Deducing Bounds on the Support of Itemsets

Toon Calders

University of Antwerp
Universiteitsplein 1, B-2610 Wilrijk, Belgium
toon.calders@ua.ac.be

Abstract. Mining Frequent Itemsets is the core operation of many data mining algorithms. This operation however, is very data intensive and sometimes produces a prohibitively large output. In this paper we give a complete set of rules for deducing tight bounds on the support of an itemset if the supports of all its subsets are known. Based on the derived bounds \([l, u]\) on the support of a candidate itemset \(I\), we can decide not to access the database to count the support of \(I\) if \(l\) is larger than the support threshold (\(I\) will certainly be frequent), or if \(u\) is below the threshold (\(I\) will certainly fail the frequency test). We can also use the deduction rules to reduce the size of an adequate representation of the collection of frequent sets; all itemsets \(I\) with bounds \([l, u]\), where \(l = u\), do not need to be stored explicitly. To assess the usability in practice, we implemented the deduction rules and we present experiments on real-life data sets.

1 Introduction

Mining frequent itemsets is a core operation in many data mining problems. Since their introduction [1], many algorithms have been proposed to find frequent itemsets, especially in the context of association rule mining [12].

The frequent itemset problem is stated as follows. Assume we have a finite set of items \(I\). A transaction is a subset of \(I\), together with a unique identifier. A transaction database \(D\) is a finite set of transactions. A subset of \(I\) is called an itemset. We say that an itemset \(I\) is \(s\)-frequent in a transaction database \(D\) if the number of transactions in \(D\) that contain all items of \(I\) is at least \(s\). The number of transactions that contain all items of \(I\) is called the absolute support of \(I\).

The frequent itemset problem is, given a support threshold \(s\) and a transaction database \(D\), find all \(s\)-frequent itemsets. In the remainder of the paper, we will always assume that we are working over a transaction database \(D\) with items in \(I\).

All algorithms for mining frequent itemsets rely heavily on the following monotonicity principle [16] to prune the search space:

Let \(J \subseteq I\) be two itemsets. In every transaction database \(D\), the support of \(I\) will be at most as high as the support of \(J\).
Thus, based on the support of a set that is below the support threshold, we can deduce, using the monotonicity rule, that also the support of its supersets will be below the threshold. This simple rule of deduction has successfully been used in practice. Because of the success of this simple rule, much more attention went into efficient counting schemes than into finding additional ways to prune the search space. The standard example of an algorithm exploiting this monotonicity is the well-known Apriori-algorithm \cite{2}. Apriori traverses the itemset-lattice level by level; in the $i$th loop, itemsets of cardinality $i$ are counted in the database. Because of the monotonicity principle, all itemsets in loop $i$ that have at least one subset that failed the support-test can be pruned; we know a priori that they will be infrequent. In this way we will never count itemsets that could be pruned using the monotonicity rule.

In this paper we present deduction rules, additional to the monotonicity rule, that calculate lower and upper bounds on the support of a candidate. As such, we continue work initiated in \cite{9}. Based on the supports of all subsets of an itemset $I$, the deduction rules we present, will compute bounds $[l, u]$ on the support of $I$. We show that the rules calculate the best possible such bounds; that is, both $l$ and $u$ are possible as supports of $I$, and thus, the interval cannot be made more tight. Based on these bounds we can limit the number of candidates we need to count. For example, if $l$ is above the support threshold, then we know without counting its support in the database that $I$ is frequent. If there is no need to know the support of $I$ exact, we can thus, in this case, omit counting $I$. If $u$ is below the threshold, then we know for sure that $I$ is not frequent, and we can prune it.

Besides reducing the number of candidate itemsets, we can also use the deduction rules to make concise representations \cite{15} of the frequent itemsets. We call an itemset derivable if its lower and upper bound are the same. Thus, an itemset is derivable if its support is uniquely determined by the supports of its subsets. Therefore, for the derivable itemsets, it is not necessary to count their supports. There is also no need to store them; we can later always find the missing supports with the deduction rules. Based on this observation, the NDI-representation is defined. We shortly discuss relations with other concise representations in the literature, including free sets \cite{5}, closed sets \cite{18-19}, and disjunction-free sets \cite{6}.

The organization of the paper is as follows. In Section 2 we give an example showing that the monotonicity rule is not complete for the deduction of supports. This example also gives a sketch of the general approach we follow to derive the deduction rules. In Section 3 we formally define important notions we will use throughout the paper. In Section 4, the deduction rules are given, and it is proven that they are complete. In Section 5 we present a concise representation based on the deduction rules. Section 6 gives the results of experiments with the deduction rules. In Section 7 we discuss related work and Section 8 concludes the paper.