UNIC: UNique Item Counts for Association Rule Mining in Relational Data

Christopher Besemann and Anne Denton
North Dakota State University
Department of Computer Science
Fargo, North Dakota 58105, USA
christopher.besemann,anne.denton@ndsu.nodak.edu

Abstract

Association rule mining (ARM) can be generalized to relational data by using joined relations as basis. We demonstrate that typically such an approach results in an overwhelming number of rules that reflect nothing but trivial properties of the data. Worse, even rules that appear interesting may be due to combinations of rather general statistical properties of the data. We introduce an ARM algorithm, UNIC, that systematically excludes any influence of correlations among items that represent the same real-world quantity but belong to different entities. The concept of UNIC is to base the ARM algorithm on items that are unique to only one entity instance of a joined relation. This strategy is highly effective at eliminating undesired contributions to rule metrics like support and confidence, while achieving most pruning even before frequent item sets are computed.

1 Introduction

While most commercially relevant data are relational, appropriate data mining techniques are still lagging behind their single-table counterparts. One type of relational data set has traditionally received particular attention, albeit under a different name. A relation that defines the relationship between entity instances of the same type, also called a reflexive relationship, can be viewed as the definition of a graph. Graphs have been used to represent social networks, biological networks, communication networks, and citation graphs, just to name a few.

A variety of techniques have been developed for data mining of relational data [11]. A typical approach is to allow maximum flexibility to the user through inductive logic programming [10, 12]. Although such an approach can in principle resolve any problem a user may be aware of, it may fall short of identifying even simple pitfalls. For the particular type of data of interest in this paper, namely an entity with a reflexive relationship, it is straight forward to apply association rule mining by simply joining relations [22]. The problem of this approach only becomes obvious upon further study of the rules: In most cases the output is dominated by rules that involve the same item in related entity instances. Leaving the resolution of this problem to users is very dangerous, since a user with little experience would be tempted to filter the rule output. As we will demonstrate below such an approach leads to highly undesirable results.

To motivate our approach let us take an example of a social network. Assume a database that stores information related to criminal records. Relationships between people, based on the existence of at least one telephone conversation, are also stored. Let us further assume applying an association rule mining algorithm on the joined relation of persons together with the persons with whom a relationship exists. Doing so naively would be likely to return many rules involving a single person such as, that somebody who is known to use ”drugs” is likely to also be convicted for ”theft”. Such rules can be found based on a single person relation and, thereby, do not require relational association rule mining. In fact, since their support and confidence based on the joined table differs from single table results it has been concluded [8] that they should not be listed for the joined table. We call such rules out-of-scope.

It has been observed that in many settings items in ARM [24] or attributes in classification [18] are highly correlated across relationships. In the above example it will probably not be a surprise if a person who uses ”drugs” is likely to communicate with another person who also uses ”drugs”. Testing such a correlation does not require association rule mining, and frequent association rules that involve the same item are likely to dominate the output of a standard ARM algorithm [22]. We will call such rules repetitious. One could consider eliminating the problem by filtering the output. This would, however, lead to unacceptable performance, and worse, inconsistent results. Consider finding the rule that a person who uses ”drugs” is likely to communicate with a person convicted for "theft". Is this an interesting rule? The best answer is that we do not know: It could be that most or all occurrences that lead to this rule are due to drug-users communicating with each other, and also being convicted for theft. In this case the rule would be redundant to one out-of-scope and one repetitious rule. Checking for redundancy alone would, however, still not resolve the problem. There may be rules that are not entirely redundant. Should those rules be listed with support and confidence that include contributions from combinations of out-of-scope and repetitious rules? Most users would probably assume that if entire classes of rules are eliminated then so is their effect on other rules.

This problem is not minor. In the data we considered, protein annotations within a protein-protein interaction graph, we found that typically over 99% of rules that followed from a standard application of association rule mining were direct results of combinations of out-of-scope and repetitious rules. We resolve the problem by completely eliminating items that occur multiple times within a relational transaction. In section 4 we show that this procedure corresponds with high accuracy to the elimination of contributions to support and confidence of probabilistically independent out-of-scope and repetitious rules. Computationally our approach is efficient since all pruning is moved to the relational joining process.

The organization of the paper is as follows: Section 2 introduces our theoretical framework, section 3 discusses related work, section 4 compares our approach with a probabilistic interpretation and section 5 discusses our implementation and results.

2 Unique item count association rules

Association rule mining is commonly defined and implemented over sets of items. For our problem we need the relational algebra framework to manipulate data from multiple relations and therefore choose an extended relational model similar to [15] for our description to account for both requirements. Attributes within this model are allowed to be set-valued, thereby violating first normal form. We go one step further by allowing sets of tuples, i.e., relations, as attribute values.

Consider a database with entity relations $R_e(T, D)$ where T is a tuple identifier and D is a set of descriptors. Tuples in R_e have the form $< t_i, D_i >$ where D_i is a relation of descriptors $< d_j >$. Descriptors are tuples with just one attribute of domain $\mathcal{D}$. We call the $< d_j >$ descriptors to distinguish them from items. Items have a second attribute to identify their entity instance of origin. We will call the sets of items that form the basis for association rule mining *basis set*s. The terms "set of tuples" and "relation" will be used interchangeably. Multiple different entity relations can be used with minor modifications provided they share the same set of descriptors $\mathcal{D}$. Note that joining two or more entity relations with different sets of descriptors does require some modifications to the theory laid out below. In the following we assume that a single entity relation is used in joins.

Definition 1 A *single-entity basis set* is identical to a set of descriptors $D_i \subseteq \mathcal{D}$. This definition is equivalent to typical definitions used in association rule mining [1].

Our goal is to mine relational basis sets that will be constructed from multiple descriptor sets that belong to the same tuple of a joined relation. Relations representing relationships have two attributes $R_R(T_l, T_r)$, with T_l as well as T_r being foreign keys that refer to identifiers in one or more entity relations. Relationships can, in principle, be represented by $R_R(T_l, T_r, D^{(R)})$ with $D^{(R)}$ being a set of relationship descriptors. We would split such a relation into a separate entity relation as well as a standard relation $R_R(T_l, T_r)$ as in [8].

Joined relation basis sets are formed in multiple steps. Relationship- and entity relations are joined through a natural join operation ($*$). Attribute names are changed [13] such that they are unique. We use this step to ensure that information about the origin of different attributes is maintained. Attributes are identified by consecutive integers, to which we will refer as origin identifiers $g \in \mathcal{G} = \{0, ..., (n-1)\}$ where n is the number of entity relations. This information will be used in a later step to actually modify the descriptors according to their origin before basis sets are constructed from multiple descriptor sets.

Definition 2 A *joined-relation basis set* is derived through the following steps. A 2-entity joined relation is created through

$$\begin{aligned} R_{2e} \quad \leftarrow \rho_{0.T,0.D}(R_e(T, D)) * \rho_{0.T,1.T}(R_R(T_l, T_r)) \\ * \rho_{1.T,1.D}(R_e(T, D)). \qquad (1) \end{aligned}$$

Generalization to n-entity joined relations is straight forward. Note, however, that we can have multiple alternatives. For four entities in an undirected setting we have the following alternatives as joined relation

$$\begin{aligned} R_{4el} \quad \leftarrow \quad & \rho_{0.T,0.D}(R_e(T, D)) * \rho_{0.T,1.T}(R_R(T_l, T_r)) \\ * \quad & \rho_{1.T,1.D}(R_e(T, D)) * \rho_{1.T,2.T}(R_R(T_l, T_r)) \\ * \quad & \rho_{2.T,2.D}(R_e(T, D)) * \rho_{2.T,3.T}(R_R(T_l, T_r)) \\ * \quad & \rho_{3.T,3.D}(R_e(T, D)) \qquad (2) \\ R_{4eg} \quad \leftarrow \quad & \rho_{0.T,0.D}(R_e(T, D)) * \rho_{0.T,1.T}(R_R(T_l, T_r)) \\ * \quad & \rho_{1.T,1.D}(R_e(T, D)) * \rho_{1.T,2.T}(R_R(T_l, T_r)) \\ * \quad & \rho_{2.T,2.D}(R_e(T, D)) * \rho_{1.T,3.T}(R_R(T_l, T_r)) \\ * \quad & \rho_{3.T,3.D}(R_e(T, D)). \qquad (3) \end{aligned}$$

Attribute renaming $\rho_{A_0 \ldots A_n}$ is used as defined in [13]. We then apply a Cartesian product of a relation consisting of a

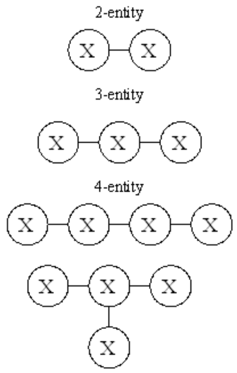
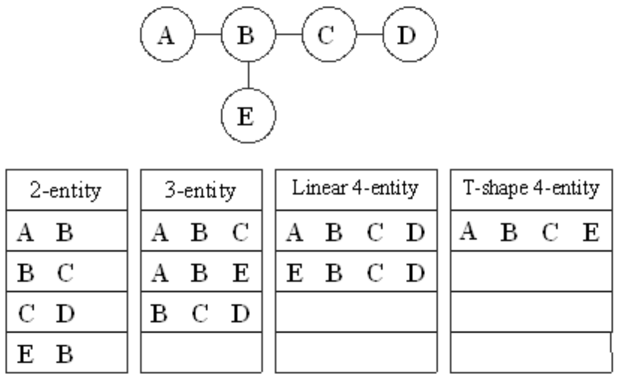


Figure 1. Item set neighborhood shapes



2-entity	3-entity	Linear 4-entity	T-shape 4-entity
A B	A B C	A B C D	A B C E
B C	A B E	E B C D	
C D	B C D		
E B			

Figure 2. Types of basis sets

single tuple containing the origin identifier $< g >$ with each set of descriptors individually. It converts the descriptors $< d_j >$ into tuples $< g, d_j >$ that we call *items*

$$\begin{aligned} g.I_i &= \ < g > \times \{< d_0 >, ..., < d_k >\} \\ &= \ \{< g, d_0 >, ..., < g, d_k >\}. \end{aligned} \quad (4)$$

g is the same origin identifier that is used as prefix in the attribute name. Note that we will use an abbreviated notation for items in the results section ($g.d_j$ instead of $< g, d_j >$). A joined-relation basis set B_i is derived as the union of descriptor sets for each tuple identified by t_i in the joined relation. For a 2-entity joined relation basis set we have

$$\forall t_i \ \ B_i = 0.I_i \cup 1.I_i. \quad (5)$$

where $g.I_i = \{< g, d_0 >, ..., < g, d_k >\}$ is the set of items of the tuple identified by t_i. The set of all basis sets is $C = \{B_0, ..., B_m\}$ where m is the number of tuples in the joined relation.

Definition 3 A *UNIC operator* U is defined as follows. For each set-valued attribute on which it operates the set difference is computed between that attribute and the union of all other attributes of that domain.

$$\begin{aligned} U(R_{ne}): \qquad & \forall t_i \ \forall_{j=0}^{(n-1)} \ j.I_i^U = \\ & j.I_i - \bigcup_{k=0, k \neq j}^{(n-1)} k.I_i \end{aligned} \quad (6)$$

with $g.I_i$ defined as in definition 2. In this paper the UNIC operator is applied to all set-valued attributes of a joined relation. Other choices are possible, such as requiring uniqueness only across a subset of relationships.

Definition 4 A *UNIC item basis set* is defined through the following steps. A n-entity joined relation is created as described in definition 2. The UNIC operator is applied to all set-valued attributes. Then the Cartesian product is used to create item tuples, and the process continues as for joined-relation basis sets.

Definition 5 Given the above definitions of basis sets, association rules are defined in their standard way. A rule has the form $X \rightarrow Y$ where X and Y are sets of items as defined in definition 2. The *support* of a rule is the probability $P(X \cup Y)$ based on the set of all joined relation basis sets C. The *confidence* of a rule is the conditional probability $P(Y|X)$. The set of all items in the rule is an item set $I = X \cup Y$.

2.1 Graph-based interpretation

So far we have discussed the joining from a relational perspective. The problem can be mapped to a graph problem in a straight forward way. Entity instances are represented as nodes and relationship instances as edges. A 2-entity joined relation corresponds to all combinations of two nodes joined by an edge. In a 4-entity joined relation in an undirected graph there are two possible sub-graph structures that can be represented as a linear path corresponding to equation (2), and a T-shaped subgraph, corresponding to equation (3). Figure 1. shows the alternatives in a graph-based representation. Cycles are removed at the join level because they cannot contribute to UNIC basis sets. Note that the number of item sets is not a monotone function of the number of entities joined. In each joining step an item set may either disappear if the joined attribute participates in no relationships, or it may result in multiple new item sets, see Figure 2. Due to cycle removal the total number of join steps is finite.

It is important to understand that any relational association rule depends on the context in which it was generated. A rule that involves only two related entities can, in principle, be found in a 2-entity joined relation and any higher order relation. The support and confidence will however vary depending on that context, and a rule that is strong in one context may not be so in another. We follow [8] in always using the lowest order possible.

Definition 6 An item set I is *out-of-scope* if one or more entities are not represented, i.e., if $|\pi_G(I)| < n$ where $||$ indicates the cardinality, π is the relational projection operation, G is the identifier attribute of the item tuples, and n is the number of entity relations that were joined.

Definition 7 An item set I is *repetitious* if at least one descriptor occurs more than once, i.e., if $|\pi_D(I)| < |I|$ where π_D is the projection on the descriptor attribute. Two items are considered *repetitious* if they belong to the same joined relation basis set, their origin identifier differs, and their descriptor is equal.

3 Related work

Relational association rule mining in general has been addressed in the context of inductive logic programming [10, 17]. These approaches are very flexible and leave most choices up to the user. It cannot, however, be expected that a user understands the implications of out-of-scope and repetitious rules sufficiently to attempt the kind of solution we are proposing in this paper. Oyama et al. [22] have applied association rule mining to joined relations of protein annotations. This work notes the problem of what we term repetitious rules but does not resolve it. The problem of rule interest has been addressed in a variety of work on redundant rules, including closed set generation [29, 9, 4]. Additional rule metrics such as support as lift and conviction have been defined [6]. These approaches do not address the problem of excluding classes of rules as well as their impact on support and confidence from the scope of the association rule mining algorithm.

A further related research area is graph-based ARM [16, 21, 27, 7]. The relations we are considering in this paper can be viewed as graphs. Graph-based ARM does not typically consider more than one label on each node or edge. The goal of graph-based ARM is to find frequent substructures in that setting. The focus is on algorithms that scale to large subgraphs. The major problem in relational ARM is scaling with respect to item set size. Scaling to large subgraphs is usually irrelevant due to the "small world" property of many types of networks. For the networks under consideration in this paper any protein can be reached from almost any other by means of no more than three interactions [3, 26]. Association rules that involve longer distances are therefore unlikely to produce meaningful results.

There are other areas of research on ARM in which related transactions are mined in some combined fashion. Sequential pattern or episode mining [2, 28, 23, 30] and intertransaction mining [25] are two main categories. Some similarities in the formalism can be observed since we are also interested in mining across what can be considered transactions. A tuple in a joined relation can ultimately be compared with sequences of transactions. Over all the goals of these approaches are too different to be applicable to our setting in any direct way.

4 Comparison with a probabilistic approach

In the introduction we motivated UNIC as a way of eliminating not only repetitious and out-of-scope rules themselves but also their contributions to support and confidence of all rules. This approach differs from other work on redundancy in association rule mining [4, 29] in which some rules are discarded and all others left unchanged.

We compare UNIC with a probabilistic approach to correlation elimination in the presence of rules that are explicitly considered irrelevant. Using the probabilistic interpretation of confidence and support in definition 5. we can calculate the joint probability of an out-of-scope and a repetitious rule under the assumption of independence. Our discussion will focus on rules of the type $\{\mathit{0.A}\} \rightarrow \{\mathit{1.B}\}$ in the 2-entity setting, which are the simplest rules to pass all trivial relevance tests. Generalizing the concept to more complex rules is straight forward.

The confidence of a rule can be written as conditional probability$\mathrm{Conf}(\{0.A\} \rightarrow \{1.B\}) = \mathrm{P}(1.B|0.A)$. Repetitious rules lead to high conditional probabilities $\mathrm{P}(1.A|0.A)$ and $\mathrm{P}(1.B|0.B)$. An out-of-scope rule $\{A\} \rightarrow \{B\}$ is defined within an entity and does thereby not depend on the entity identifier ($\mathrm{P}(0.B|0.A) = \mathrm{P}(1.B|1.A)$). Here we assume an undirected setting for the interactions. In the directed case this observation does not hold. In a probabilistic approach we make an assumption of independence for the out-of-scope and the repetitious group and subtract their contribution.

$$\begin{aligned} &\mathrm{P}_{\mathrm{irrel}}(1.B|0.A) = \mathrm{P}(1.B|1.A)\ \mathrm{P}(1.A|0.A) \\ &+\mathrm{P}(1.B|0.B)\ \mathrm{P}(0.B|0.A) \\ &-\mathrm{P}(1.B|1.A, 0.B)\ \mathrm{P}(1.A, 0.B|0.A) \end{aligned} \tag{7}$$

The joint probability, which corresponds to rule support is

$$\begin{aligned} &\mathrm{P}_{\mathrm{irrel}}(0.A, 1.B) = \frac{\mathrm{P}(0.A, 1.A)\mathrm{P}(1.A, 1.B)}{\mathrm{P}(1.A)} \\ &+\frac{\mathrm{P}(0.A, 0.B)\mathrm{P}(0.B, 1.B)}{\mathrm{P}(0.B)} \\ &-\frac{\mathrm{P}(1.A, 0.B, 1.B)\mathrm{P}(0.A, 1.A, 0.B)}{\mathrm{P}(1.A, 0.B)} \end{aligned} \tag{8}$$

We will now compare this model with the results of the UNIC algorithm. Support of the rule $\{0.A\} \rightarrow \{1.B\}$ can be written as the following probabilities

$$\mathrm{Support}_{\mathrm{UNIC}}(0.A \rightarrow 1.B)$$

$$
\begin{aligned}
&= P(0.A, \neg 1.A, \neg 0.B, 1.B)\\
&= P(0.A, 1.B) - P(0.A, 1.A, 1.B)\\
&-P(0.A, 0.B, 1.B) + P(0.A, 1.A, 0.B, 1.B) \quad (9)
\end{aligned}
$$

One central approximation has to be made to identify parts of this expression with equation (8).

$$
\begin{aligned}
\mathrm{P}(0.A, 1.B|1.A) &= \mathrm{P}(0.A|1.A)\mathrm{P}(1.B|1.A)\\
\mathrm{P}(0.A, 1.A, 1.B) &= \frac{\mathrm{P}(0.A, 1.A)\mathrm{P}(1.A, 1.B)}{1.A} \quad (10)
\end{aligned}
$$

By following equation (10) we replace pieces in equation (9) that are of the shape P(0.A, 1.A, 1.B). This allows equation (9) to be written equivalently to equation (8). It is assumed that if the properly $1.A$ ($0.B$ and $\{1.A, 0.B\}$ respectively) exists, - and only in that case - the properties $0.A$ and $1.B$ are independent. In the absence of the property $1.A$ we do not expect these conditional probabilities to be independent.

We have now shown that as far as support is concerned the standard ARM model with elimination of the joint probability of independent out-of-scope and repetitious contributions matches the UNIC algorithm to the extent that could be expected. When comparing confidence we do, however, have to make an additional assumption.

$$
\begin{aligned}
&\mathrm{Conf}_{\mathrm{UNIC}}(0.A \rightarrow 1.B)\\
&= \frac{P(0.A, \neg 1.A, \neg 0.B, 1.B)}{\mathrm{P}(0.A, \neg 1.A)}\\
&= \frac{\mathrm{Support}_{\mathrm{UNIC}}}{\mathrm{P}(0.A) - \mathrm{P}(0.A, 1.A)} \quad (11)
\end{aligned}
$$

If the denominator was $\mathrm{P}(0.A)$ we could rewrite confidence based on equation (10) to match the standard ARM model. Calculating $\mathrm{P}(0.A)$ would, however have some serious drawbacks from the implementation point of view. A separate relation from which no elements are deleted would have to be constructed, many support calculations duplicated, and no standard ARM algorithm could be used. Maybe more importantly, whereas the UNIC algorithm returns a support and confidence that is exact, based on the modified relations, the above result is an approximation to the probabilistic model. Our main observation is that there is very little difference between both models, giving additional justification to the UNIC algorithm.

5 Implementation & results

We implemented the UNIC algorithm in modular fashion. Three major parts are distinguished. Preprocessing (1.-3.) includes application of the UNIC operator (see definition 3 in section 2 and UNIC_OP below). The actual item set generation step (4.) is done based on sets of items that appear as regular sets to the ARM program. Postprocessing (5.,6.) does additional filtering at the item set and rule level. This modular design allows a user to choose an ARM algorithm to satisfy any additional requirements he may have. In our work we tried different implementations [14, 5] as the core. Results in this paper use the Apriori algorithm from Christian Borgelt [5].

Table 1. Frequent Items

Item	Abbrev.	Support
METABOLISM	META	25.37%
lethal	letha	20.89%
nucleus	nucl	20.34%
TRANSCRIPTION	TRANS	17.32%
CELL_CYCLE_AND_DNA_PROCESSING	CCaDP	14.67%
cytoplasm	cyto	14.27%
Conditional_phenotypes	Cd	12.79%

Preprocessing includes the following tasks. For undirected graphs only one direction is typically included in data sets. We create both directions to ensure correct representation and then join the relations. Cycles are detected and removed in the process. For each transaction the UNIC operator equation (6) is applied in step (8.). If the UNIC operator has removed all items related to any one of the entities the basis set is marked as deleted. Such basis sets can never contribute to in-scope item sets or rules. The basis set is therefore not passed to the ARM method. We do, however, calculate support and confidence based on the full set of joined table basis sets (9.-13.). Once the basis sets are processed into the UNIC item basis sets, standard Apriori is applied (4.). For undirected graphs symmetric versions of each item set are returned and have to be removed. Frequent item sets or closed item sets are returned as result as usual. Item sets are tested if all entities are represented (15.). If not, the item set is removed as being out-of-scope. Rules are then produced as in standard ARM by processing the frequent itemsets (16.). The algorithm concludes with a set of rules that satisfy the requirements from section 2.

UNIC-ARM Algorithm:
Number of entities in the join relation: n
n-entity joined relation basis set: B_i
Set of basis sets C:$\{B_0$,...,$B_m\}$

```
UNIC($n$,$minconf$,$minsup$,$C$)
1. For undirected graphs represent each direction
2. Join relations and eliminate cycles
3. $C^U$=UNIC_OP($n$,$C$)
4. FreqSets=Apriori:FreqItemset_Gen($C'$,$minsup$)
5. For undirected graphs remove symmetric contributions
6. UNIC_SCOPERULE($FreqSet$,$n$,$minconf$)
```

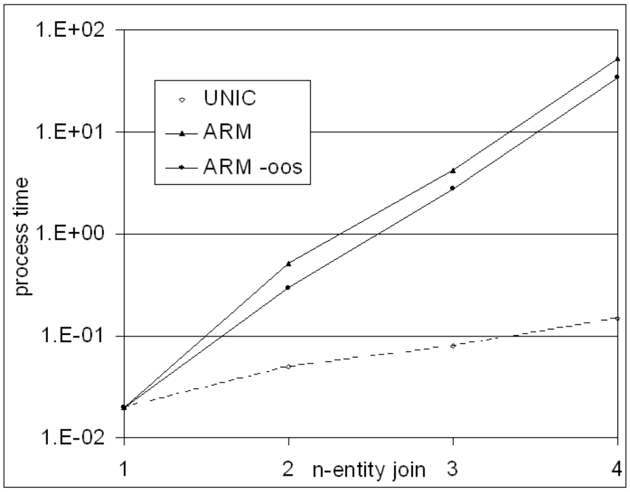


Figure 3. Reduction in Processing

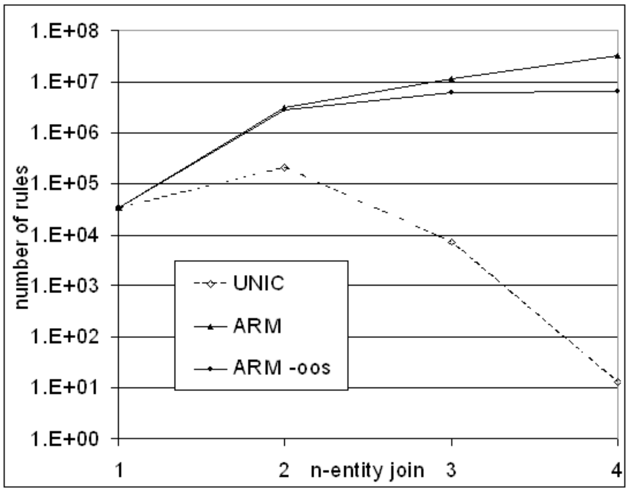


Figure 4. Reduction in Number of Rules

```
UNIC_OP($n$,$C$)
foreach transaction, $B_i \in C$
    $B_i^U = U(B_i(\{0.I_i, ..., (n-1).I_i\}))$
    foreach $j.I_i^U \in B_i^U$
        if($j.I_i^U == \emptyset$)
            mark tuple as deleted
    $C^U + = B^U$
Return→ $C'$

UNIC_SCOPERULE($FreqSet$, $n$, $minconf$)
foreach $J_i \in FreqSet$
    if($|\pi_G(J_i)| == n$ )
        Apriori:Rule_Gen($J_i$,$minconf$)
```

5.1 Data sets

YEASTP-P consists of two data tables that were gathered from the Comprehensive Yeast Genome Database at MIPS [19, 20]. We use yeast protein annotation data as the entity relation. Annotations are hierarchically structured with hierarchies for function, localization, protein class, complexes, phenotypes, and motifs. We pick the highest level in each hierarchy as items. In all categories attributes are multi-valued. ORF ids serve as tuple identifiers. Physical protein-protein interactions, which are undirected, are used to define the relationship.

We evaluated our algorithm on three UNIC relations derived from YEASTP-P, based on the following joined relations of equation (1): for R_{2e}, R_{3e}, and R_{4el}, where the 4-entity relation corresponds to the linear option equation (2). We used the 1-entity relation for comparison purposes. The 1-entity relation has 6,480 gene ORF entries, 2,282 of which have no attributes associated with them. Most entity instances have few interactions while some "hubs" have as many as 29. Of the 4,198 tuples in the entity relation, a little less than 50% (1,868) had any interactions. The average number of interactions per entity is 0.7 with a standard deviation of 1.9. 177 entities, on average, share an annotation, corresponding to an average of 288 interactions for each annotation. Each relation was mined with the standard ARM implementation (ARM) on the joined relation, standard ARM on the joined relation that removed out-of-scope rules in post-processing (ARM -oos), and the UNIC ARM implementation (UNIC). In each case the support and confidence were fixed at 3% and 20% respectively.

5.2 Performance and complexity

Three contributions to the complexity have to be distinguished: UNIC_OP, Apriori and UNIC_SCOPERULES. The most important contribution is the Apriori algorithm. Since we did not modify the algorithm itself, changes in performance come from data reduction. The resulting improvement is highly significant. Figure 3 shows the processing time of the Apriori algorithm. Recorded is the time to generate frequent itemsets. We did not include time to load the database or print the rules. As seen, UNIC outperforms ARM by a factor of 100 in the 4-entity setting. Other contributions to the processing time can be neglected since they are linear in the number of input basis sets and output item sets respectively.

A main goal in using the UNIC algorithm was elimination of rules that would not be considered interesting. We will now look at how the volume of rules has been changed by using UNIC. A comparison between standard ARM and UNIC is shown in Figure 4. It can be seen that the number of rules produced for a given support and confidence

Table 2. Standard ARM Rules

RHS		LHS	Supp	Conf
0.nucl	←	1.nucl	29.45%	68.31%
1.nucl	←	0.nucl	29.45%	68.31%
0.nucl	←	0.TRANS	27.00%	81.95%
1.nucl	←	1.TRANS	27.00%	81.95%
0.TRANS	←	0.nucl	27.00%	62.62%
1.TRANS	←	1.nucl	27.00%	62.62%
1.TRANS	←	0.TRANS	22.82%	69.28%
0.TRANS	←	1.TRANS	22.82%	69.28%
1.nucl	←	0.TRANS	22.29%	67.67%
0.nucl	←	1.TRANS	22.29%	67.67%

Table 3. Modified ARM Rules

RHS		LHS	Supp	Conf
0.nucl	←	1.nucl	29.45%	68.31%
0.TRANS	←	1.TRANS	22.82%	69.28%
0.nucl	←	1.TRANS	22.29%	67.67%
0.TRANS	←	1.nucl	22.29%	51.70%
0.letha	←	1.letha	20.39%	51.60%
0.nucl	←	0.TRANS 1.nucl	19.43%	87.17%
0.nucl	←	1.TRANS 1.nucl	19.43%	71.98%
0.TRANS	←	1.nucl 0.nucl	19.43%	65.98%
0.TRANS	←	1.TRANS 0.nucl	19.41%	87.06%
0.nucl	←	0.TRANS 1.TRANS	19.41%	85.03%

Table 4. UNIC ARM Rules

RHS		LHS	Supp	Conf
0.letha	←	1.Cp	4.33%	28.98%
0.Cp	←	1.letha	4.33%	27.14%
0.letha	←	1.META	2.63%	32.70%
0.letha	←	1.cyto	2.53%	29.33%
0.nucl	←	1.cyto	2.33%	26.98%
0.Cp	←	1.cyto	2.25%	26.10%
0.CCaDP	←	1.TRANS	2.23%	23.91%
0.TRANS	←	1.CCaDP	2.23%	20.80%
0.nucl	←	1.META	2.13%	26.42%
0.CCaDP	←	1.META	1.97%	24.53%

is reduced by a factor of up to 10^6. It becomes obvious that no analyst would be able to the large data volume of standard ARM by visual inspection. Even if we limit the output of standard relational ARM to the correct scope as (ARM -oos), the number of rules is still larger by a factor of 10^3 compared with UNIC. Note that for a 1-entity table the results of all algorithm variants are the same: In the non-relational setting the UNIC results reduce to those of standard ARM.

In a preliminary study of our data we found strong item sets within the 1-entity setting, in particular the item set {*TRANS, nucl*}. When using standard ARM on joined tables it can be seen that the same and related item sets dominate the results in various combinations, c.f. table 2. Note in particular that table 2 does include a rule of the form {*0.nucl*}←{*1.TRANS*}. Based on the standard ARM result we would conclude that this may be an interesting rule. Comparison with the UNIC algorithm does not, however, show this rule as strong. The high support in the standard ARM setting is a consequence of several other rules that are listed in the same table 2, namely {*0.nucl*}←{*0.TRANS*} and {*0.nucl*}←{*1.nucl*}.

Table 3 shows the effect of eliminating out-of-scope rules as well as symmetric copies of the same rule. Although the results look more interesting now, we know already that rules involving {*TRANS*} and {*nucl*} on both sides of the rule are likely to be redundant. The rule filtering has aggravated the problem because the user no longer sees out-of-scope rules, and may therefore not be aware of the danger of misinterpreting results.

The UNIC rule set (table 4) finally provides us with the information for which we were looking. The top rule in standard ARM that is not out-of-scope or repetitious {*0.nucl*}← {*1.TRANS*} ranks very low in UNIC at support of 0.56% and confidence of 5.98%. This confirms our suspicion that most incidents of {*TRANS*} occurring on one side of an interaction and {*nucl*} on the other are due to at least one of the two annotations occurring on both sides as well as the rule of the 1-entity relation{*nucl*}← {*TRANS*}. Support and confidence reported in UNIC reflect the non-trivial influence of the rule, which in this case is not strong enough to be listed among the top 10 rules. Experience showed that UNIC rules give insight into boundaries between network regions with homogeneous annotations. Examples of boundaries could be collaborations involving multiple teams in an organization, or, in the protein interaction network setting, compartmental cross-talk between different parts of the cell. This analysis is possible because UNIC specifically emphasis differences between the participating entities of a relationship.

6 Conclusions

We have shown in this paper that relational data has properties that make a straight forward generalization of standard data mining algorithms problematic. We identify a major source of misleading rules in association rule mining that are due to typical correlation between the same items associated with different entity instances. Rather than leaving the responsibility of solving such problems to the user we propose an association rule mining algorithm that consistently eliminates the impact of items that are shared by multiple entity instances. Our approach ensures quality of the output as well as speed through pruning that is done even before the application of the association rule mining step. We demonstrate the success of our technique for a biological data set.

7 Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. #01322899.

References

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

[2] R. Agrawal and R. Srikant. Mining sequential patterns. In *Eleventh International Conference on Data Engineering*, pages 3–14, Taipei, Taiwan, 1995. IEEE Computer Society Press.

[3] A. L. Barabasi and E. Bonabeau. Scale-free networks. *Scientific American*, 288(5):60, 2003.

[4] Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, and L. Lakhal. Mining minimal non-redundant association rules using frequent closed itemsets. *Lecture Notes in Computer Science*, 1861:972, 2000.

[5] C. Borgelt. Apriori. *http://fuzzy.cs.uni-magdeburg.de/˜borgelt/software.html*, August 2003.

[6] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proceedings ACM SIGMOD International Conference on Management of Data*, Tucson, AZ, 1997.

[7] D. J. Cook and L. B. Holder. Graph-based data mining. *IEEE Intelligent Systems*, 15(2):32–41, 2000.

[8] L. Cristofor and D. Simovici. Mining association rules in entity-relationship modeled databases. Technical report, University of Massachusetts Boston, 2001.

[9] L. Cristofor and D. Simovici. Generating an informative cover for association rules. In *Proceedings of International Conference on Data Mining*, Maebashi, Japan, 2002.

[10] L. Dehaspe and L. D. Raedt. Mining association rules in multiple relations. In *Proceedings of the 7th International Workshop on Inductive Logic Programming*, volume 1297, pages 125–132, Prague, Czech Republic, 1997.

[11] S. Džeroski and N. Lavrač, editors. *Relational Data Mining*. Springer, Berlin, 2001.

[12] S. Džeroski and N. Lavrač. *Relational Data Mining*, chapter 3. An Introduction to Inductive Logic Programming, pages 48–73. Springer, Berlin, 2001.

[13] Elmasri and Navathe. *Fundamentals of Database Systems*. Pearson, Boston, 4th edition, 2004.

[14] G. Grahne and J. Zhu. Efficiently using prefix-trees in mining frequent itemsets. In *FIMI'03 Workshop on Frequent Itemset Mining Implementations*, November 2003.

[15] J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. In *Proceedings of the 21th International Conference on Very Large Data Bases*, San Francisco, CA, 1995.

[16] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 13–23, Lyon, France, 2000.

[17] A. J. Knobbe, H. Blockeel, A. Siebes, and D. M. G. van der Wallen. Multi-relational data mining. Technical Report INS-R9908, Maastricht University, 9, 1999.

[18] S. A. Macskassy and F. Provost. A simple relational classifier. In *2nd Workshop on Multi-Relational Data Mining at KDD'03*, Washington, D.C., 2003.

[19] H. Mewes, D. Frishman, U. Gldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Mnsterkoetter, S. Rudd, and B. Weil. Mips: a database for genomes and protein sequences. *Nucleic Acids Research*, 30(1):31–44, 2002.

[20] H. W. Mewes, D. Frishman, U. Gldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Mnsterkoetter, S. Rudd, and B. Weil. Cygd. *http://mips.gsf.de/genre/proj/yeast/index.jsp*, January 2003.

[21] K. Michihiro and G. Karypis. Frequent subgraph discovery. In *Proceedings of the International Conference on Data Mining*, pages 313–320, San Jose, California, 2001.

[22] T. Oyama, K. Kitano, K. Satou, and T. Ito. Extraction of knowledge on protein-protein interaction by association rule discovery. *Bioinformatics*, 18(8):705–14, 2002.

[23] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. PrefixSpan mining sequential patterns efficiently by prefix projected pattern growth. In *Proceedings of the 17th International Conference on Data Engineering*, pages 215–226, Heidelberg, Germany, 2001.

[24] A. H. Y. Tong, M. Evangelista, A. B. Parsons, H. Xu, G. D. Bader, N. Pag, M. Robinson, S. Raghibizadeh, C. W. V. Hogue, H. Bussey, B. Andrews, M. Tyers, and C. Boone. Global mapping of the yeast genetic interaction network. *Science*, 303(5695):808–815, 2004.

[25] A. K. H. Tung, H. Lu, J. Han, and L. Feng. Breaking the barrier of transactions: Mining inter-transaction association rules. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, 1999.

[26] A. Wagner and D. A. Fell. The small world inside large metabolic networks. *Proceedings of the Royal Society: Biological Sciences*, 9:1803–10, 2001.

[27] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *Proceedings of the International Conference on Data Mining*, Maebashi City, Japan, 2002.

[28] X. Yan, J. Han, and R. Afshar. Clospan: Mining closed sequential patterns in large datasets. In *Proceedings 2003 SIAM Int.Conf. on Data Mining*, San Francisco, California, 2003.

[29] M. J. Zaki. Generating non-redundant association rules. In *Knowledge Discovery and Data Mining*, pages 34–43, Boston, MA, 2000.

[30] M. J. Zaki. SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning Journal*, 42:31–60, 2001.