same as LHOIM – the generation of link groups. For each link group, steps 5-8 in Figure 3 generate the corresponding 2-recordsets, calculating the size at a given order and storing it in RS_2. Steps 9 through 16 comprise one outer and two inner loops. The outer loop proceeds in a level-wise manner and keeps track of the sizes of recordsets. Although the (k+1)-recordsets are generated in an Apriori-like fashion based on k-recordsets from the previous iteration, no pruning is performed for recordsets. Step 11 generates the n[th]-order k-recordsets based on the n^{th}-order (k-1)-recordsets using Apriori's candidate generation ability. The size of the recordset is calculated based on the equation for $size_{n-k}(rs)$ above. For each n^{th}-order *k*-recordset generated, step 13 enumerates all possible n[th]-order k-itemsets from the recordset. Steps 14 and 15 calculate the global support for a single k-itemset across orders from two to maxorder based on the support in Equation 2. If the global support meets the threshold, the k-itemset is added to the final output in step 16.

VI. D-HOTM Framework

In this section, we outline the Distributed Higher Order Text Mining framework, which discovers rules based on higher order itemsets of entities extracted from textual data. The D-HOTM system is composed of entity extraction and association rule mining phases. More detail is in [24].

The entity extraction phase of D-HOTM is based on [33]. The technique employed by these authors, termed RRE Discovery, discovers reduced regular expressions for use in information extraction. The 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 algorithm first ascertains the most frequently appearing element of a reduced regular expression (RRE) which is called the *root* of the RRE. It then broadens the scope of the RRE in 'AND', 'GAP', and 'Start/End' learning phases. (See figure 3 in [33].)

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 our distributed latent itemsets mining algorithm. 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 social security number, each row also contains other items known to exist in the source document. It is clear that this distributed data is not horizontally fragmented because there is no guarantee that every site will include the same set of items. 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 distributed data is neither vertically nor horizontally fragmented, but is present in a form we term a *hybrid fragmentation*.

The D-HOTM framework provides different options for sharing records between databases in a distributed environment. The first is the traditional approach in which all records are fully shared and the same model is built on each site. In this approach, the final model at a given node is based on both local and remote data. Alternatively, on a given node D-HOTM can use remote data for higher order link generation, but filters remote records when generating itemsets. This enables a better local model to be built while respecting data privacy concerns. Finally, different sites can use different linkage items, again resulting in different local models.

The D-HOTM system is based on the Text Mining Infrastructure (TMI) developed by the authors [20]. Originally designed for single-processor applications, in its most recent release (version 1.3), the TMI now includes support for mining in parallel or distributed environments based on OpenMP or MPI.

V. Experimental Results and Evaluation

The D-HOTM system and algorithms were evaluated on two real-world data sets, one from a Richmond, VA police department database and one from an online e-commerce site, Gazelle.com. The Richmond, VA dataset currently contains over two hundred thousand records, each with about 70 fields. We broke the dataset into geographic neighborhoods for analysis, and executed D-HOTM on subsets of each neighborhood. In particular, we split each neighborhood into two parts, one containing records of crimes committed prior to 2003 and the other containing crimes from 2004 through 2006. We created a ground truth by applying the standard (1st-order) association rule mining algorithm Apriori to the three year portion after 2003, and obtained name-crime 2-itemset pairs. Following this, for each neighborhood tested, we executed D-HOTM on the first half of the data in order to predict name-crime pairs that would emerge as 1st-order 2-itemsets in the following three year period. Different linkage items were used in order to explore the quality of different models. This is because using more linkage items generally results in a better model. Figure 4 depicts the results of executing D-HOTM on four of the largest neighborhoods in the Richmond, VA area: Church Hills North, Gilpin, Jeff David and Shockoe Bottom. For each neighborhood we conducted six experiments, four global and two privacy preserving. As noted, the experiments involved the use of different linkage items. In addition, all experiments were based on the use of associations up to 4th-order.

The first experiment, HV, used *home address* and *vehicle ID* as the linkage items. Following this, HVI added *document ID* (some records have the same document ID). HVO added *occupation*, as did HVOI. Several trends can be seen in Figure 4. First, it is clear that D-HOTM correctly predicted increasing numbers of name-crime pairs as more linkage items were used. This can be seen by comparing the experiments that used two linkage items such as HV vs. those that used more (such as HVI). This result is not surprising and in fact is expected. A second important trend revealed in Figure 4 is the increase in recall of name-crime pairs as order increases. This trend is exhibited almost without exception regardless of the linkage items chosen. For example, all four neighborhoods showed an increase over 1st-order performance for HVO and HVOI, in many cases showing improvement right through 4th-order. This is a very significant result as it demonstrates conclusive evidence for the value of higher order association rule mining. Figure 4 also depicts the results for experiments D-HV and D-HVI in which the models were constructed by leveraging remote data during higher order record linkage, but only local data was used to generate rules. In this case too the results reveal a trend of increasing performance as order increases. Although the evidence is not as strong, still for three of the four neighborhoods, the trend is clear. Thus for two different approaches to constructing models, one privacy preserving, higher order record linkage improves performance of name-crime pair prediction.

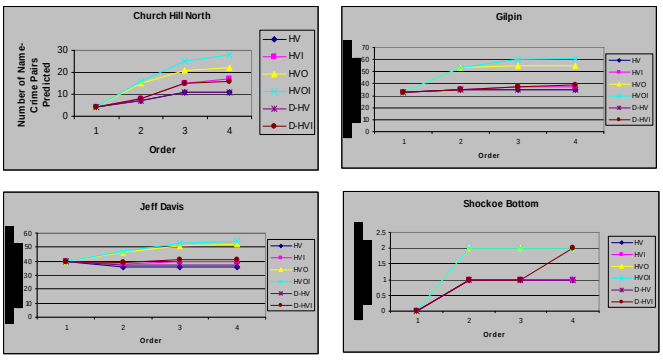

Figure 4: Predicted Name-Crime Pairs

In a second set of experiments with explicit higher order itemset mining (EHOIM), we employed a real-world dataset from the KDD Cup 2000 (Kohavi et al., 2000) competition. The data is e-commerce transactions from Gazelle.com, a now-defunct website for selling socks, pantyhose, etc. This dataset is of particular interest to us because it has proven difficult to model in prior work published on KDD Cup 2000. This may be due, for example, to the sparse nature of the transaction data.

In total, there are 530 transactions involving 244 products in the dataset. Of these, we randomly selected sets of 50, 75, 100, and 200 transactions as well as using the full 530 transaction dataset. In order to evaluate the EHOIM algorithm, we compared the itemsets generated by EHOIM with two other algorithms: Apriori (1^{st}-order) and Indirect (2^{nd}-order) (Tan et al., 2002). The EHOIM algorithm was limited to 6^{th} order.

Three different evaluation methods were employed to demonstrate the utility of higher order associations. First, in order to demonstrate the intrinsic value of higher-order itemsets, we conducted experiments that show that EHOIM discovers support for itemsets in small datasets that is only discoverable by Indirect or Apriori with larger datasets. In other words, we demonstrate the ability of EHOIM to more accurately calculate support for known good itemsets. Second, we demonstrate EHOIM's ability to discover novel itemsets, undiscovered by either Indirect or Apriori. These novel itemsets are either examples of associations that are unique to higher-orders or are examples of relationships between categories of products on Gazelle.com that are unique to higher orders. We explore these relationships using qualitative anecdotal evidence that illustrates the usefulness of the approach.

As revealed in Figure 5, results based on multiple runs of randomized data consistently demonstrated that high support itemsets could be discovered by EHOIM using smaller datasets than required by Apriori or Indirect to discover the same itemsets. Because our algorithm leverages additional, latent information, it can provide more accurate support calculations. We first ran Apriori on the entire set of 530 records, and discovered 40 itemsets with support larger than two while 145 itemsets had support larger than one. These itemsets act as our ground truth dataset[1]. In each data series, we randomly selected 50 records as the first test set, then added randomly selected records to bring the total to 75, and so on for the 100 and 200 record sets. Then, the Apriori and EHOIM algorithms were applied on each dataset respectively. To compare the higher order itemsets generated by EHOIM with the ground truth itemsets discovered by Apriori, we chose the top-N itemsets ranked in order of support from high support to low support. In the generated higher order itemsets, for example, we selected about 45 itemsets to compare with the top 45 itemsets in the ground truth. This particular number was selected to include all itemsets with the same frequency as the 45^{th} itemset in the ground truth. The same method was applied when comparing EHOIM's results with the top 145 itemsets of the ground truth. The recall comparisons are portrayed in Figure 5 for orders up to six. From these recall charts, we draw the following conclusion: for most sample datasets (sizes 50 to 200), higher order (especially 3^{rd} and higher) results in higher top-N recall of highly-ranked 1^{st}-order itemsets than the 1^{st}-order Apriori algorithm. This is a significant result in that it supports our thesis that higher order associations reveal not only novel relationships but also discover useful knowledge in smaller datasets than required by 1^{st}-order methods.

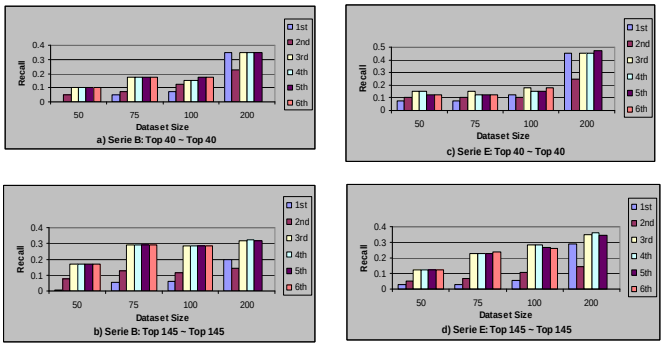

Figure 5: Recall Charts on Gazelle.com Data

For comparison purposes we also applied LHOIM on Gazelle.com data and achieved similar results for the 200 record dataset, but little to any benefit in the smaller sets. Interestingly, the Gazelle.com experiments reveal the different characteristics of LHOIM and EHOIM. Clearly, EHOIM discovers more hidden associations than LHOIM. In particular, EHOIM discovers latent information within smaller datasets that LHOIM misses. On the other hand, the time complexity of the LHOIM algorithm is significantly less than that of EHOIM.

VI. 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 that defines higher order itemsets and their corresponding contexts. In addition, the traditional definition of support was extended and two algorithms were developed corresponding to the

[1] Although these values of support are extremely low, they are in fact the highest support itemsets in the Gazelle.com dataset.

definitions of support. We also designed, implemented and tested a distributed higher order association rule mining framework, D-HOTM, 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 D-HOTM, additional theoretical 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 co-workers, family members and friends. Co-authors S. Li, C. D. Janneck, T. Wu and W. 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]. A. Kontostathis and W. M. Pottenger. A Framework for Understanding LSI Performance. *Information Processing & Management*, Volume 42, Issue 1, Pages 56-73. January, 2006.

[23]. V.I. Levenstein. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Sov. Phys. Dokl. 10:707-710, 1966.

[24]. S. Li, T. Wu and W. M. Pottenger. Distributed Higher Order Association Rule Mining Using Information Extracted from Textual Data. *SIGKDD Explorations*, vol. 7:1, June, 2005.

[25]. 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.

[26]. USDOJ News. Over 65 Arrested in International Methamphetamine Investigation. 2003. http://www.usdoj.gov/dea/pubs/pressrel/pr041503.html

[27]. 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.

[28]. 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.

[29]. H. Toroslu and M. Kantarcioglu. Mining Cyclically Repeated Patterns. Springer Lecture Notes in Computer Science 2114, p. 83, 2001.

[30]. T. UNO. An Output Linear Time Algorithm for Enumerating Chordless Cycles. *92nd SIGAL of Information Processing Society Japan*, 47-53, 2003.

[31]. 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.

[32]. Wilson. Meth crisis 'disastrous' for US. Citizen, Oct 2005.

[33]. 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.

[34]. J. Yang, W. Wang and P. Yu. Mining asynchronous periodic patterns in time series data. *Proc. SIGKDD*, 275-279, 2000.

[35]. J. Yang, W. Wang and P. Yu. Mining surprising periodic patterns. *Proc. SIGKDD*, 2001.