IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
5-7 September 2005, Sofia, Bulgaria

Mining for Contiguous Frequent Itemsets in Transaction Databases

Christos Berberidis, George Tzanis and Ioannis Vlahavas
Department of Informatics, Aristotle University of Thessaloniki,
Thessaloniki 54124, Greece, tel. +302310998418, fax +302310998362
{berber, gtzanis, 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 contiguous frequent itemsets. We formulate the problem of mining contiguous frequent itemsets in a transaction database and we present a level-wise algorithm for finding these itemsets. Contiguous frequent itemsets may contain important knowledge about the dataset, that can not be exposed by the use of classic association rule mining approaches. This knowledge may well include serious hints for the generation of a taxonomy for all or part of the items.*

Keywords *- data mining, market basket analysis, frequent itemset mining, association rule.*

I. INTRODUCTION

Association rule mining has attracted the attention of the data mining research community since the early 90s, as a means of unsupervised, exploratory data analysis. The association rule 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, behaviours etc. An example of association rule might be "*90% of the customers that purchase coffee, also purchase sugar*".

Association rules are applied in many domains that range from decision support to telecommunications alarm diagnosis and prediction [6]. However, the typical application of association rules is in analysis of sales data referred to as *market basket data*. Other applications of association rules include cross marketing and attached mailing applications, catalog design, add-on sales, store layout and customer segmentation based on buying patterns [6].

The formal statement of the problem of mining association rules [6] follows. Let I be a finite set of binary attributes called *items* and D be a finite multiset of *transactions*. Each transaction $T \in D$ is a set of items such that $T \subseteq I$. A set of items is usually called an *itemset*. The *length* or *size* of an itemset is the number of items that contains. An itemset of length k is referred to as k-itemset. For an itemset $A \cup B$, if B is an m-itemset then B is called an *m-extension* of A. We say that a transaction $T \in D$ *contains* an itemset $A \subseteq I$, if $A \subseteq T$. An association rule is an implication of the form $A \Rightarrow B$, where $A \subset I$, $B \subset I$ and $A \cap B = \varnothing$. A is called the *antecedent* and B is called the *consequent* of the rule.

There are two common interestingness measures for association rules. The *support* of an association rule $A \Rightarrow B$ is a measure of statistical significance and is equal to the support of the itemset $A \cup B$, which is calculated as the fraction of the transactions that contain itemset $A \cup B$ over the total number of transactions in D:

$$\sup(A \Rightarrow B) = \sup(A \cup B) = \frac{\left|\{T \in D \mid T \supseteq A \cup B\}\right|}{|D|} \quad (1)$$

The *confidence* of an association rule $A \Rightarrow B$ is a measure of its strength and is equal to the fraction of the transactions that contain the itemset $A \cup B$ over the number of transactions that contain only A. Confidence can be calculated as follows:

$$\mathrm{conf}(A \Rightarrow B) = \frac{\sup(A \cup B)}{\sup(A)} \quad (2)$$

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.

All the association rule mining approaches so far follow

these two steps. The basic module of all approaches is the frequent itemset mining algorithm, which is also the most computationally intensive module. Moreover, it is independent from the rule generation module. Fig. 1 illustrates the general association rule mining approach.

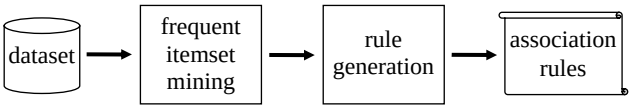

Fig. 1. The general approach for mining association rules.

One of the major problems of association rule mining is how to reduce the number of extracted rules to a small number of *interesting* ones. This can be done by setting an appropriate metric, such as the support and the confidence to use as a threshold to prune the *uninteresting* rules. Nevertheless, the use of support and confidence causes the loss of valuable knowledge. For example, an important association rule with very low support will not be extracted unless the minimum support threshold is set very low. However, by decreasing the minimum support threshold a large number of insignificant rules will also be produced.

During the last decade, a large number of algorithms have been proposed, in order to improve performance or to adjust to new needs and more complex problems. An interesting direction concerns a group of algorithms and approaches that embed a special kind of item information, called *conceptual hierarchy* or *taxonomy*. Taxonomies exist in various application domains, such as market basket analysis and the use of them provides another method to extract strong association rules. A taxonomy is a conceptual tree, where the edges represent "is-a" relationships from the child to the father. Example of such a relationship is: "Cheddar *is-a* Cheese *is-a* Dairy *is-a* Food *is-a* Product". Similarly, "Yoghurt *is-a* Dairy *is-a* Food *is-a* Product" (Fig. 2).

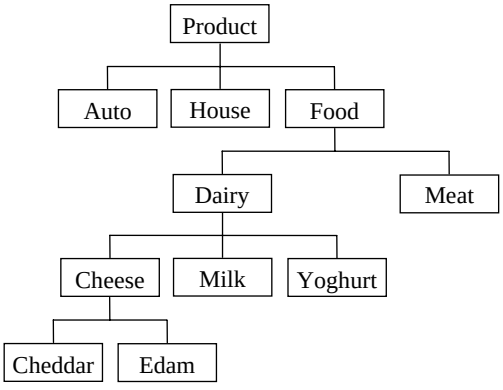

Fig. 2. Example of a taxonomy.

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 WheatBread $\Rightarrow$ SkimmedMilk is very likely to have low support, while a rule Bread $\Rightarrow$ 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. 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 serious hint regarding the existence of such information.

In this paper we propose a simple method for extending frequent itemsets and collecting information in order to summarize frequent itemsets and discover possible taxonomy information over the dataset. We define the problem of mining contiguous frequent itemsets and present the results of our experiments on various synthetic datasets. Our goal is to assist the mining procedure with knowledge that can eventually be utilized to mine for stronger association rules.

The rest of this paper is organized as follows. The next section contains a short review of the relative literature. Section 3 contains the description of the proposed approach, definitions of terms, notions used, and the proposed algorithm. In section 4 we present illustrative examples of our approach and results of our experiments. Finally, in section 5 we summarize with the conclusions we drew from this research and our ideas for future work on this topic.

II. Related Work

Association rules were first introduced in 1993 by Agrawal et al. [1]. The first algorithms for the discovery of association rules, AIS [1] and SETM [2, 3] are not very efficient, since they generate a very large number of candidate frequent itemsets. In 1994 Agrawal and Srikant [4] proposed Apriori, an algorithm which outperforms AIS and SETM. Apriori exploits the *downward closure property*, according to which any non-empty subset of a frequent itemset is also frequent. Therefore, at each step the candidate frequent itemsets are generated based only on the frequent itemsets found in the previous step. About the same time Manilla et al. [5] 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 [6]. Several algorithms have been proposed since then, others improving the efficiency, such as FPGrowth [7], and others addressing

different problems from various application domains, such as spatial [8], temporal [9] and intertransactional rules [10].

One of the major problems in association rule mining is the large number of often uninteresting rules extracted. Srikant and Agrawal [11] presented the problem of mining for generalized association rules, that utilize item taxonomies in order to discover more interesting rules. The authors propose a basic algorithm as well as some more efficient algorithms, along with a new interest measure for rules, which uses information in the taxonomy. Thomas and Sarawagi [12] propose a technique for mining generalized association rules based on SQL queries. Han and Fu [13] also describe the problem of mining "multiple-level" association rules, based on taxonomies and propose a set of top-down progressive deepening algorithms.

Another category of association rules are negative association rules. Savasere et al. [14] 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 given that a certain set of items is purchased. Wu et al. [15] proposed an efficient method for mining both positive and negative associations. Finally, Teng [16] proposed a type of augmented association rules, using negative information called dissociations. A dissociation is a relationship of the form "*A* does not imply *B*", but it could be that "when *A* appears together with *C*, this implies *B*".

III. Our Approach

Before the description of our approach it is essential to provide some definitions and formulate the problem of mining contiguous frequent itemsets. 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.

In the next lines we define the problem of mining *contiguous frequent itemsets*. 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:

$$\mathrm{lsup}(E, F) = \frac{\mathrm{gsup}(E \cup F)}{\mathrm{gsup}(F)} \tag{3}$$

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.

The importance of mining all contiguous frequent itemsets is summarized in the following two-fold intuition: First, if the extensions are frequent 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 father node in the taxonomy is broken down to many lower level supports, which are not high enough to satisfy the minimum support 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 serious hint about the taxonomy effect explained before and eventually the existence of a taxonomy.

At this point some remarks have to be noted. From equations (2) and (3) is shown that the local support of an itemset B in the A-subspace is similar to the confidence of the rule $A \Rightarrow B$. Someone, based on this observation, could argue that association rules include the knowledge exposed by contiguous frequent itemsets and consequently contiguous frequent itemsets are useless.

However, this is not true even if part of the knowledge encapsulated in a contiguous frequent itemset $F \cup E$ is also included in the association rule $F \Rightarrow E$. Using classical association rule mining approaches this rule is not generally discovered, since itemset $F \cup E$ is not always (globally) frequent. If the minimum support threshold is set very low so that the above rule can be mined, then many uninteresting rules will also be mined. For example, let the minimum global support threshold is set to 0.2 and the minimum local support threshold is set to 0.3. In order to mine all the association rules $A \Rightarrow B$ related to the contiguous frequent itemset $A \cup B$ the minimum support threshold has to be set to 0.06 ($0.2 * 0.3$), which is a significantly low value.

Another notable remark is that a frequent itemset represents a class of consumers that have particular preferences. For example, the frequent itemset A={Diapers, Milk} is very likely to represent parents that have a baby. The use of traditional association rules do not provide the alternative to explore the subset of transactions that contain A and consequently a large amount of potentially valuable knowledge remains hidden. Conversely, the contiguous frequent itemsets that contain A, reveal the hidden knowledge and is possible to generate more specific rules, that are interesting inside the A-subspace. Returning to the example, let consider itemset B = {NursingBottle, NursingBottleBrash}. It is very likely that B has very low global support, since someone does not often purchase nursing bottle and nursing bottle brash. But when we consider the customers who have a baby, it is much more likely to find these two products in their baskets. If this is the case, then B would be locally frequent in A-subspace. Moreover, the following rule NursingBottle $\Rightarrow$ NursingBottleBrash could be found that is very interesting (has high confidence) in the A-subspace. This knowledge can not be directly discovered by association rule mining.

We have implemented a level-wise algorithm in order to extract the k-contiguous frequent itemsets. The level-wise algorithm works in two steps:

1. All globally frequent itemsets are found according to a minimum global support threshold. Any frequent itemset mining algorithm can be used, to find them.
2. For each globally frequent itemset F, its subspace is mined for locally frequent itemsets, according to a local support threshold. Each locally frequent itemset E is the locally frequent extension of F.

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-extensions in a level-wise manner. This means that first all the 1-extensions for each frequent itemset will be mined, next all the 2-extensions, etc. Thus the algorithm can be stopped at any level, producing the contiguous frequent itemsets so far discovered. This can be used when the user is only interested in small extensions (i.e. of size 1 or 2) or in extreme cases when the algorithm takes too much time to terminate and an output is required quickly. The basic module of our algorithm is outlined in Table I.

TABLE I

THE CONTIGUOUS FREQUENT ITEMSET MINING FUNCTION.

Input: A multiset of transactions D, a minimum local support threshold *min_lsup* and a maximum extension size threshold *max_ext*.
Output: All k-contiguous frequent itemsets, where k ranges from 1 to *max_ext*.

```
MINE_CFI(D, min_lsup, max_ext)
(1)   FC_1 ← {1-contiguous frequent itemsets}
(2)   k ← 2
(3)   while (FC_{k-1} ≠ ∅ and k ≤ max_ext) do
(4)     C_k ← GENERATE(FC_{k-1})
(5)     FC_k ← ∅
(6)     for each C ∈ C_k do
(7)       for each E ∈ EXT(C) do
(8)         COUNT(E) ← 0
(9)     for each T ∈ D do
(10)      for each C ∈ C_k do
(11)        if (FREQ(C) ⊂ T) then
(12)          for each E ∈ EXT(C) do
(13)            if (E ⊂ T-FREQ(C)) then
(14)              COUNT(E) ← COUNT(E) + 1
(15)    for each C ∈ C_k do
(16)      min_lcount = min_lsup * COUNT(FREQ(C))
(17)      E' ← {E ∈ EXT(C)| COUNT(E) ≥ min_lcount}
(18)      if ( E' ≠ ∅ ) then
(19)        FC_k ← FC_k ∪ {(FREQ(C), E')}
(20)    k ← k + 1
(21)  return ⋃_{i=1}^{k-1} FC_i
```

Function GENERATE (line 4) generates the candidate locally frequent extensions for each frequent itemset, in the same manner as in Apriori [4]. The set of candidates at level k is based on the contiguous frequent itemsets FC_{k-1} discovered in the step k-1. A contiguous frequent itemset is represented as a pair (F, E), where F is the globally frequent itemset and E is the locally frequent extension. Functions FREQ and EXT are used to access F and E, respectively. All 1-contiguous frequent itemsets (line 1) are generated by the function outlined in Table II. The function works by reading one transaction at the time. For each frequent itemset contained in a transaction the remaining items in the transaction are considered as candidate frequent extensions and their occurrences are counted. Finally, function COUNT is used to access the frequency (number of occurrences) of an itemset.

TABLE II

THE 1-CONTIGUOUS FREQUENT ITEMSET MINING FUNCTION.

Input: A multiset of transactions *D*, a set of frequent itemsets *FD* and a minimum local support threshold *min_lsup*.
Output: All 1-contiguous frequent itemsets.

MINE_1-CFI (*D*, *FD*, min_*lsup*)
(1) $C_1 \leftarrow \varnothing$
(2) $FC_1 \leftarrow \varnothing$
(3) **for each** $F \in FD$ **do**
(4) $C_1 \leftarrow C_1 \cup \{(F, \varnothing)\}$
(5) **for each** $T \in D$ **do**
(6) **for each** $C \in C_1$ **do**
(7) **if** (FREQ(C) $\subset T$) **then**
(8) **for each** $I \in T$-FREQ(C) **do**
(9) **if** ($I \in$ EXT(C)) **then**
(10) COUNT(I) $\leftarrow$ COUNT(I) + 1
(11) **else**
(12) EXT(C) $\leftarrow$ EXT(C) $\cup$ {I}
(13) COUNT(I) $\leftarrow$ 1
(14) **for each** $C \in C_1$ **do**
(15) *min_lcount* = *min_lsup* $*$ COUNT(FREQ(C))
(16) $E' \leftarrow \{E \in \text{EXT}(C) | \text{ COUNT}(E) \geq min\_lcount\}$
(17) **if** ($E' \neq \varnothing$) **then**
(18) $FC_1 \leftarrow FC_1 \cup \{(\text{FREQ}(C), E')\}$
(19) **return** FC_1

IV. EXPERIMENTS

We illustrate our approach using an example dataset shown in Table III. The dataset contains 6 transactions and 13 different items. The items have been replaced by integers and with each transaction is associated a transaction ID (TID).

TABLE III

EXAMPLE OF A MARKET BASKET DATASET.

TID	Items
1	1, 3, 4, 2
2	5, 6, 2, 7, 1
3	8, 9, 10, 1, 2, 3
4	5, 11, 12
5	13, 1, 2
6	5, 11, 1, 7, 3

Applying our algorithm with minimum global support threshold min_gsup = 0.6 and minimum local support threshold min_lsup = 0.4 we discover the contiguous frequent itemsets listed in Table . The frequent itemsets and the locally frequent extensions that constitute each contiguous frequent itemset along with their corresponding supports are presented.

From Table IV we can see that frequent itemset {1,2} is extended by itemset {3}, with local support 0.5. Moreover, we observe that ten contiguous frequent itemsets have been produced in total. Three of them are 2-contiguous frequent itemsets ($\{1\} \cup \{2,3\}$, $\{1\} \cup \{5,7\}$ and $\{2\} \cup \{1,3\}$) and all the others are 1-contiguous frequent itemsets. According to set theory, itemset $\{1\} \cup \{2,3\}$ is equal to itemset $\{2\} \cup \{1,3\}$. However, if these two itemsets are considered as contiguous frequent itemsets, they are different. The first ($\{1\} \cup \{2,3\}$) means that itemset {2,3} is a frequent extension of the frequent itemset {1}, while the second ($\{2\} \cup \{1,3\}$) means that itemset {1,3} is a frequent extension of the frequent itemset {2}. The following observation clarifies more the difference between the two itemsets. If, for example, we had set min_lsup equal to 0.5, then the first itemset would not have been mined. Similarly, if we had set min_gsup equal to 0.8, then the second one would not have been discovered.

TABLE IV

CONTIGUOUS FREQUENT ITEMSETS MINED FROM DATASET LISTED IN TABLE III (MIN_GSUP = 0.6, MIN_LSUP = 0.4).

Frequent Itemsets		Extensions	
Itemset	*Global Support*	*Itemset*	*Local Support*
{1}	0.83	{2}	0.8
		{3}	0.6
		{5}	0.4
		{7}	0.4
		{2,3}	0.4
		{5,7}	0.4
{2}	0.67	{1}	1
		{3}	0.5
		{1,3}	0.5
{1,2}	0.67	{3}	0.5

The graphical user interface in Fig. 3 allows for setting three mining parameters (global support, local support and maximum extension size) and displays the results using a tree structure.

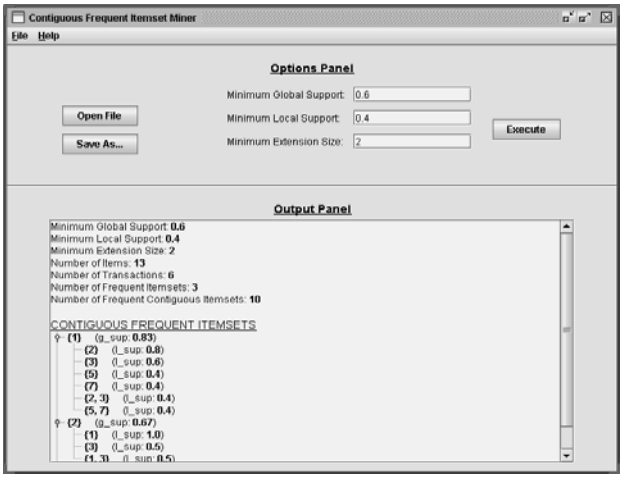

Fig. 3. Contiguous frequent itemset miner interface.

Another illustrative example of the use of our algorithm follows. Table V contains an example market

basket dataset and Table VI shows the discovered association rules. The minimum support and minimum confidence thresholds were set to 2/9 and 2/3 respectively in order to extract these rules. After applying our algorithm, we discovered the contiguous frequent itemsets shown in Table VII, along with their supports. They are all 1-contiguous frequent itemsets of the frequent 1-itemset {sugar} and the locally frequent extensions are all different types of coffee (espresso, cappuccino and decaffeinated). We observe that the above extensions are never contained in the same transaction. This observation strengthens the possibility for these items to be members of the same category (i.e. the category coffee).

TABLE V

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 VI

ASSOCIATION RULES MINED FROM THE DATASET OF TABLE V.

Association Rules	Support	Confidence
espresso $\Rightarrow$ sugar	3/9	1
decaf $\Rightarrow$ sugar	2/9	2/3
cappuccino $\Rightarrow$ sugar	2/9	2/3

TABLE VII

CONTIGUOUS FREQUENT ITEMSETS MINED FROM DATASET LISTED IN TABLE V (MIN_GSUP = 7/9, MIN_LSUP = 2/7).

Frequent Itemsets		Extensions	
Itemset	Global Support	Itemset	Local Support
{sugar}	7/9	{espresso}	3/7
		{cappuccino}	2/7
		{decaf}	2/7

The possible taxonomy derived by this analysis is depicted in Fig. 4. When we replaced these items with a single item named coffee, we were able to increase the minimum support threshold in order to acquire stronger association rules. (Table VIII).

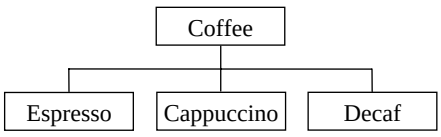

Fig. 4. A taxonomy of coffee products.

TABLE VIII

ASSOCIATION RULES MINED FROM THE DATASET OF TABLE V AFTER REPLACING THE THREE TYPES OF COFFEE BY ONE ITEM.

Association Rules	Support	Confidence
coffee $\Rightarrow$ sugar	7/9	7/9
sugar $\Rightarrow$ coffee	7/9	1

In order to evaluate the performance of our algorithm we conducted a number of experiments on synthetic data. We used a market basket dataset that contains 500 transactions. The average number of items contained in a transaction is 20, while the variance is ± 15 items. The above dataset is available in the Web [17]. The graph in Fig. 5 illustrates the performance of our algorithm in means of response time (milliseconds) while the minimum global support threshold (min_gsup) varies from 0.04 down to 0.01. The minimum local support threshold was set to 0.3. We observe that while the min_gsup decreases, the response time of the algorithm increases. This is expected, since lower values of min_gsup result more frequent itemsets to be discovered and consequently more possible extensions.

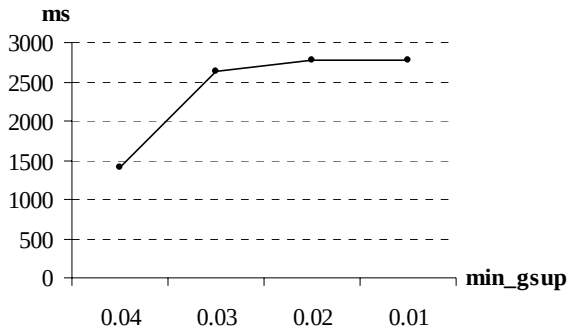

Fig. 5. Response time of our algorithm.

V. CONCLUSIONS

In this paper we presented the novel problem for mining contiguous frequent itemsets, as an extension to the association rule mining paradigm. Contiguous frequent itemsets may contain important knowledge about the dataset that is not included in the traditional association rules. Moreover, contiguous frequent itemsets provide important information to the domain expert for the construction of a taxonomy. For this purpose, we developed a level-wise algorithm and we applied it on a number of synthetic datasets in order to be tested for its validity and performance. It is within our current plans to apply it on a number of real world datasets as well as to improve it in terms of speed and memory management.

REFERENCES

[1] R. Agrawal, T. Imielinski, A. Swami. Mining association rules between sets of items in large databases. *Proceedings of the "ACM SIGMOD International Conference on Management of Data"*, Washington, DC, USA, May 26-28, 1993, pp. 207-216.

[2] M. Houtsma, A. Swami, *Set-Oriented Mining of Association Rules*, Research Report RJ 9567, IBM Almaden Research Center, San Jose, California, USA, October 1993.

[3] M. Houtsma, A. Swami. Set-oriented mining for association rules in relational databases. *Proceedings of the "International Conference on Data Engineering (ICDE'95)"*, Taipei, Taiwan, March 6-10, 1995, pp. 25-33.

[4] R. Agrawal, R. Srikant. Fast algorithms for mining association rules in large databases. *Proceedings of the "International Conference on Very Large Databases (VLDB'94)"*, Santiago de Chile, Chile, September 12-15, 1994, pp. 487-499.

[5] H. Mannila, H. Toivonen, A. I. Verkamo. Efficient Algorithms for Discovering Association Rules. *Proceedings of "AAAI Workshop on Knowledge Discovery in Databases (KDD'94)"*, Seattle, Washington, USA, July 1994, pp. 181-192.

[6] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. 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. p. 307–328.

[7] J. Han, J. Pei, Y. Yin. Mining frequent patterns without candidate generation. *Proceedings of the "ACM SIGMOD International Conference on Management of Data"*, Dallas, Texas, USA, May 16-18, 2000, pp. 1-12.

[8] K. Koperski, J. Han. Discovery of spatial association rules in geographic information databases. *Proceedings of the "International Symposium on Large Spatial Databases (SSD'95)"*, Portland, Maine, USA, August 6-9, 1995, pp. 47-66.

[9] X. Chen, I. Petrounias. Discovering Temporal Association Rules: Algorithms, Language and System. *Proceedings of the "International Conference on Data Engineering (ICDE'00)"*, San Diego, California, USA, February 28 - March 03, 2000, pp. 306.

[10] A. K. H. Tung, H. Lu, J. Han, L. Feng, Efficient Mining of Intertransaction Association Rules, *IEEE Transactions On Knowledge And Data Engineering* 15 (1) (2003). p. 43-56.

[11] R. Srikant, R. Agrawal. Mining Generalized Association Rules. *Proceedings of the "International Conference on Very Large Databases (VLDB'95)"*, Zurich, Switzerland, September 11-15, 1995, pp 407-419.

[12] S. Thomas, S. Sarawagi. Mining generalized association rules and sequential patterns using SQL queries. *Proceedings of the "International. Conference on Knowledge Discovery and Data Mining (KDD'98)"*, New York, USA, August 27-31, 1998, pp. 344-348.

[13] J. Han, Y. Fu. Discovery of multiple-level association rules from large databases. *Proceedings of the "International Conference on Very Large Databases (VLDB'95)"*, Zurich, Switzerland, September 11-15, 1995, pp. 420-431.

[14] A. Savasere, E. Omiecinski, S. B. Navathe. Mining for Strong Negative Associations in a Large Database of Customer Transactions. *Proceedings of the "International Conference on Data Engineering (ICDE'98)"*, Orlando, Florida, USA, February 23 - 27, 1998, pp. 494-502.

[15] X. Wu, C. Zhang, S. Zhang. Efficient mining of both positive and negative association rules. *ACM Transactions on Information Systems* 22 (3) (2004). p. 381 - 405.

[16] C. M. Teng. Learning form Dissociations. *Proceedings of the "International Conference on Data Warehousing and Knowledge Discovery (DaWaK'02)"*, Aix-en-Provence, France, September 4-6, 2002, pp. 11-20.

[17] Knowledge Discovery and Management Laboratory, Flinders University - Various Datasets and Routines Relating to Rule Visualisation Work. http://kdm.first.flinders.edu.au/IDM/data.html