Frequent sets with negative information

I. Fortes[1], J.L. Balcázar[2] and R. Morales[3]

[1] Dept. Applied Mathematic, E.T.S.I. Informática, Univ. Málaga. Campus Teatinos. 29071 Málaga, Spain
ifortes@ctima.uma.es
[2] Dept. LSI, Univ. Politècnica de Catalunya. Campus Nord. 08034 Barcelona, Spain
balqui@lsi.upc.es
[3] Dept. Languages and Computer Science, E.T.S.I. Informática, Univ. Málaga. Campus Teatinos. 29071 Málaga, Spain
morales@lcc.uma.es

1 Introduction

One of the most relevant subroutines in applications of data mining is finding frequent itemsets within the transactions in the database. This task consists of finding highly frequent itemsets, by comparing their frequency of occurrence within the given database with a given parameter σ. This problem can be solved by the well-known Apriori algorithm [2].

The frequent sets that result from this task can be used then to discover association rules that have support and confidence values no smaller than the user-specified minimum thresholds [1], or to solve other related Knowledge Discovery problems [9]. We do not discuss here how to form association rules from frequent itemsets, nor any other application of these; but focus on the performance of that very step, finding highly frequent patterns, whose complexity dominates by far the computational cost of many such applications.

Here we considered the case where each transaction of the database is a binary-valued function of the attributes. The difference with the itemsets view is that now we look for patterns where the non-occurrence of an item is important too. This is formalized in terms of partial functions, which, on each item, may include it (value 1), exclude it (value 0), or not to consider it (undefined).

It is known that direct use of the Apriori algorithm on real life data frequently come up with extremely large numbers of frequent sets consisting "only of zeros"; for example, in the prototypical case of market basket data, certainly the number of items is overwhelmingly larger than the average number of items bought, and this means that the output of any frequent sets algorithm will contain large amounts of information of the sort "most of the times that scotch is not bought, bourbon is not bought either, with large support". If such negative information is not desired at all, the original Apriori version can be used; but there may be cases where limited amounts of negative information are deemed useful, for instance looking for alternative products that can act as mutual replacements, and yet one does not want to be forced into a search through the huge space of all partial functions.

We are interested in producing algorithms that will provide frequent "itemsets" that have "missing" products, but in a controlled manner, so that they are useful when having some missing products in the itemsets is important but not so much as the products that are in the itemsets.

Here we present a certain theory on a certain formal language according to a certain predicate that is monotone on a generalization/specialization relation. We present results to obtain the variant of the Apriori algorithm that, if supplied with a limit k on the maximum number of negated attributes desired in the output frequent sets, will take advantage of this fact, and produce frequent itemsets for which this limit is obeyed. Of course, it does so in a much more efficient way than just applying Apriori and discarding the part of the output that does not fulfill this condition. First, because the exploration is organized in a way that naturally reflects the condition on the output. Second, because we know that items may be, or not be, in each itemset, but not both implies complementarity relationships between the frequencies of "itemsets" that contain, or do not contain, a given item. We use these relationships to find out frequencies of some "itemsets" without actually counting them, thus saving computational work.

We propose bottom-up algorithms in which the exploration of facts corresponding to items not being in the transactions is delayed with respect to positive information of items being in the transactions. We have called these algorithms Neg–Apriori and Acot–Neg–Apriori.

A preliminary version of this work can be found in [6] and the full version in my PhD [7].

2 Some definitions

We consider a database $\mathcal{T} = \{t_1, \ldots, t_N\}$ with N rows over a set $R = \{A_i : i \in I\}$ of binary-valued attributes, that can be seen as either items or columns; actually they just serve as a visual aid for their index set $I = \{1, \ldots, n\}$.

Definition 1 *Let $p \in \mathcal{P}(I)$ be arbitrary and $s \in \mathcal{P}(I-p)$, arbitrary we denote the subset $A^{p,s}$ called* itemset *and identify it with the partial function mapping the subset $A^p = \{A_i : i \in p\}$ to 1, the subset $A^s = \{A_j : j \in s\}$ to 0 and undefined on the rest. Itemsets $A^{p,s}$ are called k-negative itemsets where $|s| = k$, $k = 0, \ldots, n$. If $|s| = 0$ then we have the positive itemset $A^{p,\emptyset}$.*

A transaction can be seen as a total function. An itemset can be seen as a partial function. If the partial function can be extended to the total function corresponding to a transaction then we say that an itemset is a subset of a transaction.

We identify partial functions defined on a single attribute A_j, namely, $A^{\{j\},\emptyset}$ or $A^{\emptyset,\{j\}}$, with the corresponding symbol A_j or $\overline{A_j}$ respectively.

Definition 2 *We define the specialization relation in $\mathcal{I} = \{A^{p,s} \,/\, p \in \mathcal{P}(I),\ s \in \mathcal{P}(I-p)\}$ denoted by $\preccurlyeq$ as follows: if $X = A^{p,s}$ and $Y = A^{q,t}$ are itemsets from $\mathcal{I}$, we say that $X \preccurlyeq Y$ iff $p \subseteq q$ y $s \subseteq t$.*

With respect to this relation, the property of having frequency larger than any threshold is antimonotone, since $X \preccurlyeq Y$ implies $fr(X) \geq fr(Y)$. Thus, whenever an itemset is not frequent enough, neither is any of its extensions, and this fact allows one to prune away a substantial number of unproductive itemsets. Therefore, frequent sets algorithms can be applied rather directly to this case. Our purpose now is to aim at a somewhat more refined algorithm.

The support of an itemset is defined as follows.

Definition 3 *Let $R = \{A_i : i \in I\}$ be a set of n items and let $\mathcal{T} = \{t_1, \ldots, t_N\}$ be a database of transactions as before. The support or frequency of an itemset X is the ratio of the number of transactions on which it occurs as a subset to the total number of transactions. Therefore:*

$$fr(X) = \frac{|\{t \in \mathcal{T} : X \preccurlyeq t\}|}{N}$$

Given a user-specified minimum support value (denoted by σ), we say than an itemset A is *frequent* if its support is more than the minimum support, i.e. $fr(X) \geq \sigma$.

3 Representing Itemsets

We introduce a natural structure in the itemset space by placing them into "floors" and "levels". The floor k contains itemsets with k negative attributes. In each floor, the itemsets are organized in levels (as usual): the level is the number of the attributes of the itemset. Using the specialization relation we organized and related the itemsets.

Now, we give a simple example to show the structure of the itemset space. This example will be useful to describe the frequent itemset candidate generation and the path that follows our algorithm for it.

Example: Let $R = \{A, B, C, D\}$ be the set of four items. In this case, we use four floors to represent the itemsets with any number of negative attributes and any number of positive attributes. In each rectangle, the pair (f, ℓ) indicates the floor f (number of negative attributes in the itemsets of this rectangle) and level ℓ (cardinality of the itemsets of this rectangle). See figure 1.

4 Calculating frequencies

Our algorithm performs the same computations as Apriori on the zero floor, but then uses the frequencies computed to try to reduce the computational effort spent on 1-negative itemsets. This process goes on along all floors. Overall, bounded-neg-Apriori can be seen as a refinemet of Apriori in which the explicit evaluation of the frequency of k-negative itemsets is avoided, since it can be obtained from some itemsets of the previous floor, if they are processed in the appropriate order. This idea is based on the following theorem, that is the key of our approach.

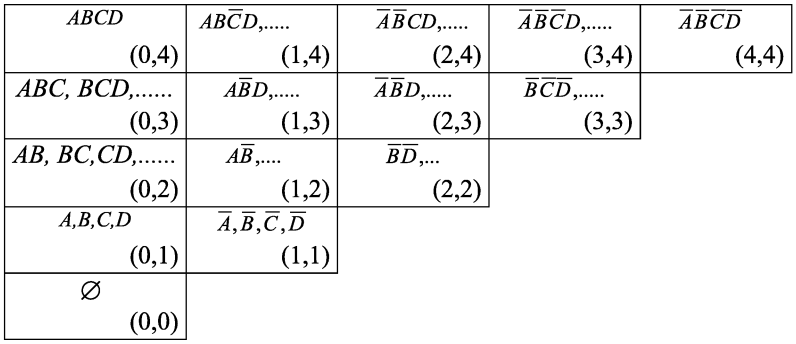

Fig. 1. The structure of the itemset space

Theorem 1 *Let $p \in \mathcal{P}(I)$ be arbitrary, and $s \in \mathcal{P}(I-p)$ with $|s| \geq 1$. Then for each $j \in s$,*

$$fr(A^{p,s}) = fr(A^{p,s-\{j\}}) - fr(A^{p\cup\{j\},s-\{j\}})$$

Proposition 1 *Let $p \in \mathcal{P}(I)$ be arbitrary, and $s \in \mathcal{P}(I-p)$ with $|s| \geq 1$. $A^{p,s}$ is frequent iff $\exists j \in s$, $fr(A^{p,s-\{j\}}) > \sigma + fr(A^{p\cup\{j\},s-\{j\}})$.*

Remark 1: Each of the up to $|s|$-many ways of decomposing $fr(A^{p,s})$ in part 1 leads to the same result: if $fr(A^{p,s-\{j\}}) < \sigma$, for any $j \in s$, then $A^{p,s}$ is not frequent.

5 Generating candidates

Moving to the next round of candidates once all frequent ℓ-itemsets have been identified corresponds to moving up, in all possible ways, one step within the same floor, and climbing up in all possible ways to the next floor.

More formally, at the floor zero, frequent set $A^{p,\emptyset}$ leads to consideration as potential candidates of the following itemsets: all $A^{q,\emptyset}$ where $q = p \cup \{i\}$ and all $A^{p,\{j\}}$, for $j \notin p$. Also, itemset $A^{p,\{j\}}$ would lead to $A^{q,\{j\}}$ for $q = p \cup \{i\}$, for $i \notin p$ and $i \neq j$; our algorithm does not use this last sort of steps.

In the other floors the movements are in the same form. For all $p \in \mathcal{P}(I)$ and $s \neq \emptyset$, from $A^{p,s}$ we can climb up to the next floor to $A^{p,t}$ where $t = s \cup \{j\}$, for $j \in \mathcal{P}(I - \{s \cup p\})$. Also, itemset $A^{p,s}$ would lead to $A^{q,s}$ for $q = p \cup \{i\}$, for $i \notin p$ and $i \notin s$ but we will not use such steps either.

Therefore the scheme of the search of frequent itemsets with k 0-valued attributes (i.e. in the floor k) is based on the following: whenever enough frequencies in the previous floor are known to test it, if $fr(A^{p,s-\{j\}}) > \sigma + fr(A^{p\cup\{j\},s-\{j\}})$ where $j \in s$, then we know $fr(A^{p,s}) > \sigma$ so that it can be

declared frequent; moreover, for $\sigma > 0.5$ this has to be tested only when that $A^{p\cup\{j\},s-\{j\}}$ turned out to be nonfrequent although $A^{p,s-\{j\}}$ was frequent.

Example: Let us turn our atention again to the example. Let us suppose that $\sigma < 0.5$; we explain the process of candidate generation and the path that our algorithm follows for it. Suppose that the maximal itemsets to be found are ABC, $AB\overline{C}$, and $A\overline{B}$. Thus, A, B, C are frequent items, and also $\overline{B}$ and $\overline{C}$ are frequent 'negative items'. At the initialization, we find that D, $\overline{A}$, and $\overline{D}$ cannot appear in any frequent itemset. The algorithm stores this information by means of the set I (defined later). In the following step, we take into consideration as potential candidates, firstly the itemsets in $(0,2)$, secondly in $(1,2)$, and at last, in $(2,2)$ that verify the conditions. There we find the frequent itemsets are AB, AC, BC, $A\overline{B}$, $A\overline{C}$, $B\overline{C}$. At this moment, we know that there do not exist frequent itemsets in $(2,2)$. So, there will not exist frequent itemsets in (f,ℓ) with $f \geq 2$, $\ell > 2$ and $\ell \geq f$. This information is used in the algorithm by means of the set J (defined later) to refine the search of candidate generation. In the following step we scan for frequent itemsets in $(0,3)$ and $(1,3)$ and ABC, $AB\overline{C}$ are frequent itemsets, and the exploration of the next level proves that, together with $A\overline{B}$, they are the maximal frequent itemsets. Along the example it is clear how the algorithm would proceed in case we are given a bound on the number of negative attributes present: this would just discard floors that do not obey that limitation.

We will also use the following easy properties regarding the relation of the threshold σ to the value one-half. They allow for some extra pruning to be done for quite high frequency values (although this case might be infrequently occurring in practice).

Proposition 2 *For each $A \in R$ following properties hold:*

1. $|fr(A) - 0,5| < |\sigma - 0,5| \Leftrightarrow |fr(\overline{A}) - 0,5| < |\sigma - 0,5|, \quad \forall\, \sigma \in [0,1]$.
2. *If $\sigma < 0,5$ then*

$$fr(A) \leq \sigma \;\Rightarrow\; fr(\overline{A}) > \sigma \quad y \quad fr(A) > 1-\sigma \;\Leftrightarrow\; fr(\overline{A}) < \sigma.$$

3. *If $\sigma = 0,5$ then*

$$fr(A) \geq \sigma \;\Leftrightarrow\; fr(\overline{A}) \leq \sigma \quad y \quad fr(A) \leq \sigma \;\Leftrightarrow\; fr(\overline{A}) \geq \sigma.$$

4. *If $\sigma > 0,5$ then*

$$fr(A) \geq \sigma \;\Rightarrow\; fr(\overline{A}) < \sigma \quad y \quad fr(A) < 1-\sigma \;\Leftrightarrow\; fr(\overline{A}) > \sigma.$$

For itemsets with cardinal more than one we have the following proposition

Proposition 3 *Let $p \in \mathcal{P}(I)$ be arbitrary and $s \in \mathcal{P}(I-p)$, arbitrary for statements not depending on p. If $\sigma > 0,5$ following properties hold:*

1. $\forall j \in s$, *if* $fr(A^{p,s-\{j\}}) > \sigma + fr(A^{p\cup\{j\},s-\{j\}})$ *then* $fr(A^{p\cup\{j\},s-\{j\}}) < \sigma$.

2. *If* $\exists j \in s \,/\, fr(A^{p\cup\{j\},s-\{j\}}) > 1-\sigma > \sigma$ *then* $fr(A^{p,s}) < \sigma$.

Proposition 4 *Let* $p \in \mathcal{P}(I)$ *be arbitrary and* $s \in \mathcal{P}(I-p)$*, arbitrary for statements not depending on* p.

If $\sigma > 0,5$ *then*

$$\sum_{x\subseteq s, x\neq\emptyset} fr(A^{p\cup x, s-x}) > 1-\sigma > \sigma \quad \Longrightarrow \quad fr(A^{p,s}) < \sigma$$

Also, it is useful:

Proposition 5 *If* $fr(A^{\emptyset,I}) = 0$ *then for each* $i \in I$

1. $fr(A^{\{i\},I-\{i\}}) = fr(A^{\emptyset,I-\{i\}})$.
2. *If* $fr(A^{\emptyset,I-\{i\}}) > \sigma$ *then for each* $j \in I-\{i\}$, $fr(A^{\{i\},I-\{i,j\}}) > \sigma$.

6 The algorithm

The algorithm has as input the set of attributes, the database, and the threshold σ on the support. The output of the algorithm is the set of all frequent itemsets with negative and positive itemsets. Also, a similar algorithm can be easily developed to find the set of all frequent itemsets with at most k negative attributes: simply impose explicitly the bound k on the corresponding loop in the algorithm.

With respect to this notation our algorithm traces the following path:
$(0,1),(1,1);(0,2),(1,2),(2,2);(0,3),(1,3),(2,3),(3,3);$, etc (recall to the example).

The algorithm refines the search of frequent itemsets by means of the set J. In each level, J indicates the floors where no frequent itemsets will exist.
The generation of candidates and the computation of their frequencies must be done by considering σ (less or more than 0.5)

If it is possible the frequencies of candidate itemsets with any number of negative attributes are obtained by using theorem one. It reduces the computational effort.

Note that, the only negative attributes that could appear in the candidate itemsets are the frequent elements of the cell $(1,1)$. So, we use this set, as soon as it is computed, to refine the index set I used later along the computation.

The pseudo code of Neg–Apriori and Acot–Neg–Apriori algorithms can be seen in [7].

6.1 Complexity of the algorithm

With respect to the complexity of the algorithm, from a theoretical point of view, two aspects are considered: candidate generation and itemset frequence computation.

In the candidate generation the worst case is reached when the threshold σ is less or equal to 0.5. In this case, two itemsets one of them with a particular attribute positive and the other itemset with the same attribute negative can be frequent simultaneously. If $\sigma > 0.5$ then by proposition 4 the generation is refined. Independtly of the σ value the sets I and J refine the candidate generation. So, the needed requirements can be reduced.

In the itemset frequence computation only itemsets with positive attributes are computed directly from the database. The frequencies of the other candidate itemsets with any number of negative attributes are obtained by using theorem one. Therefore, the number of passes through the database is like in Apriori, i.e., $n + 1$, where n is the greatest frequent itemset.

7 Conclusions and Future Work

In cases where the absence of some items from a transaction is relevant but one wants to avoid the generation of many rules relating these absences, it can be useful to allow for a maximum of k such absences from the frequent sets; even if no good guess exists for k, it may be useful to organize the search in such a way that the itemsets with m items show up in the order mandated by how many of them are positive: first all positive, then $m - 1$ positive and one negative, and so on. Our algorithm allows one to do it and takes advantage of a number of facts, corresponding to relationships between the itemset frequencies, to avoid the counting of some candidates.

Of course, it makes sense to try to combine this strategy together with other ideas that have been used together with Apriori, like random sampling to evaluate the frequencies, or instead of Apriori, like alternative algorithms such as DIC [4] or Ready-and-Go [3]. Also, we will study how to integrate our approach of finding frequent itemsets with negative information in the GRD algorithm [10] to fin ghe k–most interesting negative rules. Experimental developments can lead to improved results, and we continue to work along this line.

Another natural line is the study of frequent negative information in sequences.

References

1. Agrawal R., Imielinski T., Swami A.N.: Mining association rules between sets of items in large databases. *Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD'93)*, ACM Press Washington D.C. , May 26-28 (1993) 207–216.
2. Agrawal R., Mannila H., Srikant R., Toivonen H., Verkamo A.I.: Fast discovery of association rules, in Fayyad U.M., Piatetsky-Shapiro G., Smyth Rp., Uthurusamy R. Eds, *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA; (1996) 307–328.
3. Baixeries J., Casas-Garriga G. and Balcázar J.L.: Frequent sets, sequences, and taxonomies: new, efficient algorithmic proposals. Tech. Rep. LSI-00-78-R. UPC. Barcelona (2000).

4. Brin S., Motwani R., Ullman J.D., Tsur S.: Dynamic Itemset Counting and Implication Rules for Market Basket Data. *Int. Conf. Management of Data*, ACM Press (1997) 255–264.
5. Fayyad U.M., Piatetsky–Shapiro G., Smyth P.: From data mining to knowledge discovery: An overview. In Fayyad U.M., Piatetsky–Shapiro G., Smyth P. and Uthurusamy R., eds, *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA, (1996) 1–34.
6. Fortes I. Balcázar J.L., Morales R. Bounding Negative Information in Frequent Sets Algorithms. *Lecture Notes in Artificial Intelligence*, 2226, (2001), 50–58.
7. Fortes I. Prospección de datos, aprendizaje computacional y técnicas estadísticas pra la obtención de reglas. Tesis Doctoral. Universidad de Mlaga (2002).
8. Gunopulos D., Khardon R., Mannila H., Toivonen H. Data Mining, Hypergraph Transversals, and Machine Learning. *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, ACM Press, Tucson, Arizona, May 12-14, (1997) 209–216.
9. Mannila H:, Toivonen H.: Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery.* **1(3)** (1997) 241–258.
10. Thiruvady D.R., Webb G. I.: Mining Negative Rules using GRD. H.Dai, R. Srikant, C. Zhang (Eds.) PADKDD 2004, *LNAI* 3056, (2004) 161–165.