On the Discovery of Mutually Exclusive Items in a Market Basket Database

George Tzanis, Christos Berberidis and loannis Vlahavas

Machine Learning and Knowledge Discovery Group
Department of Informatics, Aristotle University of Thessaloniki
Thessaloniki 54124, Greece
{gtzanis, berber, vlahavas}@csd.auth.gr - http://mlkd.csd.auth.gr

Abstract. Mining a transaction database for association rules is a particularly popular data mining task, which involves the search for frequent co-occurrences among items. One of the problems often encountered is the large number of weak rules extracted. Item taxonomies, when available, can be used to reduce them to a more usable volume. In this paper we introduce a new data mining paradigm, which involves the discovery of pairs of mutually exclusive items. We call this new type of knowledge mutual exclusion, as opposed to association, and we propose its use to tackle the aforementioned problem. We formulate the problem of mining for mutually exclusive items, provide important background information, propose a novel mutual exclusion metric and finally, present a mining algorithm that we test on transaction data.

1 Introduction

Association rules mining has attracted the attention of the data mining research community since the early 90s, as a means of unsupervised, exploratory analysis, originally of market basket data. An association rule implies the co-existence of a number of items in a portion of a transaction database. The goal of the exploratory mining process is to provide the decision maker with valuable unsupervised knowledge about a certain domain modeled by a transaction database. The frequent existence of two or more items in the same transaction implies a relationship among them. For example the existence of bread and peanut butter in the same basket implies a possible buying behavior pattern that can be further investigated in order to improve the sales of both products. Intuitively, the rare or the absolutely non co-existence of two products could also imply a relationship, a mutual exclusion property between them.

During the last decade, a group of algorithms and approaches that embed a special kind of item information, called *taxonomy* has emerged. A taxonomy is basically an "*is-a*" concept hierarchy and whenever such information is available for a database, it can be used to summarize a large number of association rules, reducing thus the produced knowledge to a more comprehensible and usable volume. This kind of association rule is referred to as *multiple-level* or *generalized* or *hierarchical* association rule. Several algorithms and approaches have been proposed so far and in all of them taxonomy information is always available or known in advance. However, this is not always the case and sometimes it would be useful if we had a valuable hint regarding the existence of such information.

In this paper we present the problem of mining a transaction database for mutually exclusive items. Our goal is to assist the mining procedure with knowledge that can eventually be utilized to mine for stronger association rules. We present a simple method for collecting information in order to summarize frequent itemsets, when no taxonomy information is previously known. Moreover, we define the problem of mining for mutually exclusive items. The intuition behind this idea is that two or more items may be mutually exclusive (they never or rarely exist together in the same itemset) because they are very similar (for example, rarely someone buys two different brands of cola) and therefore they may be taxonomically related. We claim that this can be a serious lead for the expert to look into, in order to create a taxonomy based on consumer preferences for his application domain and eventually mine for stronger association rules.

The paper is organized as follows. The next section presents the required background knowledge. Section 3 contains a short review of the relative literature. Section 4 contains the description of the proposed approach, definitions of terms and notions used, the proposed algorithm, some new metrics for evaluating the discovered mutually exclusive items and an illustrative example of our approach along with a discussion about the intuition of our approach. Finally, in section 6 we summarize with the conclusions we drew from this research and our ideas for future work on this topic.

2 Preliminaries

The association rules mining paradigm involves searching for co-occurrences of items in transaction databases. Such a co-occurrence may imply a relationship among the items it associates. These relationships can be further analyzed and may reveal temporal or causal relationships, behaviors etc.

The formal statement of the problem of mining association rules can be found in [3]. Given a finite multiset of transactions D, the problem of mining association rules is to generate all association rules that have support and confidence at least equal to the user-specified minimum support threshold (*min_sup*) and minimum confidence threshold (*min_conf*) respectively.

The problem of discovering all the association rules can be decomposed into two subproblems [1]:

1. The discovery of all itemsets that have support at least equal to the user-specified minimum support threshold. These itemsets are called *large* or *frequent* itemsets.
2. The generation of all rules from the discovered frequent itemsets. For every frequent itemset *F*, all non-empty subsets of *F* are found. For every such subset *S*, a rule of the form $S \Rightarrow F\text{-}S$ is generated, if the confidence of the rule is at least equal to the minimum user-specified confidence threshold.

Another method to extract strong rules is the use of *concept hierarchies*, also called *taxonomies* that exist in various application domains, such as market basket analysis.

A taxonomy is a concept tree, where the edges represent "is-a" relationships from the child to the parent. Example of such a relationship is: Cheddar *is-a* cheese *is-a* dairy product *is-a* food *is-a* product. When a taxonomy about a domain of application is available, a number of usually high-confidence rules that are too specific (having low support) can be merged, creating a rule that aggregates the support and therefore the information, in a higher abstraction level, of the individual rules. In other words, "looser" associations at the lower levels of the taxonomy are summarized, producing "winner" associations of higher levels. For example, the rule "*if a customer buys 0.5 lb. wheat bread then he also buys 1 lb. skimmed milk*" is very likely to have low support, while a rule "*if a customer buys bread then he also buys milk*" is very possible that it has much higher support, because it includes all types, brands and packages of bread and milk bought by the customers of the store.

3 Related Work

Association rules were first introduced in 1993 by Agrawal et al. [1] as a market basket analysis tool. In 1994 Agrawal and Srikant [2] proposed Apriori, a level-wise algorithm, that works by generating candidate itemsets and testing if they are frequent by scanning the database. Apriori exploits the downward closure property, according to which any non-empty subset of a frequent itemset is also frequent. Therefore, at each level the candidate frequent itemsets are generated based only on the frequent itemsets found in the previous level. About the same time Manilla et al. [11] discovered independently the same property and proposed a variation of Apriori, the OCD algorithm. A joint paper combining the previous two works was later published [3]. Several algorithms have been proposed since then, others improving the efficiency, such as FPGrowth [9], and others addressing different problems from various application domains, such as spatial [10], temporal [6] and intertransactional rules [16], which can be also used for prediction [4].

One of the major problems in association rules mining is the large number of often uninteresting rules extracted. Srikant and Agrawal [13] presented the problem of mining for generalized association rules. These rules utilize item taxonomies in order to discover more interesting rules. For example, given that <Gouda is-a Cheese> and <Cheddar is-a Cheese>, we discover a rule such as "If a customer buys Bread then he will also buy Cheese", with support higher than of a rule for a specific cheese. The authors propose a basic algorithm as well as three more efficient algorithms, along with a new interestingness measure for rules, which uses information in the taxonomy. Thomas and Sarawagi [15] propose a technique for mining generalized association rules based on SQL queries. Han and Fu [8] also describe the problem of mining "multiple-level" association rules, based on taxonomies and propose a set of top-down progressive deepening algorithms. Teng in [14] proposes a type of augmented association rules, using negative information called dissociations. A dissociation is relationship of the form "X does not imply Y", but it could be that "when X appears together with Z, this implies Y". Han and Fu [7] propose some algorithms for dynamic refinement and dynamic generation of concept hierarchies. The generation of concept hierarchies concerns only numerical attributes and is based on data distribution. The dynamic refinement of a given or even a generated concept hierarchy is based on a particular learning request, the relevant set of data and database statistics.

Another kind of association rules are negative association rules. Savasere et al. [12] introduced the problem of mining for negative associations. Negative associations deal with the problem of finding rules that imply what items are not likely to be purchased when a certain set of items is purchased. The approach of Savasere et al. demands the existence of a taxonomy and is based on the assumption that items belonging to the same parent of taxonomy are expected to have similar types of associations with other items. Since they consider only those cases where the expected support can be calculated based on the taxonomy, only a subset of the whole set of negative association rules can be discovered. They propose a naive and an improved algorithm for mining negative association rules along with a new measure of interestingness. In a more recent work Wu et al. [17] present an efficient method for mining positive and negative associations and propose a pruning strategy and an interestingness measure. Their method extends the traditional positive association rules ($A \Rightarrow B$) to include negative association rules of the form $A \Rightarrow -B$, $-A \Rightarrow B$, and $-A \Rightarrow -B$ (the minus sign means not). The last three rules indicate negative associations between itemsets *A* and *B*. A mutual exclusion can not be expressed by one such rule. If items *A* and *B* are mutually exclusive, then $A \Rightarrow -B$ and $B \Rightarrow -A$ concurrently, that is different from $-A \Rightarrow -B$.

4 Mining for Mutually Exclusive Items

Our approach for the discovery of mutually exclusive items has two phases. In the first phase, we mine for contiguous frequent itemsets, producing the extensions that will be used in the next phase as candidate mutually exclusive items.

4.1 Mining for Contiguous Frequent Itemsets

In the following lines we provide some definitions and formulate the problem of mining contiguous frequent itemsets as defined in our previous work [5]. Let *I* be a finite set of items and *D* be a finite multiset of transactions. Each transaction $T \in D$ is a set of items such that $T \subseteq I$. Mining for frequent *k*-itemsets involves searching in a search space, which consists of all the possible combinations of length *k* of all items in *I*. Every frequent itemset $F \subseteq I$ divides the search space in two disjoint subspaces: the first consists of the transactions that contain *F* and from now on will be called the *F*-subspace and the second all the other transactions.

Now, let $F \subseteq I$ be a frequent itemset in *D*, according to a first-level support threshold and $E \subseteq I$ be another itemset. The itemset $F \cup E$ is considered to be a contiguous frequent itemset, if $F \cap E = \varnothing$ and *E* is frequent in the *F*-subspace, according to a second level support threshold. Itemset *E* is called the locally frequent extension of *F*. The term locally is used, because *E* may not be frequent in the whole set of transactions. In order to avoid any confusion, from now on we will use the terms local and locally, when we refer to a subset of *D* and the terms global and globally when we refer to *D*. For example, we call global support (*gsup*) the first-level support and local support (*lsup*) the second-level support. An itemset *F* that satisfies the minimum global support threshold (*min_gsup*) is considered to be globally frequent and an itemset E that is frequent in the F-subspace, according to the minimum local support threshold (*min_lsup*), is considered to be locally frequent. The global support of an itemset can be calculated as in equation (1) and the local support of an itemset *E* in the *F*-subspace can be calculated as follows:

$$\mathrm{lsup}(E,F) = \frac{\mathrm{gsup}(E \cup F)}{\mathrm{gsup}(F)} \qquad \mathbf{(1)}$$

The local support threshold can be set arbitrarily by the user-expert or can be the same as the global support threshold. The contiguous frequent itemsets that contain a locally frequent extension of length *k* are called *k-contiguous frequent itemsets*.

Given a finite multiset of transactions *D*, the problem of mining contiguous frequent itemsets is to generate all itemsets $F \cup E$ that consist of an itemset *F* that has global support at least equal to the user-specified minimum global support threshold and an extension *E* that has local support at least equal to the user-specified minimum local support threshold.

We apply a level-wise technique in order to extract the contiguous frequent itemsets. The intuition behind this approach is twofold: First, if these extensions are frequent indeed in the subspace of a frequent itemset then they could be important information about these itemsets, lost by a number of reasons. Second, if a large number of itemsets share the same extensions and these common extensions are frequent in the subspace of these itemsets, they are likely to be of the same category and the same level of taxonomy. In such cases, the total support of the parent node in the taxonomy is broken down to many lower level supports, which are not high enough to satisfy the minimum set threshold and which explains the possible loss of potentially valuable knowledge. The support of the current itemset is reduced because of the low support of the extensions and eventually fails to qualify as a frequent itemset. When no taxonomy information is available in advance, the information gathered from this process can be a valuable hint about the taxonomy effect explained here and eventually the existence of a taxonomy. Mining a transaction database to discover the underlying taxonomy is an elusive task with questionable results. In the existing applications, the existence of a taxonomy assists the miner to discover valuable knowledge. In our setup, the Holy Grail would be exactly the inverse procedure. In a mining problem where no taxonomy information is provided in advance, is it possible to follow a kind of reverse engineering procedure in order to mine nuggets of taxonomy information from the data? If so, then the benefit would be twofold. First, the domain expert would be provided with pieces of potentially valuable knowledge about concept hierarchies among items in the database. Second, this knowledge could be used again to mine for multiple-level or generalized association rules, especially those hierarchical relationships that the expert verifies or wants to test for their validity.

4.2 The Mutually Exclusive Items Mining Algorithm

Based on the notions explained so far, we propose an algorithm for mining for mutually exclusive items in a transaction database and we focus on its application on market basket data. The *mutual exclusion* of two database items could imply an alternative type of association. Of course, this kind of mined knowledge needs to be confirmed by the domain expert so that it can be utilized eventually.

Mining for mutually exclusive items in a database possibly containing several thousands of different items, involves searching in a space that consists of all the possible pairs of items, because virtually any of them could contain two items that exclude each other. However we consider this approach naive and simplistic. We propose a more intuitive approach, which is based on the assumption/observation that every frequent itemset expresses a certain behavior of a group of customers and therefore it could be used to guide our search. Items that appear with high frequency in the subspace of a frequent itemset are more likely to be systematically mutually exclusive or taxonomically related, because they follow a customer behavioral pattern and not because of pure chance or unusual cases. However, we cover these cases with an extra pass over the database in order to remove the candidate mutually exclusive pairs that for any reason are not confirmed by the transactions. Our algorithm is level-wise, like Apriori, and has three steps:

Step 1: Find all frequent itemsets according to a global support *s*, using any frequent itemset mining algorithm.

Step 2: For each frequent itemset discovered, mine for extensions.

Step 3: Mine for mutually exclusive items in the set defined by the union of the frequent itemsets and the extensions.

Step 2 requires a number of scans over the database, which is proportional to the size of the extensions discovered. In the case of the basic Apriori algorithm the number of scans is equal to the size of the itemsets, but there are some improvements in later versions that require less scans and they are more efficient. Moreover, algorithm in step 2 generates the *k*-sized frequent itemsets in a level-wise manner and thus can be stopped at any time, producing the frequent itemsets so far discovered. This can be used when the user is only interested in the small extensions (of size 1 or 2) or in extreme cases when the algorithm takes too much time to terminate and an output is required quickly.

4.3 Mutual Exclusion Evaluation Metrics

In order to distinguish when two items are mutually exclusive, we need a measure to evaluate the degree of the mutual exclusion between them. Initially, we should be able to evaluate this within the subspace of a frequent itemset (locally) and then it should be evaluated globally, with all the frequent itemsets that support this candidate pair to contribute accordingly. For this purpose, we propose the use of a metric we call MEM (Mutual Exclusion Metric) that can be calculated in two phases, the first one is local and is required for the second one, which is the global one.

Local metric. We propose the following local metric, which will be called Local MEM for the evaluation of a candidate pair of mutually exclusive items that is supported by a frequent itemset I and its range is [0, 1].

$$M_I = \left[P(A-B)+P(B-A)\right]\min\left[P(A-B\,|\,A),P(B-A\,|\,B)\right]= \tag{2}$$
$$\left[(S_A - S_{AB})+(S_B - S_{AB})\right]\min\left[\frac{(S_A - S_{AB})}{S_A},\frac{(S_B - S_{AB})}{S_B}\right]=$$
$$(S_A + S_B - 2S_{AB})\left[1-\frac{S_{AB}}{\min(S_A,S_B)}\right]$$

For the above formula *P(I)* = 1. S_X is the fraction of transactions that contain *X* over the number of transactions that contain *I*. Figure 1 shows that the Local MEM increases linearly when the local support of A and B increase, as long as there is no overlapping (S_A, S_B<0.5 and S_{AB}=0), until it reaches its maximum value, which is 1.We assume that S_A=S_B for simplicity, without any loss of generality. When the overlapping starts (S_A, S_B>0.5), it drops fast down to zero, because even a small degree of overlapping is strong evidence against the possibility of two items to be mutually exclusive.

Global Metric. We propose the following global metric (5) for the evaluation of a candidate pair of mutually exclusive items that is supported by a set IS of frequent itemsets.

$$GM_1 = IIF\left(\sum_{I\in \mathrm{IS}} M_I S_I\right) \tag{3}$$

IIF stands for *Itemset Independence Factor* and is calculated as the ratio of the number of the distinct items contained in all itemsets that support a candidate pair over the total number of items contained in these itemsets. For example, the *IIF* of the itemsets {*a,b,c*} and {*a,d*} is 0.8, since there are 4 distinct items (*a,b,c* and *d*) over a total of 5 ones (*a,b,c, a* and *d*). The *IIF* is used in order to take into account the possible overlapping of two candidate mutually exclusive items. We do this, because the overlapping between the transactions that contain two different itemsets may mean overlapping between the transactions that contain the pair. Alternatively, one can use (6), which is a normalized version of (5). The range of values of GM_1 is [0, +∞), while the range of GM_2 is [0, 1].

$$GM_2 = IIF\frac{\sum_{I\in IS} M_I S_I}{\max_{IS\in FS}\left(\sum_{I\in IS} S_I\right)} \tag{4}$$

FS is the set that contains all the sets of frequent itemsets that support a candidate pair of mutually exclusive items. The difference between (5) and (6) is the denominator in (6). This term is used in order to normalize the metric, since the numerator is always less or equal to the denominator. In other words, for each candidate pair we calculate the sum of the supports of the frequent itemsets that support the pair. The greatest sum is used to normalize the metric. This normalization is useful to the user who wants to know a priori the maximum value of the metric in order to use the proper threshold. However, this is not desirable when the user wants to compare the results of different datasets or the results obtained by running the algorithm with different parameters. The algorithm for mining mutually exclusive pairs of items is presented in Table 1.

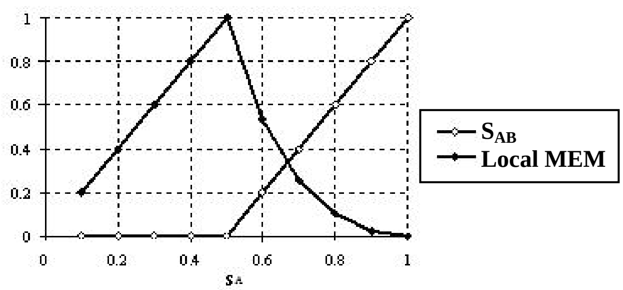

Fig. 1. The Local MEM against the local support of the candidate mutually exclusive items

Table 1. The mutuallty exclusive items mining algorithm.

Input: A multiset of transactions *D*, a set of frequent itemsets *FI*, a minimum local MEM threshold *min_lMEM* and a minimum global MEM threshold *min_gMEM*.
Output: All mutually exclusive pairs.

```
for each (T ∈ D)
   for each (I ∈ FI)
      if (I ⊆ T)
         for each (E ∈ T-I)
            Extensions(I) ← Extensions(I) ∪ E
            Count[I][E] ← Count[I][E] + 1
for each (T ∈ D)
   for each (I ∈ FI)
      for each (E1, E2 ∈ Extensions(I))
         if ({E1, E2} ⊆ T)
            ExtensionPairs(I) ← ExtensionPairs(I) ∪ {E1, E2}
            Count[I][{E1, E2}] ← Count[I][{E1, E2}] + 1
for each (I ∈ FI)
   for each (EP ∈ ExtensionPairs(I))
      if (Local_MEM(EP) ≥ min_lMEM)
         AllExtensionPairs ← AllExtensionPairs ∪ EP
for each (EP ∈ AllExtensionPairs)
   if (Global_MEM(EP) ≥ min_gMEM)
      MutuallyExclusivePairs ← MutuallyExclusivePairs ∪ EP
return MutuallyExclusivePairs
```

4.4 Example

The following tables show an illustrative example. Table 2 contains an example market basket dataset and Table 3 the discovered association rules. The minimum support was set to 2/9 in order to extract these rules.

After we applied our algorithm, we discovered the mutually exclusive pairs of items shown in Table 4, along with their metrics and corresponding frequent itemsets. The extracted pairs are all different types of coffee (espresso, cappuccino and decaffeinated). Replacing these items with a single item, we were able to increase the minimum support threshold in order to acquire stronger association rules (Table 5).

Table 2. An example market basket dataset

TID	Items in the Basket
1	espresso, sugar, newspaper
2	espresso, sugar, cola
3	espresso, sugar
4	cappuccino, cigarettes
5	cappuccino, sugar
6	cappuccino, sugar, sweets
7	decaf, sugar, chewing_gums
8	decaf, soda, vinegar
9	decaf, sugar, cigarettes

Table 3. Association rules mined from the dataset of Table 2

Association Rules	Support	Confidence
espresso ⇒ sugar	3/9	1
decaf ⇒ sugar	2/9	2/3
cappuccino ⇒ sugar	2/9	2/3

Table 4. Mutually exclusive items and their corresponding frequent itemsets mined from dataset of Table 2

Mutually Exclusive Pairs (Global MEM, Global Support)	Frequent Itemsets	
	Support	*Local MEM*
{espresso, cappuccino}: 5/9, 0	{sugar}: 7/9	0.714
{espresso, decaf}: 5/9, 0	{sugar}: 7/9	0.714
{cappuccino, decaf}: 4/9, 0	{sugar}: 7/9	0.571

Table 5. Generalized association rules, using the taxonomy discovered

Association Rules	Support	Confidence
sugar ⇒ coffee	7/9	1
coffee ⇒ sugar	7/9	7/9

4.5 Discussion

The knowledge that two items are mutually exclusive can be further analyzed in order to decide whether the two products could be included in the same level of taxonomy and under the same parent node. Searching for mutually exclusive pairs of items in a "blind" manner would produce a huge amount of candidate pairs. Moreover, most of the discovered mutually exclusive items would be uninteresting. The intuition behind searching for mutually exclusive items between the extensions of frequent itemsets is twofold. First, the search space is reduced sensibly, and second, a frequent itemset represents a class of consumers that have particular preferences. So, the mutually exclusive items found in the subspace of a frequent itemset are probably more interesting. Let, for example, consider a large store that sells cloths, shoes, sportswear, etc. Let also, consider itemset A = {Socks, AthleticShoes, Rackets}, that represents athletes who play tennis and itemset B = {Socks, AthleticShoes, Handball}, that represents athletes who play handball. Itemsets A and B may not have high support, while itemset $C = A \cap B$ = {Socks, AthleticShoes} is more likely to have higher support, since it represents the whole class of athletes. However, the two items, Rackets and Handball are expected to be locally frequent in the subset of transactions containing *C*. The use of traditional association rules does not provide the alternative to explore the *C*-subspace and consequently a large amount of potentially valuable knowledge will remain hidden. Conversely, with our algorithm Rackets and Handball would be discovered as mutually exclusive items indicating a taxonomical relationship.

The taxonomy generation is needed in many domains. Although taxonomies already exist, they may require refinement either because the consumers' preferences change by time, or because new products are produced, or even because some considerations made when the taxonomy was building were false. Moreover, preferences of consumers vary in different locations and consequently the taxonomies are different. For example, a taxonomy used in a supermarket in a highland place, is not appropriate for a supermarket located near the sea. Finally many different taxonomies can be built for a set of items depending on the various viewpoints. Returning to the example, maybe there is not any taxonomy containing Rackets and Handball at the same level and under the same parent node. Probably, a new taxonomy should be created or an old one should be refined.

5 Experiments

In order to evaluate the performance of our algorithm we conducted a number of experiments on an IBM-Artificial market basket dataset (T10I4D100K). This dataset contains 100000 transactions. The graphs in Figure 2 illustrate the performance of our mutually exclusive items algorithm in means of response time (seconds) while the minimum local support threshold varies from 0.1 to 0.03 and the minimum global support threshold varies from 0.01 to 0.04. As expected we observe that while the minimum local support threshold decreases, the response time of the algorithm increases. Moreover, observing the scale of the *y*-axis on the four graphs we can say that the response time depends more on the minimum global support threshold than the local one. This means that our algorithms performance decreases more rapidly with the number of frequent itemsets rather than with the number of frequent extensions of frequent itemsets.

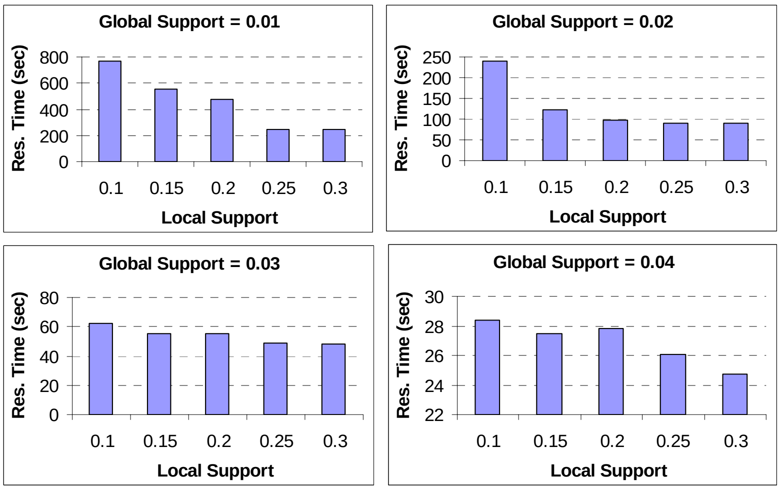

Fig. 2. Response time of the mutually exclusive items mining algorithm depending on local support for four fixed vales of global support

6 Conclusions

In this paper we have presented the novel problem of mining for mutually exclusive items, as an extension to the association rules mining paradigm. When two items are mutually exclusive, this can be used as a valuable hint when looking for previously unknown taxonomical relationships among them. In such case, this can be an interesting type of knowledge to the domain expert but more importantly, it can be used to mine the database for hierarchical or generalized association rules. For this purpose, we propose an intuitive approach, formulated the problem providing definitions of terms and evaluation metrics. We have also developed a mining algorithm, which extends Apriori, and we have applied it on a large synthetic dataset in order to test its performance. In the future, we would like to apply our approach on real world data sets in order to further investigate its performance.

Acknowledgements

This work was partially supported by the Greek R&D General Secretariat through a PABET-NE program (EPAN M.4.3.2, No. 04BEN51).

References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. Proceedings of the "ACM SIGMOD Conference on Management of Data", Washington, D.C., May 1993, pages 207-216.
2. Agrawal, R., and Srikant, R. Fast algorithms for mining association rules in large databases. In *Proceedings of the International Conference on Very Large Databases (VLDB'94),* (Santiago de Chile, Chile, September 12-15, 1994). 487-499.
3. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H. and Verkamo, A. I. Fast discovery of association rules. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (Editors), Advances in Knowledge Discovery and Data Mining. AAAI Press, Menlo Park, California 94025, USA, 1996. 307-328.
4. Berberidis, C., Angelis, L., and Vlahavas, I. PREVENT: An Algorithm for Mining Inter-transactional Patterns for the Prediction of Rare Events, In *Proceedings of the 2nd European Starting A1 Researcher Symposium.* (Valencia, Spain, 23-24 August 2004).128-136.
5. Berberidis, C., Tzanis, G., Vlahavas, I. Mining for Contiguous Frequent Itemsets in Transaction Databases, In *Proceedings of the IEEE 3rd International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, IEEE, Sofia, Bulgaria, 2005.
6. Chen, X., Petrounias, I. Discovering Temporal Association Rules: Algorithms, Language and System. In *Proceedings of the 16th ICDE.* (San Diego, California, 2000, 306.
7. Han, J., and Fu, Y. Dynamic Generation and Refinement of Concept Hierarchies for Knowledge Discovery in Databases. In *AAAI Workshop on Knowledge Discovery in Databases (KDD'94).* (Seattle, WA, July 1994).157-168.
8. Han, J., and Fu, Y. Discovery of multiple-level association rules from large databases. In *Proceedings of the 21st International Conference on Very Large Databases (VLDB'95).* (Zurich, Switzerland, September 11-15,1995). 420-431.
9. Han, J., Pei, J., and Yin, Y. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data.* (Dallas, Texas, USA, May 16-18, 2000).1-12.
10. Koperski, K. and Han, J. Discovery of spatial association rules in geographic information databases. In *Proceedings of the 4th International Symposium on Large Spatial Databases (SSD'95).* (Portland, Maine, USA, August 1995), 47-66.
11. Mannila, H., Toivonen, H., and Verkamo, A. I. Efficient Algorithms for Discovering Association Rules. In *Proceedings of AAAI Workshop on Knowledge Discovery in Databases (KDD'94),* (Seattle, Washington, USA, July 1994).181-192.
12. Savasere, A., Omiecinski, E, and Navathe, S. B. Mining for Strong Negative Associations in a Large Database of Customer Transactions. In *Proceedings of the 14th International Conference on Data Engineering (ICDE'98).* (Orlando, Florida, USA, February 23 - 27, 1998). IEEE Computer Society, Washington, DC, USA, 494-502.
13. Srikant, R., and Agrawal, R. Mining Generalized Association Rules. In *Proceedings of the 21st VLDB Conference,* Zurich, Switzerland, September 11-15, 1995. 407-419
14. Teng, C. M. Learning form Dissociations. In *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery (DaWaK'02).* (Aix-en-Provence,France, September 4-6, 2002). LNCS2454,11-20.
15. Thomas, S., and Sarawagi, S.. Mining generalized association rules and sequential patterns using SQL queries. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining.* (New York City, NY, USA, August 27-31,1998). 344-348.
16. Tung, A. K. H., Lu, H., Han, J., and Feng, L. Efficient Mining of Inter-transaction Association Rules. *IEEE Trans. On Knowledge And Data Engineering, 15,1,* 2003, 43-56.
17. Wu, X., Zhang, C., and Zhang, S. Efficient mining of both positive and negative association rules. *ACM Transactions on Information Systems (TOIS),* 22, 3, (July 2004), 381405.