On the Effectiveness and Efficiency of Computing Bounds on the Support of Item-Sets in the Frequent Item-Sets Mining Problem

Bassem Sayrafi, Dirk Van Gucht [*] and Paul W. Purdom
Computer Science Department
Indiana University
Lindley Hall 215
Bloomington, IN 47405, USA
{bsayrafi,vgucht,pwp}@cs.indiana.edu

ABSTRACT

We study the relative effectiveness and the efficiency of computing support-bounding rules that can be used to prune the search space in algorithms to solve the frequent item-sets mining problem (FIM). We develop a formalism wherein these rules can be stated and analyzed using the concept of differentials and density functions of the support function. We derive a general bounding theorem, which provides lower and upper bounds on the supports of item-sets in terms of the supports of their subsets. Since, in general, many lower and upper bounds exists for the support of an item-set, we show how to the best bounds. The result of this optimization shows that the best bounds are among those that involve the supports of all the strict subsets of an item-set of a particular size q. These bounds are determined on the basis of so called q-rules. In this way, we derive the bounding theorem established by Calders [5]. For these types of bounds, we consider how they compare relative to each other, and in so doing determine the best bounds. Since determining these bounds is combinatorially expensive, we study heuristics that efficiently produce bounds that are usually the best. These heuristics always produce the best bounds on the support of item-sets for basket databases that satisfies independence properties. In particular, we show that for an item-set I determining which bounds to compute that lead to the best lower and upper bounds on $\texttt{freq}(I)$ can be done in time $O(|I|)$. Even though, in practice, basket databases do not have these independence properties, we argue that our analysis carries over to a much larger set of basket databases where local "near" independence hold. Finally, we conduct an experimental study using real baskets databases, where we compute upper bounds in the context of generalizing the Apriori algorithm. Both the analysis and the study confirm that the q-rule (q odd and larger than 1) will almost always do better than the 1-rule (Apriori rule) on large dense baskets databases. Our experiment reveal that on these baskets databases, the 3-rule prunes almost 100% of the search space while, the 1-rule prunes 96% of the search space in the early stages of the algorithm. We also observe a reduction in wasted effort when applying the 3-rule to sparse baskets databases. In addition, we give experimental evidence that the combined use of the lower and upper bounds determine the exact support of many frequent item-sets without counting.

[*]The first two authors were supported by NSF Grant IIS-0082407.

OSDM'05, August 21, 2005, Chicago, Illinois, USA.

1. INTRODUCTION

We consider the relative effectiveness of various support bound rules for the *frequent item-sets mining problem* (FIM) [1, 2], as well as the problem of efficiently computing the best support bounds. The FIM problem is the following: given a set of items $\mathcal{I}$, a list $\mathcal{B}$ of subsets (*baskets*) of $\mathcal{I}$, and a nonnegative integer threshold $k \geq 0$, determine the *frequency status* for each subset of $\mathcal{I}$, that is, determine for each subset of $\mathcal{I}$ whether it is contained in at least k of the baskets in $\mathcal{B}$. A subset that satisfies (violates) this frequency condition is called a *frequent item-set* (*infrequent item-set*, respectively). Given, a threshold k, the problem asking whether there exists a frequent item-set of a certain size in a given database has been shown to be NP-complete [8].

The frequency status of an item-set I can be determined in two ways. *Counting*: count the number of baskets in $\mathcal{B}$ that contain I, and compare this count with the threshold k; or *Deduction*: infer I's frequency status from the (known) frequency status of other item-sets. The best known deduction methods for the FIM problem are based on the *monotonicity property* and, its counterpart, the *anti-monotonicity property*. The monotonicity property states that if an item-set I has a subset that is infrequent, the I is infrequent, and the anti-monotonicity property states its opposite: if an item-set I has a superset that is frequent, then I is frequent. Good examples of how these properties have been harnessed exists in the Apriori, the FP-growth, and the Eclat algorithms [2, 9, 15]. Given an nonempty item-set I whose frequency status is unknown, the Apriori Algorithm consults the frequency status of each of subsets of size $|I| - 1$. If one of these subsets is infrequent, the algorithm deduces that I is infrequent, otherwise, the algorithm determines the frequency status of I by counting.

We will determine support bounding rules which are based on properties of the support (frequency) function beyond just the monotonicity property. The bounding rules can be used in any algorithms

for solving the frequent item-set mining problem, thus our analysis are algorithm independent. We develop a general bounding theorem to approximate the support of a set from the supports of some of its subsets. We construct this theorem through the use of differentials and density functions associated with support functions.

The notion of differentials was consider by the first two authors in the study of the implication problem of differential constraints [13]. To illustrate where these inequalities arise in the FIM problem, consider a list of baskets $\mathcal{B}$ over some set of items $\mathcal{I}$. The *support function* $\mathtt{supp}$ associated with $\mathcal{B}$ gives for each item-set $I \subseteq \mathcal{I}$, the value $\mathtt{supp}(I)$ which is number of times that I is contained in the baskets in $\mathcal{B}$. Given $\mathtt{supp}$, we can reason about $\mathcal{B}$ satisfying certain types of inequalities. For example, $\mathtt{supp}(I) \geq 0$ states that the support of item-set I is nonnegative. More generally, if l and u are values in the interval $[0, 1]$, then the inequality $l|\mathcal{B}| \leq \mathtt{supp}(I) \leq u|\mathcal{B}|$ states that $\mathcal{B}$ has at least $l|\mathcal{B}|$, but not more than $u|\mathcal{B}|$, baskets containing I. These types of support inequalities were studied by Calders and Paredaens [5, 7]. Here we study inequalities of a different type. Consider item-sets K, L_1, and L_2 of $\mathcal{I}$. The inequality $\mathtt{supp}(K) \geq \mathtt{supp}(I)$ states that K occurs at least as frequently as I in $\mathcal{B}$. A more subtle example satisfied by $\mathtt{supp}$ is the inequality $\mathtt{supp}(K) - \mathtt{supp}(K \cup L_1) \geq \mathtt{supp}(K \cup L_2) - \mathtt{supp}(K \cup L_1 \cup L_2)$, which can be proved by an inclusion-exclusion argument. This inequality can be used as a lower bound for $\mathtt{supp}(K \cup L_1 \cup L_2)$ [11]. Observe that we can write these two last inequalities as finite difference equations:

$$\begin{aligned}&\mathtt{supp}(K) - \mathtt{supp}(K \cup L_1) \geq 0.\\ &(\mathtt{supp}(K) - \mathtt{supp}(K \cup L_1))\\ &\quad -(\mathtt{supp}(K \cup L_2) - \mathtt{supp}(K \cup L_1 \cup L_2)) \geq 0.\end{aligned}$$

One of the main results of this paper is a general support bounding theorem which we derive through the use of the differentials of support functions. This is done in Section 3. From this theorem stems a class of support bounding rules that can be used in the deduction of the support status of an item-set. Within this class of rules, we derive special rules (so called q-rules) that lead to the best support bounds. The class of q-rules was previously considered by Calders in his work on deduction rules for the FIM problem [6, 5]. Since finding the best support bounds on the support of an item-set is combinatorially expensive, we propose heuristics that can lead to good approximations of these best bounds and that can be computed efficiently. In Section 4, a detailed analysis of the bounding theorem reveals that the anti-monotonicity property will outperform other bounds except in certain situations where the data is highly frequent. In Section 5, we give a complete solution for the problem of determining which lower and upper bounds are best in the context of basket databases that satisfy independence properties. In particular, we show that for an item-set I, this can be done in $O(|I|)$ (Proposition 4.3). Given these theoretical insights, we develop heuristics based on these ideas (Section 5), and provide experimental results where the heuristics are used (Section 6). The results of these experiments underscore that the theoretical results predict well what happens on real-world basket databases, and that introducing these heuristics in FIM algorithms leads to improved algorithms and that the overhead incurred is small.

$\mathcal{I}$	nonempty finite set set of *items*
I, J, K, L	subsets of $\mathcal{I}$ *item-sets*
$\mathcal{L}$	set of item-sets of $\mathcal{I}$
$[X, Y]$	the set $\{U \mid X \subseteq U \subseteq Y\}$
$A_1 \ldots A_n$	the set $\{A_1, \ldots, A_n\}$
$X \subset Y$	X is a *strict* subset of Y
$\overline{X}$	the set $\mathcal{I} - X$

Table 1: Notations

2. PRELIMINARIES

We review the definitions of the density, support, and frequency functions associated with a basket database. In addition, we will introduce the notion of the *differential* of the support function. Differentials are at the core of deriving bounds on the support (frequency) of item-sets. For ease of reference, in Table 2, we collect some notations used in the paper.

2.1 Density, support, and frequency functions of basket databases

DEFINITION 2.1. *Let $\mathcal{D} = (\mathcal{I}, \mathcal{B})$ be a basket database. The* density*, the* support*, and the* frequency *functions associated with $\mathcal{D}$ are defined such that for each $I \subseteq \mathcal{I}$,*

$$\begin{aligned}\mathtt{dens}(I) &= |\{i \mid I = \mathcal{B}[i]\}|\\ \mathtt{supp}(I) &= |\{i \mid I \subseteq \mathcal{B}[i]\}|\\ \mathtt{freq}(I) &= \tfrac{\mathtt{supp}(I)}{|\mathcal{B}|}\end{aligned}$$

The density and support functions of a basket database are related in the following way ($\mathtt{dens}$ is the *Möbius inverse* of $\mathtt{supp}$):

$$\mathtt{supp}(I) = \sum_{I \subseteq J \subseteq \mathcal{I}} \mathtt{dens}(J) \quad (1)$$

$$\mathtt{dens}(I) = \sum_{I \subseteq J \subseteq \mathcal{I}} (-1)^{|J|-|I|} \mathtt{supp}(J) \quad (2)$$

2.2 Differentials of support functions

Reconsider the following inequalities discussed in the introduction:

$$\mathtt{supp}(K) \geq 0 \quad (3)$$

$$\mathtt{supp}(K) - \mathtt{supp}(K \cup L_1) \geq 0 \quad (4)$$

$$\mathtt{supp}(K) - \mathtt{supp}(K \cup L_1) - \mathtt{supp}(K \cup L_2) + \mathtt{supp}(K \cup L_1 \cup L_2) \geq 0 \quad (5)$$

We can write these inequalities in a single format as follows:

$$\sum_{\mathcal{J} \subseteq \mathcal{L}} (-1)^{|\mathcal{J}|} \mathtt{supp}(K \cup \bigcup_{J \in \mathcal{J}} J) \geq 0,$$

where for inequality (3), $\mathcal{L} = \emptyset$, for inequality (4), $\mathcal{L} = \{L_1\}$, and for inequality (5), $\mathcal{L} = \{L_1, L_2\}$. This leads to the definition of differentials first considered in [12, 13, 14].

DEFINITION 2.2. *Let $\mathcal{L}$ be a set of item-sets of $\mathcal{I}$, and let $\mathtt{supp}$ be the support function of a basket database $\mathcal{D} = (\mathcal{I}, \mathcal{B})$. The* $\mathcal{L}$-differential *of $\mathtt{supp}$, denoted $D^{\mathcal{L}}_{\mathtt{supp}}$, is the function from $2^{\mathcal{I}}$ into $\mathbf{N}$, such that for $K \subseteq \mathcal{I}$,*

$$D^{\mathcal{L}}_{\mathtt{supp}}(K) = \sum_{\mathcal{J} \subseteq \mathcal{L}} (-1)^{|\mathcal{J}|} \mathtt{supp}(K \cup \bigcup_{J \in \mathcal{J}} J).$$

EXAMPLE 2.3. Let $\mathcal{I} = \{A, B, C, D\}$. Then,

$$D^{\{B,CD\}}_{\mathtt{supp}}(A) = \mathtt{supp}(A) - \mathtt{supp}(AB) - \mathtt{supp}(ACD) + \mathtt{supp}(ABCD).$$

As shown in [13], density functions and differentials are related. The concepts of *witness sets* and *lattice decompositions* are crucial in establishing their relationship.

DEFINITION 2.4. *A set W is a* witness set *of $\mathcal{L}$ ($\mathcal{L}$ is a set of some subsets of $\mathcal{I}$) if (1) $W \subseteq \bigcup_{L\in\mathcal{L}} L$ and (2) W has a nonempty intersection with each set in $\mathcal{L}$: $\forall L \in \mathcal{L} : W \cap L \neq \emptyset$. The set of all witness sets of $\mathcal{L}$ is denoted by $\mathcal{W}(\mathcal{L})$.*

Let K be a subset of $\mathcal{I}$. The lattice decomposition $\mathbf{L}(K, \mathcal{L})$ *of the pair $(K, \mathcal{L})$ is defined as follows:*

$$\mathbf{L}(K, \mathcal{L}) = \bigcup_{W\in\mathcal{W}(\mathcal{L})} [K, \overline{W}].$$

As shown in [13], the relationship between differentials and density function can now be formulated as follows:

THEOREM 2.5 (DIFFERENTIAL DECOMPOSITION THEOREM). Let $\mathcal{D} = (\mathcal{I}, \mathcal{B})$ be a basket database, let K be a subset of $\mathcal{I}$, and let $\mathcal{L}$ be a set of subsets of $\mathcal{I}$. Then,

$$D^{\mathcal{L}}_{\mathtt{supp}}(K) = \sum_{J\in\mathbf{L}(K,\mathcal{L})} \mathtt{dens}(J). \quad (6)$$

Observe that, since dens is a nonnegative function, D is also a nonnegative function.

EXAMPLE 2.6. Reconsider Example 2.3. Then,

$$\begin{aligned}
\mathcal{W}(\{B, CD\}) &= \{BC, BD, BCD\}\\
\mathbf{L}(A, \{B, CD\}) &= [A, \overline{BC}] \cup [A, \overline{BD}] \cup [A, \overline{BCD}]\\
&= [A, AD] \cup [A, AC] \cup [A]\\
&= \{A, AC, AD\}\\
D^{\{B,CD\}}_{\mathtt{supp}}(A) &= \mathtt{dens}(A) + \mathtt{dens}(AC) + \mathtt{dens}(AD).
\end{aligned}$$

3. SUPPORT BOUNDING THEOREMS

We use the results obtained in Section 2 to obtain various support bounding theorems. In particular, we are concerned with obtaining lower and upper bounds on $\mathtt{supp}(I)$ for some item-set $I \subseteq \mathcal{I}$ in terms of the support values of I's subsets. After stating a general bounding theorem, we will derive a bounding theorem which gives the best lower and upper bounds on $\mathtt{supp}(I)$. This bounding theorem was first formulated by Calders [5].

3.1 The support bounding theorem

For a given item-set $I \subseteq \mathcal{I}$, we will consider the set of all $(K, \mathcal{L})$ pairs such that $I = K \cup \bigcup_{L\in\mathcal{L}} L$. For each such pair, we will derive a lower (upper) bound on $\mathtt{supp}(I)$ when $|\mathcal{L}|$ is even (odd, respectively). To state these bounds, we first define the following sets:

DEFINITION 3.1. *Let $I \subseteq \mathcal{I}$. The sets* $\mathtt{Pairs}(I)$, $\mathtt{EvenPairs}(I)$, $\mathtt{OddPairs}(I)$, *as follows:*

$$\begin{aligned}
\mathtt{Pairs}(I) &= \{(K, \mathcal{L}) \mid I = K \cup \bigcup_{L\in\mathcal{L}} L\}\\
\mathtt{EvenPairs}(I) &= \{(K, \mathcal{L}) \in \mathtt{Pairs}(I) \mid |\mathcal{L}| \text{ is even}\}\\
\mathtt{OddPairs}(I) &= \{(K, \mathcal{L}) \in \mathtt{Pairs}(I) \mid |\mathcal{L}| \text{ is odd}\}
\end{aligned}$$

For a pair $(K, \mathcal{L}) \in \mathtt{Pairs}(I)$, we have, by Definition 2.2 and Theorem 2.5 that

$$(-1)^{|\mathcal{L}|}\left[\mathtt{supp}(I) - \sum_{\mathcal{J}\subset\mathcal{L}} (-1)^{|\mathcal{L}|-|\mathcal{J}|-1}\mathtt{supp}(K \cup \bigcup_{J\in\mathcal{J}} J)\right] = \sum_{J\in\mathbf{L}(K,\mathcal{L})} \mathtt{dens}(J). \quad (7)$$

Since dens is a nonnegative function, Eqn. 7 implies that following inequality:

$$(-1)^{|\mathcal{L}|}\left[\mathtt{supp}(I) - \sum_{\mathcal{J}\subset\mathcal{L}} (-1)^{|\mathcal{L}|-|\mathcal{J}|-1}\mathtt{supp}(K \cup \bigcup_{J\in\mathcal{J}} J)\right] \geq 0. \quad (8)$$

Inequality 8, implies upper and lower bounds on $\mathtt{supp}(I)$.

DEFINITION 3.2. *Let $\mathcal{D} = (\mathcal{I}, \mathcal{B})$ be a basket database. The* bounding function B *associated with $\mathcal{D}$ is,*

$$B(K, \mathcal{L}) = \sum_{\mathcal{J}\subset\mathcal{L}} (-1)^{|\mathcal{L}|-|\mathcal{J}|-1}\mathtt{supp}(K \cup \bigcup_{J\in\mathcal{J}} J).$$

By Eqn. 7, we can reformulate $B(\mathcal{L}, K)$ as follows:

$$B(K, \mathcal{L}) = \mathtt{supp}(I) - \sum_{J\in\mathbf{L}(K,\mathcal{L})} \mathtt{dens}(J) \quad |\mathcal{L}| \text{ is even} \quad (9)$$

$$= \mathtt{supp}(I) + \sum_{J\in\mathbf{L}(K,\mathcal{L})} \mathtt{dens}(J) \quad |\mathcal{L}| \text{ is odd.} \quad (10)$$

Thus, $B(\mathcal{L}, K)$ is a lower (upper) bound on $\mathtt{supp}(I)$ when $|\mathcal{L}|$ is even (odd, respectively). The error in the bound is given by the summation over the density function in the above equations. Since the summation is nonnegative we arrive at the following bounding theorem.

THEOREM 3.3 (SUPPORT BOUNDING THEOREM). Let $\mathcal{D} = (\mathcal{I}, \mathcal{B})$ be a basket database and let I be a subset of $\mathcal{I}$. Then, for each $(K, \mathcal{L}) \in \mathtt{Pairs}(I)$,

$$\begin{aligned}
B(K, \mathcal{L}) &\leq \mathtt{supp}(I) \quad \text{if } (K, \mathcal{L}) \in \mathtt{EvenPairs}(I)\\
B(K, \mathcal{L}) &\geq \mathtt{supp}(I) \quad \text{if } (K, \mathcal{L}) \in \mathtt{OddPairs}(I)
\end{aligned}$$

3.2 The best support bounds theorem

We now consider the problem of finding the best bounds on $\mathtt{supp}(I)$ derivable from $\mathtt{Pairs}(I)$. In fact, we will show that the best bounds come from pairs of the form $(K, \{\{l\} \mid l \in I - K\})$, which we will denote as $(K, \mathbf{I} - \mathbf{K})$.

By Theorem 3.3, for a pair $(K, \mathcal{L}) \in \mathtt{Pairs}(I)$, $B(K, \mathcal{L})$ is either a lower or an upper bound on $\mathtt{supp}(I)$. Some of these pairs,

however, lead to trivial bounds and we will eliminate these from further consideration. A pair $(K, \mathcal{L})$ is *trivial* if there exists an $L \in \mathcal{L}$ such that $L \subseteq K$. For such a pair, it is easy to verify that $B(K, \mathcal{L}) = \mathtt{supp}(I)$. It will be useful to introduce the following subsets of $\mathtt{Pairs}(I)$.

DEFINITION 3.4. *Let $I \subseteq \mathcal{I}$. Then,*

$$\begin{aligned}
\mathtt{NonTrivPairs}(I) &= \{(K, \mathcal{L}) \in \mathtt{Pairs} \mid \forall L \in \mathcal{L} : L \not\subseteq K\} \\
\mathtt{NonTrivEvenPairs}(I) &= \mathtt{NonTrivPairs}(I) \cap \mathtt{EvenPairs}(I) \\
\mathtt{NonTrivOddPairs}(I) &= \mathtt{NonTrivPairs}(I) \cap \mathtt{OddPairs}(I)
\end{aligned}$$

$$\begin{aligned}
\mathtt{AtomicPairs}(I) &= \{(K, \mathbf{I} - \mathbf{K}) \mid K \subseteq I\} \\
\mathtt{AtomicEvenPairs}(I) &= \mathtt{AtomicPairs}(I) \cap \mathtt{EvenPairs}(I) \\
\mathtt{AtomicOddPairs}(I) &= \mathtt{AtomicPairs}(I) \cap \mathtt{OddPairs}(I)
\end{aligned}$$

THEOREM 3.5. Let $\mathcal{D} = (\mathcal{I}, \mathcal{B})$ be a basket database and let I be a $\mathcal{I}$. Then,

(1) $\max(\{B(K, \mathcal{L}) \mid (K, \mathcal{L}) \in \mathtt{NonTrivEvenPairs}(I)\}) = \max(\{B(K, \mathcal{L}) \mid (K, \mathcal{L}) \in \mathtt{AtomicEvenPairs}(I)\})$

(2) $\min(\{B(K, \mathcal{L}) \mid (K, \mathcal{L}) \in \mathtt{NonTrivOddPairs}(I)\}) = \min(\{B(K, \mathcal{L}) \mid (K, \mathcal{L}) \in \mathtt{AtomicOddPairs}(I)\})$

Proof: We will establish (1). (The case for (2) is analogous.) Let $K \subseteq I$ with $|I-K|$ even and let $(K, \mathcal{L}) \in \mathtt{NonTrivEvenPairs}(I)$. We will show that $B(K, \mathcal{L}) \leq B(K, \mathbf{I} - \mathbf{K})$. By Eqn. 9, this is equivalent to showing that

$$\sum_{\mathbf{L}(K, \mathcal{L})} \mathtt{dens}(U) \geq \sum_{\mathbf{L}(K, \mathbf{I}-\mathbf{K})} \mathtt{dens}(U).$$

This inequality is true when $\mathbf{L}(K, \mathbf{I} - \mathbf{K}) \subseteq \mathbf{L}(K, \mathcal{L})$, or equivalently, when $\mathcal{W}(\mathbf{I} - \mathbf{K}) \subseteq \mathcal{W}(\mathcal{L})$. From the definition of witness sets (Definition 2.4), it follows that $\mathcal{W}(\mathbf{I} - \mathbf{K}) = \{I - K\}$. Thus, all we need to show is that $I - K \in \mathcal{W}(\mathcal{L})$. In particular, we must show that (1) $I - K \subseteq \bigcup_{L \in \mathcal{L}} L$, and (2) $\forall L \in \mathcal{L} : L \cap (I - K) \neq \emptyset$. To show condition (1), assume that there exists an $l \in I - K$ such that $l \notin \bigcup_{L \in \mathcal{L}} L$. But, since $K \cup \bigcup_{L \in \mathcal{L}} L = I$, l must be in K, but that contradicts $l \in I - K$. To show condition (2), assume that there exists an $L \in \mathcal{L}$ such that $L \cap (I - K) = \emptyset$. But this implies $L \subseteq K$, a contradiction since $(K, \mathcal{L})$ is a nontrivial pair.

We can now use Theorem 3.5 to establish the following theorem which states how to derive the best bounds:

THEOREM 3.6 (BEST SUPPORT BOUNDS THEOREM). Let $\mathcal{D} = (\mathcal{I}, \mathcal{B})$ be a basket database and let I be a subset of $\mathcal{I}$. Then,

$$\begin{aligned}
\max(\{B(K, \mathcal{L}) \mid (K, \mathcal{L}) \in \mathtt{AtomicEvenPairs}(I)\}) &\leq \mathtt{supp}(I) \\
\min(\{B(K, \mathcal{L}) \mid (K, \mathcal{L}) \in \mathtt{AtomicOddPairs}(I)\}) &\geq \mathtt{supp}(I)
\end{aligned}$$

Calders [4] was the first to prove the Best Support Bounds Theorem (Theorem 3.6) which, we have shown to be a consequence of the Support Bounding Theorem (Theorem 3.3). He applied the Best Support Bounds Theorem to the problem of mining non-derivable item-sets [4]. Specifically, Calders was interested in item-sets I, where the lower bound on $\mathtt{supp}(I)$ is *equal* to the upper bound on $\mathtt{supp}(I)$. In that case $\mathtt{supp}(I)$ need not be computed by counting it in the basket database. The effectiveness of this technique in the context of mining frequent item-sets was empirically observed in [6].

An alternative way to state the Best Support Bounds Theorem (Theorem 3.6) is in terms of the cardinality of $I - K$. In particular, the best lower (upper) bound on $\mathtt{supp}(I)$ is

$$\max(\{B(K, \mathbf{I} - \mathbf{K}) \mid q = |I - K| \,\&\, q \text{ is even}\}) \quad (11)$$

$$\min(\{B(K, \mathbf{I} - \mathbf{K}) \mid q = |I - K| \,\&\, q \text{ is odd}\}). \quad (12)$$

These bounds lead to the notion of q-rules, where $q \in [0, |I|]$. The q-rule for I is the following statement, relating $\mathtt{supp}(I)$ with the best bounds that are obtainable from sets $K \subseteq I$ for which $|I - K| = q$:

$$\mathtt{supp}(I) \geq B(K, \mathbf{I} - \mathbf{K}) \qquad q \text{ is even} \quad (13)$$

$$\mathtt{supp}(I) \leq B(K, \mathbf{I} - \mathbf{K}) \qquad q \text{ is odd.} \quad (14)$$

For example, the 0-rule and the 2-rule state the following (let i_1 and i_2 be two distinct elements in I):

$q = 0$, $K = I$

$$\mathtt{supp}(I) \geq 0.$$

$q = 2$, $K = I - \{i_1, i_2\}$

$$\mathtt{supp}(I) \geq \mathtt{supp}(K \cup \{i_1\}) + \mathtt{supp}(K \cup \{i_2\}) - \mathtt{supp}(K).$$

And, the 1-rule and the 3-rule state the following (let i_1, i_2 and i_3 be distinct elements in I):

$q = 1$, $K = I - \{i_1\}$

$$\mathtt{supp}(I) \leq \mathtt{supp}(K).$$

$q = 3$, $K = I - \{i_1, i_2, i_3\}$

$$\begin{aligned}
\mathtt{supp}(I) \leq\ & \mathtt{supp}(K \cup \{i_1, i_2\}) + \mathtt{supp}(K \cup \{i_1, i_3\}) \\
& + \mathtt{supp}(K \cup \{i_2, i_3\}) - \mathtt{supp}(K \cup \{i_1\}) \\
& - \mathtt{supp}(K \cup \{i_2\}) - \mathtt{supp}(K \cup \{i_3\}) + \mathtt{supp}(K).
\end{aligned}$$

The q-rules highlight two issues that are relevant in the computation of the best bounds on $\mathtt{supp}(I)$. The first is that if we just consider sets $K \subseteq I$ such that $|I - K|$ is fixed and even (odd), what is an efficient way to compute the best bound obtainable on $\mathtt{supp}(I)$ from these K's? The second is when we have K and K' such that both $|I - K|$ and $|I - K'|$ are even (odd) but of different sizes, which of them leads to the best bound on $\mathtt{supp}(I)$? The first issue will be addressed in the next subsection. The second issue is more advantageously addressed by analyzing it for basket databases that satisfy certain conditions (such as independence, see Section 4). This is because the relative effectiveness of the rules is determined by the comparing the sum $\sum_{K \subseteq J \subseteq \overline{I-K}} \mathtt{dens}(J)$ to the sum $\sum_{K' \subseteq J \subseteq \overline{I-K'}} \mathtt{dens}(J)$. Since the values of these sums are expressed in terms of the density function of the basket database, their comparison is entirely controlled by the properties of the data. Thus, to gain a better understanding of this issue, we need to make certain assumptions on this data. This will be addressed in Section 4.

3.3 Computing and approximating best support bounds

From the Best Support Bounds Theorem (Theorem 3.5), it follows that the best lower bound on $\texttt{supp}(I)$ is given by $\max(\{B(K,\mathcal{L}) \mid (K,\mathcal{L}) \in \texttt{AtomicEvenPairs}(I)\})$, and the best upper bounds is given by $\min(\{B(K,\mathcal{L}) \mid (K,\mathcal{L}) \in \texttt{AtomicOddPairs}(I)\})$. Determining these bounds can be computationally expensive. A naive complexity analysis gives that this can be done in PSPACE. However, we can consider heuristics that would approximate the best bounds, but that may lead to a more efficient search for good (but not necessarily optimum) candidates K for obtaining bounds on $\texttt{supp}(I)$. The following proposition is insightful in this regard.

PROPOSITION 3.7. Let $\mathcal{D} = (\mathcal{I}, \mathcal{B})$ be a basket database and let I be a subset of $\mathcal{I}$. Then, determining the best lower (upper) bound on $\texttt{supp}(I)$ can be done by minimizing the sum

$$\sum_{K \subseteq J \subseteq \overline{I-K}} \texttt{dens}(J)$$

over all subsets K of I.

Proof: By Theorem 3.5, the best lower (upper) bound on $\texttt{supp}(I)$ is obtained by maximizing (minimizing) $B(K, \{\{l\} \mid l \in I-K\})$. By Eqn.9 (Eqn.10), this is equivalent to minimizing $\sum_{K \subseteq J \subseteq \overline{I-K}} \texttt{dens}(J)$ over all $K \subseteq I$.

The significance of Proposition 3.7 is that it allows us to determine the error of the best bounds in terms of minimizing a sum over values of the density function. Since, as stated above, this minimization is combinatorially expensive, it is useful to find good approximations for these sums. In particular, we are interested in heuristically determining a $K \subseteq I$ that would be a good candidate for computing the best bounds on $\texttt{supp}(I)$. In this regard, we can use the following inequality:

$$\texttt{supp}(K) = \sum_{K \subseteq J \subseteq \mathcal{I}} \texttt{dens}(J) \geq \sum_{K \subseteq J \subseteq \overline{I-K}} \texttt{dens}(J), \quad (15)$$

and then use the heuristic that K can be selected by finding the smallest value for $\texttt{supp}(K)$, and then use that K to compute a hopefully good bound on $\texttt{supp}(I)$. (In Section 5, we introduce the heuristic called Hq based on these ideas.)

Another technique to more efficiently compute an approximation of the best bounds is to use a specific q-rule. In that case, only K's such that $|I - K| = q$ need to be considered. Thus rather than considering all possible $2^{|I|}$ K sets, only $\binom{|I|}{q}$ (i.e. a polynomial of degree q in $|I|$) K sets need to be considered. (In Section 5, we consider applying this technique for $q = 1$, 2, and 3 and we label these cases by R1, R2, and R3, respectively.)

4. ANALYSIS FOR BASKET DATABASES WITH INDEPENDENCIES

We will now study the relative effectiveness and efficiency of computing lower and upper bounds, as stated in of Theorem 3.6, for basket databases that satisfy a condition of independence. Specially, we say that a basket database $\mathcal{D} = (\mathcal{I}, \mathcal{B})$ satisfies the *independence property* if for each K and L subsets of $\mathcal{I}$,

$$\texttt{supp}(K \cup L)\texttt{supp}(K \cap L) \quad = \quad \texttt{supp}(K)\texttt{supp}(L), \quad (16)$$

or, equivalently, by the definition of freq,

$$\texttt{freq}(K \cup L)\texttt{freq}(K \cap L) \quad = \quad \texttt{freq}(K)\texttt{freq}(L). \quad (17)$$

In the rest of the paper, we will focus on freq rather than supp because certain expressions are more transparent in terms of freq. We use $\texttt{freq}(i)$ as a shorthand for $\texttt{freq}(\{i\})$. With this notation, the independence property can be shown to be equivalent to the following statement: for each $I \subseteq \mathcal{I}$,

$$\texttt{freq}(I) \quad = \quad \prod_{i \in I} \texttt{freq}(i) \quad (18)$$

Using the frequency function we can represent the ideas from Section 3 as follows.

DEFINITION 4.1. *Let $\mathcal{D} = (\mathcal{I}, \mathcal{B})$ be a basket database. For each pair $(K, \mathcal{L})$, the* $(K, \mathcal{L})$-bound*, $B_{\texttt{freq}}(K, \mathcal{L})$, on $\texttt{freq}(I)$ is defined such that $B_{\texttt{freq}}(K, \mathcal{L}) = \sum_{\mathcal{J} \subset \mathcal{L}} (-1)^{|\mathcal{L}|-|\mathcal{J}|-1} \texttt{freq}(K \cup \bigcup_{J \in \mathcal{J}} J)$.*

THEOREM 4.2. Let $\mathcal{D} = (\mathcal{I}, \mathcal{B})$ be a basket database and let I be a subset of $\mathcal{I}$. Then,

$$\begin{aligned} \max(\{B_{\texttt{freq}}(K,\mathcal{L}) \mid (K,\mathcal{L}) \in \texttt{AtomicEvenPairs}(I)\}) &\leq \texttt{freq}(I) \\ \min(\{B_{\texttt{freq}}(K,\mathcal{L}) \mid (K,\mathcal{L}) \in \texttt{AtomicOddPairs}(I)\}) &\geq \texttt{freq}(I) \end{aligned}$$

Furthermore, in analogy with Proposition 3.7, determining the best lower (upper) bound on $\texttt{freq}(I)$ can be done by minimizing the sum $\sum_{K \subseteq J \subseteq \overline{I-K}} \texttt{dens}(J)$ over all subsets K of I.

In the case where the basket database $\mathcal{D} = (\mathcal{I}, \mathcal{B})$ satisfies the independence condition, the bound $B_{\texttt{freq}}(K, \mathbf{I} - \mathbf{K})$ is such that,

$$B_{\texttt{freq}}(K, \mathbf{I} - \mathbf{K}) = \texttt{freq}(K) \sum_{J \subset I-K} (-1)^{|I-K|-|J|-1} \texttt{freq}(J) \quad (19)$$

In the following subsections, we present an analysis for determining lower an upper bounds on $\texttt{freq}(I)$ for basket databases that satisfy the independence condition. In Subsection 4.1, we study the relative effectiveness of different q-rules on the quality of the bounds. In Subsection 4.2, within the context of a fixed q-rule, we examine different heuristics to efficiently determine a K such that $B_{\texttt{freq}}(K, \mathbf{I} - \mathbf{K})$ is the best possible bound.

4.1 Inter-analysis between different q-rules

From Eqn. 13 and Eqn. 14, we learned that we can enumerate different lower bounds and upper bounds on $\texttt{freq}(I)$ depending on the cardinality q of $I - K$. We are interested in studying the relative effectiveness of different q rules to obtain good bounds. To that end, we consider a comparison between a q-rule and a $q + 2$-rule where q is even (odd). Specifically, let K be such that $|I - K| = q$ and let i_1 and i_2 be two different elements of K, and let K' denote the set $K - \{i_1, i_2\}$. Thus $I - K' = (I - K) \cup \{i_1, i_2\}$ and therefore $|I - K'| = q + 2$. By Eqn. 19, comparing $B_{\texttt{freq}}(K, \mathbf{I} - \mathbf{K})$

and $B_{\texttt{freq}}(K', \mathbf{I} - \mathbf{K'})$ leads to the comparison:

$$\texttt{freq}(K)(-1)^{|I-K|-1} \sum_{J \subset I-K} (-1)^{|J|}\texttt{freq}(J) \quad vs \quad \texttt{freq}(K')(-1)^{|I-K'|-1} \sum_{J' \subset I-K'} (-1)^{|J'|}\texttt{freq}(J')$$

After some algebraic simplification, this is equivalent to,

$$(-1)^{|I-K|-1}\texttt{freq}(K)\left[\prod_{j \in I-K}(1-\texttt{freq}(j))\right] \quad vs \quad (-1)^{|I-K|-1}\texttt{freq}(K')\left[\prod_{j \in (I-K)\cup i_2 i_3}(1-\texttt{freq}(j))\right],$$

which, after further algebraic reductions, yields the following comparison:

$$(-1)^{|I-K|-1}\texttt{freq}(K')\left[\prod_{j \in I-K}(1-\texttt{freq}(j))\right] [\texttt{freq}(i_1)+\texttt{freq}(i_2)-1] \quad vs \quad 0. \quad (20)$$

This shows that for q even (q odd), the bound obtained using the $q+2$-rule will be better than the bound obtained using the q-rule if and only $\texttt{freq}(i_1)+\texttt{freq}(i_2) > 1$.

4.2 Intra-analysis within a fixed q-rule

Within a context of a fixed q-rule, we now consider the problem of determining a K such that $B_{\texttt{freq}}(K, \mathbf{I}-\mathbf{K})$ is the best possible bound for that q. To that end, let X be a subset of I, and let i and i' be two different elements in $I - X$. Let $K = X \cup \{i\}$ and let $K' = X \cup \{i'\}$. Furthermore, assume that $|I - K| = q$ (clearly, therefore $|I - K'| = q$). We are interested in comparing $B_{\texttt{freq}}(K, \mathbf{I}-\mathbf{K})$ and $B_{\texttt{freq}}(K', \mathbf{I}-\mathbf{K}')$. By Eqn. 19, this leads to the following comparison:

$$(-1)^{|I-K|-1}\texttt{freq}(X)\texttt{freq}(i)\left[\prod_{j \in I-K}(1-\texttt{freq}(j))\right] \quad vs \quad (-1)^{|I-K'|-1}\texttt{freq}(X)\texttt{freq}(i')\left[\prod_{j \in I-K'}(1-\texttt{freq}(j))\right],$$

which, after algebraic simplification is equivalent to the following comparison:

$$(-1)^{|I-K|-1}\texttt{freq}(X)\left[\prod_{j \in I-(K\cup K')}(1-\texttt{freq}(j))\right] \left[\texttt{freq}(i)-\texttt{freq}(i')\right] \quad vs \quad 0. \quad (21)$$

Assuming that $\texttt{freq}(X)\left[\prod_{j \in I-(K\cup K')}(1-\texttt{freq}(j))\right] \neq 0$, the above comparison shows that, for q even (q odd) $B_{\texttt{freq}}(K, \mathbf{I}-\mathbf{K})$ will be the better lower bound (upper bound) if and only if $\texttt{freq}(i) \leq \texttt{freq}(i')$.

Using this fact inductively implies that, given I, a $K \subseteq I$ with $|I - K| = q$, which leads to the best q-bound can be obtained by letting K consists of those elements of I from which have been removed the q elements of I with the highest frequencies. For example, if $I = \{i_1, i_2, i_3, i_4\}$ and $\texttt{freq}(i_1) \leq \texttt{freq}(i_2) \leq \texttt{freq}(i_3) \leq \texttt{freq}(i_4)$, and $q = 1$, the best K will be the set $I - \{i_4\} = \{i_1, i_2, i_3\}$. When $q = 2$, $K = \{i_1, i_2\}$, when $q = 3$, $K = \{i_1\}$, and when $q = 4$, $K = \emptyset$.

Combining the inter-analysis and intra-analysis results yields the following proposition:

PROPOSITION 4.3. Let $\mathcal{D} = (\mathcal{I}, \mathcal{B})$ be a basket database and let $I = \{i_1, \ldots, i_n\}$ ($n \geq 1$) be a subset of $\mathcal{I}$. Furthermore, without loss of generality, assume that $\texttt{freq}(i_1) \leq \cdots \leq \texttt{freq}(i_n)$, with $n \geq 2$. If $\mathcal{D}$ satisfies the independence property, then a K that leads to the best lower (upper) bound on $\texttt{freq}(I)$ consists of those elements of I from which have been removed the q elements of I with the highest frequencies, where q is the largest even (odd) value in $[0, n-1]$ at which the condition $\texttt{freq}(i_{n-q})+\texttt{freq}(i_{n-q-1}) \geq 1$ holds. Furthermore, this search for a best K can be done in $O(|I|)$, provided that the frequencies on the items in I are known.

REMARK 4.4.

1. As should be clear from the formulas used in the previous analysis, requiring the independence property on $\mathcal{D}$ is not necessary. For example, for Proposition 4.3 to hold at a particular I, it is sufficient that the independence property holds locally at I. In addition, examining the previous analysis in more detail shows that the results still hold for situations where *near* independence holds. We are currently working on formally determining precise definitions of "near" which are sufficient to determine these results.

2. If there exists no q that satisfies the condition $\texttt{freq}(i_{n-q}) + \texttt{freq}(i_{n-q-1}) \geq 1$, then the 0-rule, which yield the bounds 0, and the 1-rule (i.e., the Apriori rule) give the best lower and upper bound on $\texttt{supp}(I)$.

3. Notice that determining the best q does *not* require computing the bound values. This is in contrast with methods that explicitly need the computation of the bound values to determine the best bounds.

5. ALGORITHMIC IMPACT

We will now investigate methods of applying the theoretical results about bounding the FIM problem. We propose using heuristics in the FIM problem in two places. The first place is to use different q-rules as heuristics in pruning the search space of the FIM problem. We will call these *FIM heuristics*. The second place is, in the context of a fixed q-rule, to use heuristics that can efficiently compute good candidates for K that lead to bounds that well-approximate or equal the best bounds for the q-rule. We call these *bounding rules heuristics*.

FIM heuristics

FIM heuristics have been used in many FIM algorithms, such as Apriori, FP-growth and Eclat [2, 9, 15]. In these algorithms, the FIM heuristics that is used is the 1-rule to bound $\texttt{supp}(I)$ in terms of the supports of its subsets of size $|I| - 1$. Obviously, if one of these supports has been found to fall below threshold, then I is an infrequent item-set and can be pruned from the search space. However, if all that is determined is that the supports of some or all of these subsets are above threshold, then the 1-rule proposes to determine the $\texttt{supp}(I)$ by counting, and depending on its value to determine whether it is frequent or not. We propose using other q-rules (q odd) to possibly compute better bounds to further reduce

Function: R3
Input: Item-Set I.
Output: Return true if I needs to be counted.

1) For each $|K| = |I| - 3$
2) Get support of J where $K \subseteq J \subset I$.
3) If J is infrequent return false.
4) If $\texttt{upperbound}(K, \mathbf{I} - \mathbf{K}) > threshold$
5) return true
7) Else
8) return false.

Function: O3
Input: Item-Set I.
Output: Return true if I needs to be counted.

1) Set K equal to I without is 3 most frequent items
2) Get support of J where $K \subseteq J \subset I$.
3) If J is infrequent return false.
4) If $\texttt{upperbound}(K, \mathbf{I} - \mathbf{K}) > threshold$
5) return true
6) Else
7) return false.

Figure 1: Pseudo-code of the functions R3 and O3 test_candidate Function

the search space in FIM problems. Furthermore, when, in the computation of such bounds, it is discovered (by counting) that a subset of I is infrequent, then we stop and do not further compute the bound. Thus the algorithm harnesses both the monotonicity property, and the value of the bounds against the threshold. Using this breakout technique the q-rules will always do as well as the 1-rule.

Bounding rules heuristics

Given a fixed q, the best bound on $\texttt{supp}(I)$ is obtained by selecting a K that minimizes $\sum_{K \subseteq J \subseteq \overline{I-K}} d(J)$ (see Proposition 3.7). The crude way of doing that is by computing the bound for every possible K (of fixed size) and selecting the best bound. We describe this method below.

Rq The exhaustive way to determine K is to search through $\binom{|I|}{q}$ sets and for each of these sets, compute the bound $B(K, \mathbf{I} - \mathbf{K})$ is computed and tested against the support threshold. We label this approach Rq (e.g. R1 and R3 when $q = 1$ and $q = 3$ respectively). This will involve obtaining the support of the 2^{q-1} subsets of I. Thus the total cost of Rq is no more than $\binom{|I|}{q} 2^{q-1}$.

In the quest for a set K of a fixed size so that the bound computed is the best among the bounds computed using other K sets of the same size, we propose two different heuristics.

Hq We can reduce the cost needed for finding K such that $\sum_{K \subseteq J \subseteq \overline{I-K}} d(J)$ is minimized if we use Eqn. (15) and approximate that sum with $\texttt{supp}(K)$. Thus, this approach will first search through the $\binom{|I|}{q}$ subsets of I and determine a K with minimum support. For such a K, the bound $B(K, \mathbf{I} - \mathbf{K})$ is computed and tested against the support threshold. We label this approach Hq (e.g. H3 for $q = 3$). The total cost of Hq is no more than $\binom{|I|}{q} + 2^{q-1}$.

Oq (Oq*) We develop a heuristic aimed at reducing the cost of Hq. The idea is to assume that $\texttt{freq}(K)$ and $\texttt{supp}(K)$ are minimized when the frequency of the individual items comprising K are minimal. This idea is based on the theoretical results of Section 4. To implement this heuristic, the frequencies of the individual items of an item-set of size $|I|$ are obtained and sorted by frequency. Then K is selected such that the items with the highest q frequencies are filtered out. Once K is selected, one can proceed in the usual way to compute the bound $B(K, \mathbf{I} - \mathbf{K})$ and compare it against the support threshold. We label this approach Oq (e.g. O3 when $q = 3$). The total cost of Oq is no more than $|I| + 2^{q-1}$. Note that unlike Rq neither Hq nor Oq have R1 built in.

Oq* is the same as Oq, except that, when computing the bound on $\texttt{supp}(I)$, the minimum of the R1 bound and the Oq bound is used. Thus Oq* has both the R1 and Oq built in.

6. EXPERIMENTS

We now report on experiments we performed which use the heuristics discussed above. We emphasize that our ideas are not specific to the Apriori algorithm but rather target FIM algorithms in general.

Experimental setup

We used Ferenc Bodon's implementation of the Apriori algorithm [3], and extended it to include the heuristics in two places as discussed in Section 5. For FIM heuristics, we only use the 3-rule enumerated from our bounding theorem instead of the 1-rule. We assume that, in a pre-processing procedure, the frequencies of the items in $\mathcal{I}$ have been computed and sorted. For the bounding rules heuristics, we use H3, O3 and O3* (recall that O3* is the best of R1 and O3). These heuristics take place when generating and testing candidates. Figure 1 shows the test-candidate functions which incorporated the R3 and O3 heuristics, respectively. (The function incorporating H3 is very similar to O3. H3 looks up the supports of every K such that $|K| = |I| - 3$). Notice that only pre-candidates are subjected to these various rules. This means that in all cases the item-sets have been subjected to the most important part of the R1 rule, namely we know that the sets $I - \{i_1\}$, and $I - \{i_2\}$ are frequent (where i_1 and i_2 are the first and second most frequent items). This reduces the possibility of the 3-rule from being effective.

Baskets databases

We ran experiments on chess data, and census data (PUMS) which were provided by Roberto Bayardo from the UCI baskets databases [11]. Finally, we ran our approaches on webdocs - a 1.5GB baskets database donated by Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Fabrizio Silvestri to the FIMI repository [10] and built from a spidered collection of web html documents. Table 2 provides the essential statistics of baskets databases involved. Pumsb$*$ is the same baskets database as pumsb minus all the items with 80% or more frequency.

Experiments were run on a $3.06Ghz$ Intel Xeon system with $4GB$ of memory running RedHat Enterprise Linux. We ran three types of experiments describe below; the results are shown in Tables 3–6 and Appendix A. In all these experiments, we do not include

Table 2: Statistics of the baskets databases sizes

Baskets Database	Baskets	Avg. Basket Length	Distinct Items
chess	3,196	37	76
pumsb	49,046	74	7117
pumsb∗	49,046	50	7117
webdocs	1,692,082	177	5,267,657

Table 3: chess results at 50% threshold

Set		Freq	Exp1: Wasted effort				Exp2: Derivable items		
size	Pre-cand.	item-sets	R1	O3	O3*	R3	R1=O2	O3=O2	O3*=O2
3	4,504	4,000	503	69	69	69	366	1,560	1,729
4	19,847	18,565	1,198	55	55	11	3,442	11,741	12,029
5	60,960	58,172	2,408	54	48	0	18,089	45,576	46,408
6	134,191	129,952	3,468	34	29	0	57,633	115,407	116,670
7	218,411	214,297	3,347	18	17	0	122,979	204,138	205,167

Table 4: PUMSB results at 80% threshold

Set		Freq	Exp1: Wasted effort				Exp2: Derivable items		
size	Pre-cand.	item-sets	R1	O3	O3*	R3	R1=O2	O3=O2	O3*=O2
3	1,842	1,760	82	3	3	3	252	99	340
4	7,668	6,999	566	8	8	2	906	1,475	1,584
5	20,458	18,215	1,733	8	4	0	2,613	4,834	5,229
6	35,717	31,532	2,873	20	17	0	6,177	10,235	11,659
7	41,290	36,382	2,666	13	1	0	11,016	13,653	16,612

Table 5: PUMSB∗ results at 25% threshold

Set		Freq	Exp1: Wasted effort				Exp2: Derivable items		
size	Pre-cand.	item-sets	R1	O3	O3*	R3	R1=O2	O3=O2	O3*=O2
3	8,817	6,106	2,551	1,051	1,051	1,051	1,172	112	1,231
4	27,436	23,331	829	286	286	119	8,710	6,218	9,771
5	72,715	65,625	432	196	52	5	33,441	30,191	37,115
6	155,887	142,594	280	281	35	0	85,187	79,353	91,110
7	267,801	245,939	171	452	13	0	160,024	147,127	165,838

Table 6: webdocs results at 10% threshold

Set		Freq	Exp1: Wasted effort				Exp2: Derivable items		
size	Pre-cand.	item-sets	R1	O3	O3*	R3	R1=O2	O3=O2	O3*=O2
3	26,125	12,343	13,711	13,711	13,711	13,711	0	0	0
4	40,619	31,857	8,480	8,361	8,345	8,345	0	0	0
5	57,336	52,084	5,011	3,135	3,130	2,919	0	0	0
6	58,560	55,867	2,346	739	739	565	0	50	50
7	40,057	39,299	650	160	160	101	36	787	810
8	17,828	17,710	109	109	18	8	80	2,266	2,280

results for H3 since it produced identical results to O3 except in one instance.

Exp1 (See Tables 3–6.) In this experiment, we use the 3-rule as an FIM heuristic and use it to prune the search space. The idea is for every item-set we compute the upper bound using R3, O3 or O3* instead of the 1-rule, and count the number of candidates that the rules report as possibly frequent, yet, when counting them in the basket database, turn out to be infrequent (i.e. wasted effort).

Exp2 (See Tables 3–6.) We count the number of derivable item-sets [6] that have their lower bound and upper bound equal. We also present below this data, the remaining candidates that need to be counted as a percentage of the pre-candidates. We computed the upper bound on the support of an item-set using R1, O3, O3* and R3 with the lower bound being computed using O2. This experiment is interesting since for the item-sets that have their upper bounds and lower bounds equal, one does not need to count their supports.

Exp3 (See Appendix A.) We modify **Exp1** by using the lower bound rule (O2 heuristic) to further prune the search space. The idea is that, if the lower bound is greater than the threshold, then we know that item-set is frequent without counting. Thus, in this experiment we only report the number of candidates that require counting. These are the item-sets whose lower bound is below the threshold and their upper bound is above the threshold. Note that when implementing this algorithm one may reach a stage where the supports of subsets of an item-set are not available (since the frequency status of these subsets may have been determined without counting) [11]. We do not address this issue.

Interpretation of results

From the data presented in Tables 3–4, it is apparent that the $3-$rule prunes the search space and reduces the number of infrequent sets that need to be counted almost perfectly in **Exp1**. The 1-rule doing a good job in eliminating the infrequent item-sets but the 3-rule is doing an almost perfect job. This is consistent with our theoretical results in Section 4, which predict that the 3-rule will do better on dense baskets databases. The other observation deals with the performance of our proposed heuristics: O3 prunes the search space and reduces wasted effort almost as good as O3* and R3 in **Exp1**. In Tables 3–4 of **Exp2** we find more derivable item-sets when computing the lower and bounds using the O2 and O3 than using O2 and R1. The number of derivable items found using O3 and O3* (in conjunction with the lower bound) is very close to the number of derivable items found using R3 with the lower bound. This demonstrates the performance of our proposed approaches. In Appendix A in **Exp3** we find that the remaining candidates that need to be counted after using O3 (or O3*) and O2 to prune the search space is: (1) noticeably less than those remaining candidates that need to be counted after using only R1 and O2; (2) very close to those remaining candidates that need to be counted after using R3 and O2 (the best).

For the data presented in Tables 5–6 of **Exp1**, computing the upper bound using 3-rule with R3 is still reducing wasted effort more than the 1-rule. However the pruning relative effectiveness of these rules is less as can be clearly seen when comparing O3 to R1 in Table 6. Observe there exists an outlier in our results in Table 5 where O3 does slightly worse than the 1-rule at stage 7 of the algorithm. We do not see this as a major problem since it is not observed in O3*. The decrease in pruning effectiveness of these rules can also be observed in Table 6 of **Exp2**, where all rules are unsuccessful in both pruning wasted effort and finding derivable item-sets in the early stages of the algorithm. More importantly, we observe that the use of R1 and O2 to find derivable item-sets is more effective than using O3 and O2 (Observe this does not occur in O3* or R3 with O2 since both these approaches have R1 built into them). This is consistent with our theoretical results in Section 4, which predict the 1-rule be more effective than the 3-rule on sparse baskets databases. We also observe a similar reduction in pruning effectiveness in Appendix A of **Exp3** where we find the remaining candidates that need to be counted using either of O3, O3* or R3 (in conjunction with O2) very close to those candidates that need to be counted using R1 (in conjunction with O2).

7. SUMMARY

- The computational cost of computing a particular bound is equivalent to finding a K such that $\sum_{K \subseteq J \subseteq \overline{I-K}} d(U)$ is minimized and then computing the bound $\bar{B}(K, \mathbf{I} - \mathbf{K})$. While the cost of computing $B(K, \mathbf{I} - \mathbf{K})$ is the same for each heuristic, the cost of finding a K-set such that the sum over density is minimized can be reduced significantly by using the heuristics we propose in this paper.

- The performance of the approaches suggests that all algorithms produce more or less the same number of false candidates, with O3 typically performing better.

- For dense baskets databases, O3 and O3* produce less false candidates than R1, and thus resulting in better performance. For baskets databases too big to fit in memory, it is desirable to avoid false candidates as much as possible. O3 (or O3*) will perform better since the computational cost of O3 (in memory) will usually outperform the cost of false candidates (since that involves I/O access).

- The heuristics we suggest will work for both lower bounds and upper bounds. This will have a positive impact on performance of algorithms like MAXMINER and AprioriLB [11], and concise representation algorithms such as the NDI-Algorithm [6].

- We recommend that one may just as well run the 3-rule in the Apriori algorithm, at least at level 3. This does not present a significant overhead. The advantages are that for dense baskets databases, it will pay off reducing wasted effort to almost 0. Furthermore, even on sparse baskets databases such as webdocs, a reduction in wasted effort can be expected.

Acknowledgment: We thank Ferenc Bodon for use of his implementation of the Apriori algorithm [3].

8. REFERENCES

[1] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216. ACM Press, 1993.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499, 1994.

[3] Ferenc Bodon. A fast apriori implementation. In *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, 2003.

[4] Toon Calders. *Axiomatization and Deduction Rules for the Frequency of Itemsets*. PhD dissertation- University of Antwerp, 2003.

[5] Toon Calders. Computational complexity of itemset frequency satisfiability. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 143–154, 2004.

[6] Toon Calders and Bart Goethals. Mining all non-derivable frequent itemsets. In *Proceedings European Conference on Principles of Data Mining and Knowledge Discovery*, volume 2431 of *LNCS*, pages 74–85. Springer-Verlag, 2002.

[7] Toon Calders and Jan Paredaens. Axiomatization of frequent sets. In *Proceedings of the international conference on database theory*, pages 204–218, 2001.

[8] Dimitrios Gunopulos, Heikki Mannila, and Sanjeev Saluja. Discovering all most specific sentences by randomized algorithms. In *Proceedings of the 6th International Conference on Database Theory*, pages 215–229. Springer-Verlag, 1997.

[9] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 1–12. ACM Press, 2000.

[10] FIMI Repository. http://fimi.cs.helsinki.fi/data.

[11] Jr. Roberto J. Bayardo. Efficiently mining long patterns from databases. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 85–93. ACM Press, 1998.

[12] Bassem Sayrafi and Dirk Van Gucht. Inference systems derived from additive measures. *Workshop on Causality and Causal Discovery, London, Canada*, 2004.

[13] Bassem Sayrafi and Dirk Van Gucht. Differential constraints. In *Proceedings of the twenty fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM Press, 2005.

[14] Bassem Sayrafi, Dirk Van Gucht, and Marc Gyssens. Measures in databases and datamining. Tech. Report TR602, Indiana University Computer Science, 2004.

[15] Mohammed J. Zaki. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12(3):372–390, 2000.

APPENDIX

A. LOWER BOUND RESULTS

We present some results from combining the $2-$rule (denoted by LB in the tables below) with R1 and O3. Remaining candidates refers to the candidate sets that remain after pruning the search space using the rules noted. It is only for these sets that we are required to visit the database and count their support. Observe in the chess and PUMSB results, that the number of remaining candidates is reduced significantly when computing the lower and upper bounds using O2 and O3 versus O2 and the 1-rule. On the other baskets databases (sparse baskets databases), we observe that the same effect only on a smaller scale.

chess results at 50% support

			Remaining candidates		
Set size	Pre-cand.	Freq	LB + R1	LB + O3	LB+O3*
3	4,504	4,000	596	162	162
4	19,847	18,565	1,388	245	245
5	60,960	58,172	2,716	356	356
6	134,191	129,952	3,763	324	324
7	218,411	214,297	3,539	209	209

PUMSB results at 80% support

			Remaining candidates		
Set size	Pre-cand.	Freq	LB + R1	LB + O3	LB+O3*
3	1,842	1,760	97	18	18
4	7,668	6,999	622	64	64
5	20,458	18,215	1,917	192	188
6	35,717	31,532	3,445	592	589
7	41,290	36,382	3,635	982	970

PUMSB$*$ results at 25% support

			Remaining candidates		
Set size	Pre-cand.	Freq	LB + R1	LB + O3	LB+O3*
3	8,817	6,106	3,967	2,467	2,467
4	27,436	23,331	3,008	2,465	2,465
5	72,715	65,625	4,046	3,810	3,666
6	155,887	142,594	5,941	5,942	5,696
7	267,801	245,939	7,849	8,130	7,691

webdocs results at 10% support

			Remaining candidates		
Set size	Pre-cand.	Freq	LB + R1	LB + O3	LB+O3*
3	26,125	12,343	19,422	19,422	19,422
4	40,619	31,857	12,743	12,624	12,608
5	57,336	52,084	7,177	5,301	5,296
6	58,560	55,867	3,207	1,600	1,600
7	40,057	39,299	858	368	368
8	17,828	17,710	125	34	34