# Support Approximations Using Bonferroni-type Inequalities

Szymon Jaroszewicz and Dan A. Simovici

University of Massachusetts at Boston,
Department of Computer Science,
Boston, Massachusetts 02125, USA
{sj,dsim}@cs.umb.edu

**Abstract.** The purpose of this paper is to examine the usability of Bonferroni-type combinatorial inequalities to estimation of support of itemsets as well as general Boolean expressions. Families of inequalities for various types of Boolean expressions are presented and evaluated experimentally.
**Keywords:** frequent itemsets, query support, Bonferroni inequalities

## 1 Introduction

In [MT96] a question has been raised of estimating supports of general Boolean expressions based on supports of frequent itemsets discovered by a datamining algorithm. The accuracy of this estimation (using the inclusion-exclusion principle) is influenced by the supports of the frequent items set; when, for various reasons, some of these supports are missing this accuracy may be compromised. The problem has been addressed in [LN90,KLS96] but the results presented there can be applied only for the case when we know supports of all itemsets up to a given size. This is usually not the case with datamining algorithms which compute supports of only some of the itemsets of a given size.
A similar problem has been addressed in the area of statistical data protection, where it is important to assure that inferences about individual cases cannot be made from marginal totals (see [Dob01,BG99] for an overview). Those methods concentrate on obtaining the most accurate bounds possible (in order to rule out information disclosure), computational efficiency being a secondary concern. Algorithms usually involve repeated iterations over full contingency tables [BG99], branch and bound search [Dob01] or numerous applications of linear programming.
The approach we take in this paper is based on a family of combinatorial inequalities called Bonferroni inequalities [GS96]. In their original form the inequalities require that we know supports of all itemsets up to a given size. We address the problem by using the inequalities recursively to estimate supports of missing itemsets. The advantage of Bonferroni inequalities is that we can choose an arbitrary limit on the size of the marginals involved, thus allowing for trading off accuracy for speed. Our experiments revealed that it is possible to obtain good bounds even if only marginals of small size are used.

A table is a triple $\tau = (T, H, \rho)$, where $T$ is the name of the table, $H = \{A_1, \ldots, A_n\}$ is the heading of the table and $\rho = \{t_1, \ldots, t_m\}$ is a finite set of functions of the form $t_i : H \longrightarrow \bigcup_{A \in H} \mathrm{Dom}(A)$ such that $t_i(A) \in \mathrm{Dom}(A)$ for every $a \in A$. Following the relational database terminology we shall refer to these functions as $H$-tuples, or simply as tuples. If $\mathrm{Dom}(A_i) = \{0, 1\}$ for $1 \leq i \leq n$, then $\tau$ is a binary table.

Let $\tau = (T, A_1 \cdots A_n, \rho)$ be a binary table.

An *itemset* of $\tau$ is an expression of the form $A_{i_1} \cdots A_{i_k}$. A *minterm* of $\tau$ is an expression of the form $A_{i_1}^{b_1} \cdots A_{i_k}^{b_k}$, where $b_i \in \{0, 1\}$ for $1 \leq i \leq k$ and

$$A^b = \begin{cases} A & \text{if } b = 1 \\ \bar{A} & \text{if } b = 0. \end{cases}$$

The support of a minterm $M = A_{i_1}^{b_1} \cdots A_{i_k}^{b_k}$ of a table $\tau = (T, H, \rho)$

$$\mathrm{supp}(M) = \frac{|\{t \in \rho \,|\, t[A_{i_1} \cdots A_{i_k}] = (b_1, \ldots, b_k)\}|}{|\rho|}.$$

Note that the support of minterms is actually a probability measure on the free Boolean algebra $\mathcal{Q}(H)$ generated by the attributes of the heading $H$. We refer to such polynomials as *queries*. The atoms of this algebra are the minterms and every boolean polynomial over the set $H$ can be uniquely written as a disjunction of minterms. The least and the largest element of this Boolean algebra are denoted by $\emptyset$ and $\Omega$, respectively.

Consider a table whose heading is $H = ABC$ and assume that the distribution of the values of the tuples in this table is given by:

| $A$ | $B$ | $C$ | Frequency |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0.10 |
| 0 | 1 | 1 | 0.25 |
| 1 | 0 | 0 | 0.10 |
| 1 | 0 | 1 | 0.25 |
| 1 | 1 | 0 | 0.05 |
| 1 | 1 | 1 | 0.25 |

A run of the Apriori algorithm ([AMS$^+$96]) on a dataset conforming to that distribution, with the minimum support of 0.35 will yield the following itemsets:

| Itemset | Support |
|---|---|
| $A$ | 0.65 |
| $B$ | 0.65 |
| $C$ | 0.75 |
| $AC$ | 0.50 |
| $BC$ | 0.50 |

To estimate the unknown support of the itemset $ABC$ we can use Bonferroni inequalities of the form:

$$\mathrm{supp}(ABC) \geq 1 - \mathrm{supp}(\bar{A}) - \mathrm{supp}(\bar{B}) - \mathrm{supp}(\bar{C}), \quad (1)$$

$$\begin{aligned} \mathrm{supp}(ABC) \leq\ & 1 - \mathrm{supp}(\bar{A}) - \mathrm{supp}(\bar{B}) - \mathrm{supp}(\bar{C}) \\ & + \mathrm{supp}(\bar{A}\bar{B}) + \mathrm{supp}(\bar{A}\bar{C}) + \mathrm{supp}(\bar{B}\bar{C}). \end{aligned} \quad (2)$$

Note that since the support of $AB$ is below the minimum support its value is not returned by the Apriori algorithm and this creates a problem for this estimation. All the itemset supports, except for $\mathrm{supp}(\bar{A}\bar{B})$, in the previous expression can be determined from known itemset supports using inclusion-exclusion principle. For example, we have

$$\mathrm{supp}(\bar{A}\bar{C}) = 1 - \mathrm{supp}(A) - \mathrm{supp}(C) + \mathrm{supp}(AC) = 0.1.$$

Since all needed probabilities are known exactly, the lower bound (1) is easy to compute giving

$$\mathrm{supp}(ABC) \geq 1 - 0.35 - 0.35 - 0.25 = 0.05.$$

To compute the upper bound we proceed as follows.
Since $\mathrm{supp}(\bar{A}\bar{B})$ is not known, we apply Bonferroni inequalities recursively to get an upper bound for it. We have

$$\mathrm{supp}(\bar{A}\bar{B}) = 1 - \mathrm{supp}(A) - \mathrm{supp}(B) + \mathrm{supp}(AB),$$

and, since $AB$ is not frequent, we know that its support is less than the 0.35 minimum support, giving

$$\mathrm{supp}(\bar{A}\bar{B}) < 1 - \mathrm{supp}(A) - \mathrm{supp}(B) + \mathsf{minsupp} = 0.05.$$

Substituting into (3) we get

$$\begin{aligned} \mathrm{supp}(ABC) &< 1 - \mathrm{supp}(\bar{A}) - \mathrm{supp}(\bar{B}) - \mathrm{supp}(\bar{C}) \\ &\quad + 0.05 + \mathrm{supp}(\bar{A}\bar{C}) + \mathrm{supp}(\bar{B}\bar{C}) \\ &= 1 - 0.35 - 0.35 - 0.25 + 0.05 + 0.1 + 0.1 = 0.3. \end{aligned}$$

Note that both bounds are not trivial since the lower bound is greater than 0, and the upper bound is less than the minimum support.
In the next section we present a systematic method for obtaining bounds for support of an itemset using supports of its subsets.
To obtain certain Bonferroni-type inequalities we use a technique known as the method of indicators [GS96]. For an event $Q$ in the probability space define the random variable $I_Q$ called the indicator of $Q$ as

$$I_Q = \begin{cases} 1 & \text{if } Q \text{ occurs} \\ 0 & \text{otherwise} \end{cases}$$

Then, for the expected value of $I_Q$ we have $E(I_Q) = P(Q)$. For our specific probability space we have $E(I_Q) = \mathrm{supp}(Q)$ for every query $Q$.
Let $Q_1, \ldots, Q_m$ be $m$ queries. Define the random variable $J_k$ as

$$J_k = \sum \{ I_{Q_{i_1}} \cdots I_{Q_{i_k}} \mid 1 \leq i_1 \leq \cdots \leq i_k \leq m \}.$$

Clearly, $J_k$ takes as a value the number of $k$-conjunctions $Q_{i_1} \wedge \cdots \wedge Q_{i_k}$ that are satisfied. If $\nu_m$ is the number of queries that are satisfied among the $m$ queries $Q_1, \ldots, Q_m$, then we clearly have $J_k = \binom{\nu_m}{k}$, which implies

$$E\binom{\nu_m}{k} = E(J_k) = \sum \{\mathrm{supp}(Q_{i_1} \wedge \cdots \wedge Q_{i_k}) \mid 1 \leq i_1 \leq \cdots \leq i_k \leq m\}.$$

See [GS96] for further details.
Using the indicator method we proved in [JSR02] the following general result:

**Theorem 1.** *Let $Q = I_1 \oplus I_2 \oplus \ldots \oplus I_m$ be a boolean query, where $\oplus$ denotes the exclusive or operation, and $I_1, I_2, \ldots, I_m$ are itemsets. The following inequalities hold for any natural number $t$:*

$$\sum_{k=1}^{2t} (-2)^{k-1} \sum_{i_1 < \ldots < i_k} \mathrm{supp}(I_{i_1} \wedge \ldots \wedge I_{i_k}) \leq \mathrm{supp}(I_1 \oplus \ldots \oplus I_m) \leq \sum_{k=1}^{2t+1} (-2)^{k-1} \sum_{i_1 < \ldots < i_k} \mathrm{supp}(I_{i_1} \wedge \ldots \wedge I_{i_k}).$$

Note that since every query can be represented as an exclusive or of positive conjunctions, the above theorem allows us to obtain bounds for any boolean query expressed in terms of supports of positive conjunctions. However, these bounds are not always tight, and in fact, we showed in [JSR02] that for certain queries it is not possible to obtain tight bounds at all.

## 2 A Recursive Procedure for Computing Bonferroni Bounds from Frequent Itemsets

Since the Apriori algorithm only discovers supports of itemsets (as opposed to other types of queries), we need to express all inequalities in terms of supports of itemsets.

**Theorem 2.** *Let $Q_1, \ldots, Q_m$ be $m$ queries in $\mathcal{Q}(H)$. The following inequalities hold for any $t \in \mathbb{N}$:*

$$\sum_{k=0}^{2t+1} (-1)^k \sum_{r < i_1 < \ldots < i_k \leq m} \mathrm{supp}(Q_1 \ldots Q_r Q_{i_1} \ldots Q_{i_k}) \leq \mathrm{supp}(Q_1 \ldots Q_r \bar{Q}_{r+1} \ldots \bar{Q}_m) \leq \sum_{k=0}^{2t} (-1)^k \sum_{r < i_1 < \ldots < i_k \leq m} \mathrm{supp}(Q_1 \ldots Q_r Q_{i_1} \ldots Q_{i_k}).$$

*Proof.* By Rényi's Theorem [Rén58] it suffices to prove the claim for $Q_i \in \{\Omega, \emptyset\}$ for all $1 \leq i \leq m$, where $\Omega$ denotes the space of elementary events. When $Q_i = \emptyset$ for some $1 \leq i \leq r$, then both sides of the inequalities reduce to 0 and the result is immediate. For the case $Q_i = \Omega$ for all $1 \leq i \leq r$ we have $\mathrm{supp}(Q_1 \dots Q_r \bar{Q}_{r+1} \dots \bar{Q}_m) = \mathrm{supp}(\bar{Q}_{r+1} \dots \bar{Q}_m)$, and for all $k$ and for all $r < i_1 < \dots < i_k \leq m$, $\mathrm{supp}(Q_1 \dots Q_r Q_{i_1} \dots Q_{i_k}) = \mathrm{supp}(Q_{i_1} \dots Q_{i_k})$. The result now follows from Bonferroni inequalities. □

**Corollary 1.** *Let $A_1 A_2 \dots A_r \bar{A}_{r+1} \bar{A}_{r+2} \dots \bar{A}_m$ be a minterm. The following inequalities hold for any natural number $t$:*

$$\sum_{k=0}^{2t+1} (-1)^k \sum_{r<i_1<\dots<i_k\leq m} \mathrm{supp}(A_1 \dots A_r A_{i_1} \dots A_{i_k}) \leq \mathrm{supp}(A_1 \dots A_r \bar{A}_{r+1} \dots \bar{A}_m) \leq \sum_{k=0}^{2t} (-1)^k \sum_{r<i_1<\dots<i_k\leq m} \mathrm{supp}(A_1 \dots A_r A_{i_1} \dots A_{i_k})$$

*Proof.* This statement follows immediately from Theorem 2. □

Below we present results which form the basis of our algorithm for approximative computations of supports of itemsets. The binomial symbol $\binom{n}{k}$ will allow negative values of $n$, in which case its value is defined by the usual formula

$$\binom{n}{k} = \frac{n(n-1)\cdots(n-k+1)}{k!}.$$

**Lemma 1.** *For $m, k, h, s \in \mathbb{N}$ we have:*

$$\sum_{k=0}^{s} (-1)^{s-k} \binom{m-k-1}{s-k} \binom{h}{k} = \binom{h-m+s}{s}.$$

*Proof.* We begin by showing that for every $a, b, c, d \in \mathbb{N}$ we have

$$\sum_{k=0}^{a} (-1)^k \binom{a-k}{b} \binom{c}{k-d} = (-1)^{a+b} \binom{c-b-1}{a-b-d}. \tag{3}$$

The proof is by induction on $c$. The basis step, $c = 0$, follows after elementary algebraic transformations. Suppose that the equality holds for numbers less than $c$. We have:

$$\begin{aligned}
&\sum_{k=0}^{a} (-1)^k \binom{a-k}{b} \binom{c}{k-d} \\
&= \sum_{k=0}^{a} (-1)^k \binom{a-k}{b} \binom{c-1}{k-d} + \sum_{k=0}^{a} (-1)^k \binom{a-k}{b} \binom{c-1}{k-d-1} \\
&= (-1)^{a+b} \binom{c-b-2}{a-b-d} + (-1)^{a+b} \binom{c-b-2}{a-b-d-1} \\
&\quad \text{(by the inductive hypothesis)} \\
&= (-1)^{a+b} \binom{c-b-1}{a-b-d}.
\end{aligned}$$

By using the complimentary combinations and Lemma 1 we can write:

$$\sum_{k=0}^{s}(-1)^{s-k}\binom{m-k-1}{s-k}\binom{h}{k} = \sum_{k=0}^{s}(-1)^{s+k}\binom{m-k-1}{m-s-1}\binom{h}{k} =$$

$$(-1)^s \cdot \sum_{k=0}^{s}(-1)^k\binom{m-k-1}{m-s-1}\binom{h}{k} = (-1)^s \cdot (-1)^{2m-2-s}\binom{h-m+s}{s}$$

$$= \binom{h-m+s}{s}.$$

□

Note that if $h = m$, the previous lemma implies

$$\sum_{k=0}^{s}(-1)^{s-k}\binom{m-k-1}{s-k}\binom{m}{k} = 1.$$

Our method of obtaining bounds is based on the following theorem

**Theorem 3.** *The following inequalities hold for any natural number $t$:*

$$\mathrm{supp}(A_1A_2\ldots A_m) \leq \sum_{k=0}^{2t}(-1)^k\binom{m-k-1}{2t-k}S_k \tag{4}$$

$$\mathrm{supp}(A_1A_2\ldots A_m) \geq \sum_{k=0}^{2t+1}(-1)^{k+1}\binom{m-k-1}{2t+1-k}S_k \tag{5}$$

*where*

$$S_k = \sum_{1\leq i_1<\ldots<i_k\leq m}\mathrm{supp}(A_{i_1}\ldots A_{i_k}),$$

*and* $S_0 = 1$.

*Proof.* We use the method of indicators previously discussed.
Let $\nu_m$ be a random variable equal to the number of events $A_1, \ldots, A_m$ that actually occur. By Lemma 1 we have:

$$\sum_{k=0}^{s}(-1)^{s-k}\binom{m-k-1}{s-k}\binom{\nu_m}{k} = \binom{\nu_m-m+s}{s}$$

$$= \begin{cases} 1 & \text{if } \nu_m = m \\ 0 & \text{if } \nu_m < m \text{ and } \nu_m \geq m-s \\ \binom{\nu_m-m+s}{s} & \text{if } \nu_m < m-s. \end{cases}$$

By taking expectations of the above equation we get

$$\sum_{k=0}^{s}(-1)^{s-k}\binom{m-k-1}{s-k}S_k = \mathrm{supp}(\nu_m = m)$$

$$+\sum\left\{\binom{\nu_m(\omega)-m+s}{s}\mathrm{supp}(\omega) : \omega \in \Omega, \nu_m(\omega) < m-s\right\},$$

where $\Omega$ denotes the space of elementary events. Note that when $\nu_m < m-s$ the sign of $\binom{\nu_m-m+s}{s}$ is identical to that of $(-1)^s$. Replacing $s$ by $2t$ or $2t+1$ yields the result. □

## 3 The Estimation Algorithm

The main problem in using Bonferroni-type inequalities on collections of frequent itemsets is that some of the probabilities in the $S_k$ sums are not known. We solved this problem by estimating the missing probabilities using Theorem 3. Given below is an algorithm that computes bounds on support of an itemset based on a collection of itemsets with known supports.

**Algorithm 1.**
**Input:** *Itemset $I$, natural number $r$, collection $\mathcal{F}$ of itemsets, and their supports*
**Output:** *Bounds $L(I), U(I)$ on the support of $I$*
The algorithm is implemented by functions $L$ and $U$ given below

*Function* $L(I, \mathcal{F}, r)$.
1. *If* $I \in \mathcal{F}$
2. *return* $\mathrm{supp}(I)$
3. *else*
4. *return* $\max_{-1\le 2t+1\le r} \sum_{k=0}^{2t+1} S^L\left((-1)^{k+1}\binom{m-k-1}{2t+1-k}, k, I, \mathcal{F}\right)$

*Function* $U(I, \mathcal{F}, r)$.
1. *If* $I \in \mathcal{F}$
2. *return* $\mathrm{supp}(I)$
3. *else*
4. $U \leftarrow \min_{0\le 2t\le r} \sum_{k=0}^{2t} S^U\left((-1)^{k}\binom{m-k-1}{2t-k}, I, k, \mathcal{F}\right)$
5. $U \leftarrow \min\{U, \mathit{minsupp}, \min_{J\subset I} U(J)\}$
6. *return* $U$

The functions $S^L$ and $S^U$ are defined below

*Function* $S^L$*(real coefficient $c$, itemset $I = A_1A_2\dots A_m$, $\mathcal{F}$, integer $k$)*
1. *If* $k = 0$ *return* $c$
2. *If* $c \ge 0$
3. *return* $c \cdot \sum_{i_1<\dots<i_k\le m} L(A_{i_1}A_{i_2}\dots A_{i_k}, \mathcal{F}, k-1)$
4. *else*
5. *return* $c \cdot \sum_{i_1<\dots<i_k\le m} U(A_{i_1}A_{i_2}\dots A_{i_k}, \mathcal{F}, k-1)$

*Function* $S^U$*(real coefficient $c$, itemset $I = A_1A_2\dots A_m$, $\mathcal{F}$, integer $k$)*
1. *If* $k = 0$ *return* $c$
2. *If* $c \ge 0$
3. *return* $c \cdot \sum_{i_1<\dots<i_k\le m} U(A_{i_1}A_{i_2}\dots A_{i_k}, \mathcal{F}, k-1)$
4. *else*
5. *return* $c \cdot \sum_{i_1<\dots<i_k\le m} L(A_{i_1}A_{i_2}\dots A_{i_k}, \mathcal{F}, k-1)$

Of course upper and lower bounds for itemsets are cached during computations to avoid repeated evaluations for the same itemset. The parameter $r$ controls the maximum size of marginals (itemsets) used in the estimation.

The use of minsupp in step 5 of function $U$ requires some comment. Including the value of minsupp in the minimum is possible only if we can determine that the estimated itemset $I$ is not frequent. This can be done for example if $\mathcal{F}$ contains all frequent itemsets, or when $\mathcal{F}$ contains all frequent itemsets up to a given size $k$, and $|I| \leq k$. If we don't know whether $I$ is frequent or not, we have to drop minsupp from the minimum.

## 4 Experimental results

In this section we present experimental evaluation of the bounds. Our algorithm works best on dense datasets, which are more difficult to mine for frequent itemsets than sparse ones. However, the algorithm was tested on both dense and sparse data (artificial market basket data was used). The rest of the paper is focused on experiments performed on dense databases.

As dense databases we used the `mushroom` database from the UCI Machine Learning Archive [BM98], and a census data of elderly people from the University of Massachusetts at Boston Gerontology Center available at `http://www.cs.umb.edu/ ~sj /datasets /census.arff.gz`. Since both datasets involve multivalued attributes, we replaced each attribute (including binary ones) with a number of Boolean attributes, one for each possible value of the original attribute.

Before we present a detailed experimental study of the quality of bounds, we present the results of applying the bounds to a practical task. Suppose that we did not have enough time or computational resources to run the Apriori (or similar) algorithm completely, and we decided to stop the algorithm after finding frequent itemsets of size less than or equal to 2. We then use lower bounds to find frequent itemsets of size greater than 2. The experimental results for `mushroom` and `census` databases are shown in Figures 1 and 2 respectively.

The figures show, for various values of minimum support, the true number of frequent itemsets of sizes 3 and 4, the number of itemsets that we discovered to be frequent by using our bounds, and the ratio of the two numbers.

For large values of minimum support we are more likely to classify an itemset correctly than for smaller ones. The data shows that for itemsets with largest support the chances of actually being determined to be frequent without consulting the data can be as high as 80%.

We now present an experimental analysis of the bounds obtained. In what follows, by *trivial bounds* for the support of an itemset $I$ we mean 0 for the lower bound, and for the upper bound: the minimum of the upper bounds of the supports of all proper subsets of $I$ and of the minimum support. As in the example above here too we mine frequent itemsets with at most two items, and compute bounds for larger ones.

Table 1 (a) contains the results for the `census` dataset with minimum support of 1.8%.

The parameter $r$ in Algorithm 1 was chosen for each itemset $I$ to be $|I| - 1$ for maximum accuracy. This causes an increase in estimation time for larger itemsets. Later in the section we present results showing

| Itemset size | Min. support | 18% | 25% | 30% | 37% | 43% | 49% | 55% | 61% | 73% |
|---|---|---|---|---|---|---|---|---|---|---|
| 3 | Frequent | 1761 | 893 | 498 | 308 | 152 | 70 | 45 | 23 | 13 |
| | Est. Freq. | 345 | 244 | 179 | 127 | 86 | 54 | 34 | 19 | 10 |
| | ratio (%) | 19.59% | 27.32% | 35.94% | 41.23% | 56.58% | 77.14% | 75.56% | 82.61% | 76.92% |
| 4 | Frequent | 4379 | 1769 | 795 | 368 | 147 | 48 | 29 | 16 | 6 |
| | Est. Freq. | 298 | 202 | 131 | 85 | 53 | 31 | 18 | 10 | 2 |
| | ratio (%) | 6.81% | 11.42% | 16.48% | 23.10% | 36.05% | 64.58% | 62.07% | 62.50% | 33.33% |

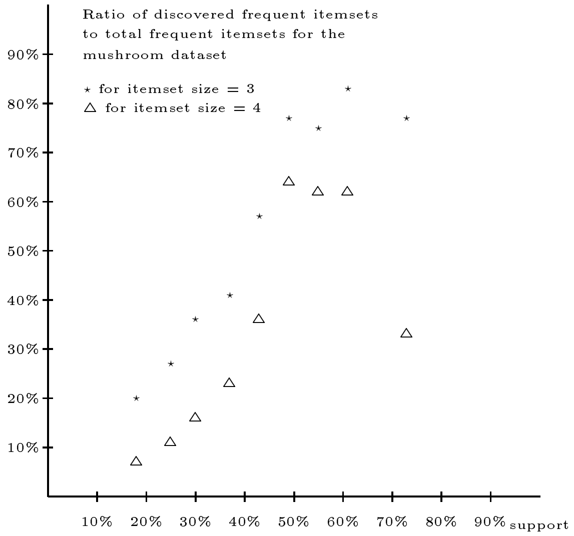


**Fig. 1.** Discovered vs. total frequent itemsets for the `mushroom` dataset

| Itemset size | Min. support | 1% | 2% | 3% | 5% | 10% | 15% | 30% | 50% |
|---|---|---|---|---|---|---|---|---|---|
| 3 | Frequent | 1701 | 1377 | 1145 | 879 | 503 | 312 | 112 | 40 |
| | Est. Freq. | 154 | 149 | 146 | 137 | 108 | 90 | 47 | 21 |
| | ratio (%) | 9.05% | 10.82% | 12.75% | 15.59% | 21.47% | 28.85% | 41.96% | 52.50% |
| 4 | Frequent | 5050 | 3560 | 2728 | 1901 | 852 | 485 | 105 | 20 |
| | Est. Freq. | 103 | 98 | 94 | 85 | 64 | 48 | 18 | 3 |
| | ratio (%) | 2.04% | 2.75% | 3.45% | 4.47% | 7.51% | 9.90% | 17.14% | 15.00% |

**Fig. 2.** Ratios of discovered to total frequent itemsets for the `census` data

| itemset size | 3 | 4 | 5 | 6 |
|---|---|---|---|---|
| average interval width | 0.0482797 | 0.0313103 | 0.0228579 | 0.0196316 |
| average upper bound | 0.0568679 | 0.0319395 | 0.0228771 | 0.0196316 |
| average lower bound | 0.00858817 | 0.000629199 | 1.925e-05 | 0 |
| itemsets with nontrivial bounds | 7.04% | 0.59% | 0.04% | 0.00% |
| itemsets with nontrivial lower | 4.06% | 0.39% | 0.02% | – |
| average lower improvement | 0.211321 | 0.161151 | 0.0962518 | – |
| itemsets with nontrivial upper | 6.43% | 0.47% | 0.03% | – |
| average upper improvement | 0.0225656 | 0.00983444 | 0.00262454 | – |
| time [ms/itemset] | 0.2 | 0.3 | 1 | 7 |

(a) 1.8% minimum support, all itemsets

| itemset size | 3 | 4 | 5 | 6 |
|---|---|---|---|---|
| average interval width | 0.102848 | 0.105024 | 0.106997 | 0.110767 |
| average upper bound | 0.127438 | 0.109572 | 0.107491 | 0.110767 |
| average lower bound | 0.0245896 | 0.00454846 | 0.00049354 | 0 |
| itemsets with nontrivial bounds | 20.17% | 4.25% | 0.58% | 0.02% |
| itemsets with nontrivial lower | 11.64% | 2.82% | 0.46% | – |
| average lower improvement | 0.211321 | 0.161151 | 0.106164 | – |
| itemsets with nontrivial upper | 18.41% | 3.43% | 0.40% | 0.02% |
| average upper improvement | 0.0225656 | 0.00983444 | 0.00333985 | 0.00338427 |

(b) 1.8% minimum support, frequent itemsets only

| itemset size | 3 | 4 | 5 | 6 |
|---|---|---|---|---|
| average interval width | 0.171608 | 0.205194 | 0.222602 | 0.231362 |
| average upper bound | 0.235004 | 0.223174 | 0.225491 | 0.231362 |
| average lower bound | 0.0633963 | 0.0179804 | 0.00288882 | 0 |
| itemsets with nontrivial bounds | 48.55% | 16.79% | 3.40% | 0.14% |
| itemsets with nontrivial lower | 30.00% | 11.16% | 2.72% | – |
| average lower improvement | 0.211321 | 0.161151 | 0.106164 | – |
| itemsets with nontrivial upper | 44.00% | 13.56% | 2.33% | 0.14% |
| average upper improvement | 0.0238776 | 0.00983444 | 0.00333985 | 0.00338427 |

(c) 9% minimum support, frequent itemsets only

**Table 1.** Results for the `census` dataset

that limiting the value of $r$ can give very fast estimates with a very small impact on the quality of the bounds. All experiments were run on a 100MHz Pentium machine with 64MB of memory.

The bounds obtained are fairly accurate. The width of the interval between the lower and upper bounds varied from 0.048 to 0.019 for itemsets of size 3. Note that the estimates become more and more accurate for larger itemsets. The reason is that the bulk of large itemsets will have subsets whose support is very small, thus giving better average trivial bounds. Nontrivial upper bounds occur slightly more frequently than nontrivial lower bounds; however, lower bounds give on average much better improvement over the trivial bounds (this is due to the fact that our trivial upper bounds are quite sophisticated, while the trivial lower bound is just assumed to be 0).

The percentage of itemsets having nontrivial bounds is quite small. However those itemsets who have high support (and thus are the most interesting) are more likely to get interesting nontrivial bounds. This can be seen in Tables 1(b) and 1(c), where up to 48% of itemsets have nontrivial bounds proving the usefulness of Theorem 3. Note that in this case the interval width increases with the size of the itemsets. This is due to the fact that for high supports we don't have large number of itemsets with low supports that would create trivial upper bounds.

The conclusions were analogous for the `mushroom` database.

Table 2 shows how the choice of the argument $r$ in Algorithm 1 influences the computation speed and the quality of the bounds. The results when $r$ is set to the highest possible value (size of the estimated itemset minus one) is given in Table 1(a).

The results show that limiting the value of $r$ to 2 or 3 gives a large speedup at a negligible decrease in accuracy. This is the approach we recommend. Also note that the proportion of itemsets with nontrivial bounds is higher for lower values of $r$. The same experiments repeated for frequent itemsets only yielded analogous results, so we omitted the data here.

Census Data with 1.8% Minimum Support

| $r = 2$ | | | | |
|---|---|---|---|---|
| itemset size | 3 | 4 | 5 | 6 |
| average interval width | 0.0482797 | 0.0315442 | 0.022993 | 0.0196671 |
| average upper bound | 0.0568679 | 0.0321734 | 0.0230122 | 0.0196671 |
| average lower bound | 0.00858817 | 0.000629199 | 1.925e-05 | 0 |
| itemsets with nontrivial bounds | 7% | 1% | 0.10% | 0% |
| time [ms/itemset] | 0.18 | 0.24 | 0.34 | 0.46 |
| $r = 3$ | | | | |
| itemset size | 3 | 4 | 5 | 6 |
| average interval width | 0.0482797 | 0.0313103 | 0.0228666 | 0.0196328 |
| average upper bound | 0.0568679 | 0.0319395 | 0.0228859 | 0.0196328 |
| average lower bound | 0.00858817 | 0.000629199 | 1.925e-05 | 0 |
| itemsets with nontrivial bounds | 7% | 0.50% | 0% | 0% |
| time [ms/itemset] | 0.18 | 0.3 | 0.53 | 0.92 |

**Table 2.** Influence of the order of inequalities on the bounds

Our last experimental result concerns estimating support of conjunctions allowing negated items using Corollary 1. Table 3 shows the results for the `census` ataset, with supports of all frequent 1- and 2-itemsets known (1.8% minimum support). In each of the itemsets exactly two of the items were negated. Again the inequalities gave fairly tight bounds.

Census Data with 1.8% Minimum Support

| itemset size | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|
| avg interval width | 0.040498 | 0.081989 | 0.0668155 | 0.0392651 | 0.0180174 |
| average upper bound | 0.171319 | 0.120666 | 0.0685168 | 0.0392925 | 0.0180174 |
| average lower bound | 0.130821 | 0.0386768 | 0.00170127 | 2.73405e-05 | 0 |
| time [ms/itemset] | 0.24 | 0.46 | 0.96 | 2.54 | 5.12 |

**Table 3.** Estimates for itemsets with negations

## 5 Conclusions and Open Problems

We presented a method of obtaining bounds for support of database queries based on supports of frequent itemsets discovered by a datamining algorithm by generalizing the Bonferroni inequalities. Specialized bounds for estimating support of itemsets, itemsets with negated items, as well as bounds for arbitrary queries have been presented. An experimental evaluation of the bounds is given as well showing that the bounds are capable of providing useful approximations.

Various other specialized Bonferroni inequalities for other types of queries could be considered. General inequalities in Theorem 1 can be used for this but the bounds they give are not always tight. It has also been shown in [JSR02] that for certain queries it is not possible to obtain tight bounds at all. Nevertheless, we believe that it is possible to obtain useful bounds for a large family of practically useful queries.

## References


[AMS+96] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Inkeri 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*, pages 307–328. AAAI Press, Menlo Park, 1996.

[BG99] L. Buzzigoli and A. Giusti. An algorithm to calculate the lower and upper bounds of the elements of an array given its marginals. In *Statistical Data Protection (SDP'98), Eurostat*, pages 131–147, Luxembourg, 1999.

[BM98] C. L. Blake and C. J. Merz. *UCI Repository of machine learning databases.* University of California, Irvine, Dept. of Information and Computer Sciences, http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.

[Dob01] A. Dobra. Computing sharp integer bounds for entries in contingency tables given a set of fixed marginals. Technical report, Department of Statistics, Carnegie Mellon University, http://www.stat.cmu.edu/~adobra/bonf-two.pdf, 2001.

[GS96] J. Galambos and I. Simonelli. *Bonferroni-type Inequalities with Applications.* Springer, 1996.

[JSR02] S. Jaroszewicz, D. Simovici, and I. Rosenberg. An inclusion-exclusion result for boolean polynomials and its applications in data mining. In *Proceedings of the Discrete Mathematics in Data Mining Workshop, SIAM Datamining Conference*, Washington, D.C., 2002.

[KLS96] J. Kahn, N. Linial, and A. Samorodnitsky. Inclusion-exclusion: Exact and approximate. *Combinatorica*, 16:465–477, 1996.

[LN90] Nathan Linial and Noam Nisan. Approximate inclusion-exclusion. In *ACM Symposium on Theory of Computing*, pages 260–270, 1990.

[MT96] H. Mannila and H. Toivonen. Multiple uses of frequent sets and condensed representations. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96)*, pages 189–194, Portland, Oregon, 1996.

[Rén58] A. Rényi. Quelques remarques ser les probabilites des evenements dependants. *Journal de Mathematique*, 37:393–398, 1958.