Approximating a Collection of Frequent Sets

Foto Afrati[*]
National Technical
University of Athens
Greece
afrati@softlab.ece.ntua.gr

Aristides Gionis
HIIT Basic Research Unit
Dept. of Computer Science
University of Helsinki
Finland
gionis@cs.helsinki.fi

Heikki Mannila
HIIT Basic Research Unit
Dept. of Computer Science
University of Helsinki
Finland
mannila@cs.helsinki.fi

ABSTRACT

One of the most well-studied problems in data mining is computing the collection of frequent item sets in large transactional databases. One obstacle for the applicability of frequent-set mining is that the size of the output collection can be far too large to be carefully examined and understood by the users. Even restricting the output to the border of the frequent item-set collection does not help much in alleviating the problem.

In this paper we address the issue of overwhelmingly large output size by introducing and studying the following problem: *What are the k sets that best approximate a collection of frequent item sets?* Our measure of approximating a collection of sets by k sets is defined to be the size of the collection covered by the the k sets, i.e., the part of the collection that is included in one of the k sets. We also specify a bound on the number of extra sets that are allowed to be covered. We examine different problem variants for which we demonstrate the hardness of the corresponding problems and we provide simple polynomial-time approximation algorithms. We give empirical evidence showing that the approximation methods work well in practice.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms

Algorithms

Keywords

Foundations of data mining, mining frequent itemsets

1. INTRODUCTION

The notion of discovery of *frequent patterns* started from the work of Agrawal, Imielinski, and Swami [1] on finding association rules and frequent item sets. The same basic idea of searching for patterns which occur frequently enough in the data carries over to several pattern domains (see e.g., [2, 11, 13, 4]). The collection of frequent patterns can be used in at least two different ways: first, one can be interested in the individual patterns and their occurrence frequencies; second, one can be interested in the whole collection, trying to obtain a global view of which patterns are frequent and which are not. The algorithms for finding frequent patterns are complete: they find all patterns that occur sufficiently often. Completeness is a desirable property, of course. However, in many cases the collection of frequent patterns is large, and obtaining a global understanding of which patterns are frequent and which are not is not easy. Even restricting the output to the border of the frequent item-set collection does not help much in alleviating the problem.

In this paper we consider the problem of finding a succinct representation of a collection of frequent sets. We aim at finding small and easy-to-understand approximations of the collection. The premise of our work is that such small approximations give a better understanding of the global structure of the data set without a significant sacrifice of information. Indeed, the collection of frequent patterns is always computed with respect to a frequency threshold, i.e., a lower limit on the occurrence probability of the pattern. This threshold is almost always somewhat arbitrary, and thus, there is no single "correct" collection of frequent patterns. Hence, one can argue that there is no reason to insist on computing the exact collection with respect to that threshold.

Our measure of approximating a set collection by k sets is defined to be the number of sets in the collection that are included in at one of the k sets. To avoid overgeneralization, we restrict the number of false positives allowed. As a simple example, consider the collection of frequent sets containing the sets ABC, ABD, ACD, AE, BE, and all their subsets (i.e., these five sets are the border of the collection). We can represent this collection approximately as the set of all subsets of $ABCD$ and ABE; this covers all the original sets, and there are only two false positives.

We show that while the problem of finding the best k-

[*] Part of this work was done while the author was visiting HIIT Basic Research Unit, Department of Computer Science, University of Helsinki, Finland.

KDD'04, August 22–25, 2004, Seattle, Washington, USA.

set approximation for a given collection is NP-hard, simple algorithms can be used to obtain very good approximation quality $(1 - 1/e)$. On real data, our empirical results show that using only $k = 20$ sets (corresponding to 7.5% of the size of the border of the collection), and allowing a false-positive ratio of 10%, one can cover the 70% of the original frequent set collection. Relatively simple visualization techniques can be used to give a good intuitive feel for collections of 20–50 sets, and hence it seems that our approach yields a good summary of the frequent set structure of large 0-1 data sets.

Our algorithm is based on the greedy approximation strategy, importance sampling, and a combinatorial lemma on the structure of collections of frequent sets. The method is simple but its analysis is somewhat intricate.

Next, we describe the problem in more detail. We are given a set U of N attributes $A_1, \ldots, A_N$ and a database consisting of *transactions*, which are subsets of U. The collection $\mathcal{D}$ of frequent sets consists of all attribute sets X such that at least a fraction of σ of the transactions in the database contain X as a subset. Then $\mathcal{D}$ is *downwards closed*, i.e., if $X \in \mathcal{D}$ and $Y \subseteq X$, then $Y \in \mathcal{D}$. Given the collection $\mathcal{D}$, we define the *border* $\mathcal{B}_+(\mathcal{D})$ of $\mathcal{D}$ as the collection of maximal sets in $\mathcal{D}$, i.e., $\mathcal{B}_+(\mathcal{D}) = \{X \in \mathcal{D} | \mathcal{D} \cap \mathcal{I}(X) = \{X\}\}$, where $\mathcal{I}(X)$ denotes the collection of supersets of X. Finally, given a set X, we denote by $\mathcal{P}(X)$ the powerset of X. We refer to the *lattice* $\mathscr{L}_U$ as the partial order naturally defined on the powerset $\mathcal{P}(U)$ using the subset relation "$\subseteq$".

We are interested in describing succinctly the downwards closed collection $\mathcal{D}$, and in order to do so successfully we are willing to tolerate some error. A natural way of representing $\mathcal{D}$ is to write it as the union of all subsets of k sets $Z_1, \ldots, Z_k$. That is, denoting

$$\mathcal{S}(Z_1, \ldots, Z_k) \equiv \bigcup_{i=1}^{k} \mathcal{P}(Z_i)$$

we look for sets $Z_1, \ldots, Z_k$ such that $\mathcal{D} \approx \mathcal{S}(Z_1, \ldots, Z_k)$. We say that $\mathcal{S}(Z_1, \ldots, Z_k)$ is *spanned* by the k sets $Z_1, \ldots, Z_k$, and we call $\mathcal{S}(Z_1, \ldots, Z_k)$ a k-spanned collection. The problem of succinct representation of $\mathcal{D}$ can now be formulated as follows:

Problem 1. *Given a downwards closed collection of sets $\mathcal{D}$, find a collection of sets $\mathcal{A}$ such that $\mathcal{A}$ is* spanned *by at most k sets and $\mathcal{A}$* approximates *$\mathcal{D}$ as well as possible.*

To make the statement of Problem 1 concrete we need to define the notion of distance between the input set collection $\mathcal{D}$ and a solution set collection $\mathcal{A}$. We measure the quality of approximation between two set collections $\mathcal{A}$ and $\mathcal{D}$ using the *coverage* measure $\mathrm{C}(\mathcal{A}, \mathcal{D})$, defined as the size of the intersection between $\mathcal{A}$ and $\mathcal{D}$. Naturally the goal is to maximize coverage.

Next, one has to define which sets are allowed as spanners. Without the restriction that the spanners of $\mathcal{A}$ should belong to $\mathcal{D}$, one can very easily maximize coverage by setting $\mathcal{A} = \mathscr{L}_U$, which is a solution that covers the whole $\mathcal{D}$ and it is spanned by just one set. However, $\mathcal{A} = \mathscr{L}_U$ is not an intuitive solution, since it introduces the maximum possible number of *false positives* (the sets in $\mathcal{A} \setminus \mathcal{D}$). The first choice to avoid this kind of unintuitive solutions is to restrict the spanners of $\mathcal{A}$ to be sets from $\mathcal{D}$ and therefore we have $\mathcal{A} \subseteq \mathcal{D}$. Under the restriction $\mathcal{A} \subseteq \mathcal{D}$ the goal of maximizing the coverage is equivalent with maximizing $|\mathcal{A}|$. Obviously the spanners of $\mathcal{A}^*$ (the k-spanned collection that best approximates $\mathcal{D}$) reside at the border of $\mathcal{D}$, and thus it is sufficient to restrict our search in the border of $\mathcal{D}$.

A middle road between restricting the spanners of $\mathcal{A}$ to be in $\mathcal{D}$ and having to choose $\mathcal{A} = \mathscr{L}_U$ is to restrict the selection of spanners of $\mathcal{A}$ in some other collection $\mathcal{D}'$ that is a supercollection of $\mathcal{D}$. The choice of $\mathcal{D}'$ can be natural in some applications. For example, if $\mathcal{D}$ is a collection of frequent item sets for a support threshold σ, $\mathcal{D}'$ can be the collection of frequent item sets for a smaller support threshold σ'. In other cases $\mathcal{D}'$ can be defined implicitly in terms of $\mathcal{D}$, for example, one could use all supersets of sets of $\mathcal{D}$ having at most t additional elements, i.e., $\mathcal{D}' = \mathcal{D}_t \equiv \{X \mid \text{there exists set } Z \text{ for which } Z \subseteq X, Z \in \mathcal{D}, \text{ and } |X \setminus Z| \leq t\}$. We will mainly consider a third alternative, where $\mathcal{D}'$ consists of those sets which have a sufficiently small fraction of false positives. We write $\mathrm{C}_{\mathcal{D}'}(\mathcal{A}, \mathcal{D})$ to make explicit that the spanners of $\mathcal{A}$ are chosen from the collection $\mathcal{D}'$. As before, it is sufficient to search for the solution in the border of $\mathcal{D}'$.

We now briefly describe the results of the paper: We distinguish between the case that the input to our problem is specified by the whole collection $\mathcal{D}$ and the case that the input is specified only by the border of $\mathcal{D}$. As the size of the collection can be exponential on the size of its border, the second case is more challenging.

- For the first case and when the spanning sets are selected from $\mathcal{D}$, we show that the problem of finding the best approximation $\mathcal{A}$ spanned by k sets is NP-hard, but it can be approximated to within a factor of $\left(1 - \frac{1}{e}\right)$.
- When the spanning sets are selected from those subsets for which the false-positive ratio is smaller than α, we show a compact lemma stating that the number of such sets is bounded by the square of the size of $\mathcal{D}$. The lemma yields a $\left(1 - \frac{1}{e}\right)$ approximation result.
- For the case that the input is specified by the border of $\mathcal{D}$, we are able to obtain an almost identical result—the price of specifying the collection $\mathcal{D}$ by giving only its border, i.e., with no redundancy, is only an ϵ loss in the approximation factor, for any $\epsilon > 0$. For showing this we use techniques from importance sampling in combination with the results from the previous case.
- We give empirical results demonstrating that a frequent-set collection can be approximated well by a small number of maximal sets. We performed experiments on three real data sets. As mentioned above, a typical result shows that using only $k = 20$ sets (corresponding to 7.5% of the size of the border of the collection), and allowing a false-positive ratio of 10%, we could cover the 70% of the total frequent set collection. Although the exact numbers depend on the input data set, similar trends showed in all cases.

The rest of this paper is organized as follows. In Section 2 we define the problem variants in more detail. Section 3 considers the case when $\mathcal{D}$ is given as input, and Section 4 the case where the input is the border of the collection. In Section 5 we describe our experiments, and in Section 6 we discuss related word. Finally, in Section 7 we offer our concluding remarks.

2. PROBLEM VARIANTS

We distinguish two variants of the problem depending on how the collection $\mathcal{D}$ is specified. In Section 3 we show that the task of approximating $\mathcal{D}$ is NP-hard, so polynomial time approximation algorithms need to be designed. However, the different ways of specifying $\mathcal{D}$ might change the size of the input at an exponential rate, so different techniques are required for each problem variant. Below we describe the two variants in order of increasing difficulty, (or equivalently, in order of decreasing input size).

Collection**.** The complete collection $\mathcal{D}$ is given as input. Considering as input the complete $\mathcal{D}$ creates a lot of redundancy since $\mathcal{D}$ can be precisely specified by its border $\mathcal{B}_+(\mathcal{D})$. However, the exact requirement in this variant is that our algorithm should be polynomial in $|\mathcal{D}|$.

Border**.** The border of $\mathcal{D}$ is given as input. In this case we allow the running time of our approximation algorithm to be $O(\text{poly}(|\mathcal{B}_+(\mathcal{D})|))$. The main problem here is that the size of $\mathcal{D}$ might be exponential in the size of $\mathcal{B}_+(\mathcal{D})$, therefore different techniques are required in order to stay within polynomial running time.

To unify our notation and distinguish more easily among the two cases, we restate Problem 1 as follows:

Problem 2. *Given a downwards closed collection of sets $\mathcal{D}$, find a collection of sets $\mathcal{A}$, such that $\mathcal{A}$ is spanned by at most k sets and $\mathcal{A}$ approximates $\mathcal{D}$ as well as possible. We call the problem* Aprx-Collection *when the whole collection is specified as input, and* Aprx-Border *when only the border of the collection is given as input. The quality of approximation is measured according to the coverage measure* C*. The optimal solution to the problem for all cases is denoted by $\mathcal{A}^*$.*

3. WHOLE COLLECTION AS INPUT

3.1 Selecting spanners from the collection

We first consider the case that the spanner sets in the solution $\mathcal{A}$ are restricted to be inside $\mathcal{D}$. The problem to solve is that of selecting the k sets in $\mathcal{D}$ that maximize the intersection with $\mathcal{D}$. We can notice immediately that this problem is a special case of Max k-Cover. An instance of Max k-Cover is specified by a collection of sets, and the goal is to select the k sets in the collection that maximize the number of covered elements. A well-known algorithm for the Max k-Cover problem is the *greedy* algorithm, which can be described as follows: Initially, the algorithm puts all the elements in a list of *uncovered* elements. Then it proceeds in performing k iterations, where in each iteration one new set is added to the solution. During the j-th iteration, for $1 \leq j \leq k$, the algorithm *i*) finds the set A_j that covers the most uncovered elements, *ii*) adds A_j to the solution, and *iii*) removes the elements covered by A_j from its list of uncovered elements. The greedy algorithm is known to provide a $\left(1-\frac{1}{e}\right)$ approximation ratio to Max k-Cover (e.g., see [9, pg. 136]), so the following is immediate.

Theorem 1. *For* Aprx-Collection *as defined in Problem 2, we can find a collection $\mathcal{A}$ spanned by k sets such that* $\mathrm{C}(\mathcal{A},\mathcal{D}) \geq \left(1-\frac{1}{e}\right)\mathrm{C}(\mathcal{A}^*,\mathcal{D})$.

Next we show that Aprx-Collection is indeed an NP-hard problem. Notice that the connection with Max k-Cover described above does not imply immediately the NP-hardness result, since it is not clear how to transform an arbitrary collection to a downwards closed collection.

Theorem 2. Aprx-Collection *is NP-hard.*

Proof. We show a transformation from the 3D Matching problem [6]. An instance of 3D Matching is specified by a set of "edges" $E \subseteq X \times Y \times Z$, where X, Y, and Z are disjoint sets having the same number q of elements. The goal is to determine if there is a complete matching for E, i.e., a subset $M \subseteq E$ of cardinality $|M| = q$ such that no elements in M agree in any coordinate. The transformation to Aprx-Collection is defined as follows:

Consider an instance $\mathcal{I} = (X, Y, Z, E)$ of 3D Matching. An instance of Aprx-Collection can then be defined by considering a collection of sets $\mathcal{D}(\mathcal{I})$ over the universe of elements $U = X \cup Y \cup Z$. For each edge $e = (x_i, y_j, z_k) \in E$ we add in the collection $\mathcal{D}(\mathcal{I})$ the subcollection $\mathcal{D}(e) = \mathcal{P}(\{x_i, y_j, z_k\}) = \{\{x_i, y_j, z_k\}, \{x_i, y_j\}, \{x_i, z_k\}, \{y_j, z_k\}, \{x_i\}, \{y_j\}, \{z_k\}, \emptyset\}$, that is $\mathcal{D}(\mathcal{I}) = \bigcup_{e \in E} \mathcal{D}(e)$. By construction, the collection $\mathcal{D}(\mathcal{I})$ is downwards closed. The value of k required for Aprx-Collection is set to q, and the target value of $\mathrm{C}(\mathcal{A}, \mathcal{D}(\mathcal{I}))$ for a solution $\mathcal{A}$ is set to $7q + 1$.

The transformation can be clearly computed in polynomial time. Furthermore, we can show that in the instance $\mathcal{I}$ there exists a complete matching if and only if in the collection $\mathcal{D}(\mathcal{I})$ there exists a collection $\mathcal{A}$ that is spanned by q sets and it has coverage at least $7q+1$. To prove the equivalence, disregarding the $\emptyset$ set that is covered by selecting any other set, any set in the collection $\mathcal{D}(\mathcal{I})$ covers at most 7 sets. Therefore, the only way to obtain a solution $\mathcal{A}$ that is spanned by q sets and it has coverage value $7q+1$ is that all of the q spanners of $\mathcal{A}$ are 3-element sets and their powersets are pairwise disjoint. However, such a solution in $\mathcal{D}(\mathcal{I})$ corresponds to a complete matching in $\mathcal{I}$. The proof of the converse is based on the same idea, i.e., a complete matching in $\mathcal{I}$ corresponds to disjoint (except the $\emptyset$ set) 3-element spanners in $\mathcal{D}(\mathcal{I})$. □

3.2 Selecting spanners outside the collection

In this section we consider the case that the spanner sets for a solution $\mathcal{A}$ are allowed to be outside the collection $\mathcal{D}$. A motivating example for investigating this case is the following.

Example 1. *Imagine the contrived but illustrative situation that* all *sets of size t of the lattice are frequent, and* no *sets of size $t' > t$ are frequent. Intuitively, we say that the collection of frequent item sets is "flat". In this situation, it is reasonable to consider reporting a set of size $t + 1$, even though such a set is not frequent itself. Quantitatively, by reporting one infrequent set (one false positive), we capture $t + 1$ frequent sets of the border. If t is large enough, one could also consider reporting a set of size $t + 2$: in that way by having $t+3$ false positives, we capture $\binom{t+2}{2}$ frequent sets of the border.*

Assuming that the collection of candidate spanners $\mathcal{D}'$ is somehow specified to the problem instance, one can use the greedy algorithm described in the previous section and obtain a similar kind of approximation guarantee for the measure $\mathrm{C}_{\mathcal{D}'}(\mathcal{A}, \mathcal{D})$. Two methods of specifying the collection

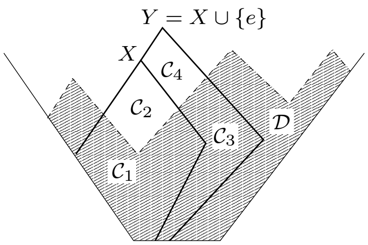

Figure 1: A schematic view of the concepts used in the proof of Lemma 1.

$\mathcal{D}'$ were mentioned earlier. The first method is to reduce the support threshold from σ to σ' and consider as $\mathcal{D}'$ the (border of the) σ'-frequent item sets. The second method is to *extend* all sets in $\mathcal{D}$ by considering some of their supersets. However, one can verify that in both of these methods the size of the collection $\mathcal{D}'$ can potentially be superpolynomial in the size of $\mathcal{D}$. On the other hand, the running time of the greedy algorithm is polynomial in both $|\mathcal{D}|$ and $|\mathcal{D}'|$, therefore one cannot guarantee that the running time remains polynomial in the size of $\mathcal{D}$. In the rest of this section, we will describe an intuitive definition for the candidate collection $\mathcal{D}'$ that guarantees that its size is polynomial in the size of $\mathcal{D}$, and therefore the running time of the greedy algorithm is polynomial.

After looking again at Example 1, it seems intuitive to consider adding in $\mathcal{D}'$ a set S that is not in $\mathcal{D}$, only if S covers a large part of $\mathcal{D}$ while it does not introduce many additional sets. To formalize this notion, for each set S we define the *false-positive ratio* function $f_+(\cdot)$ to be the ratio of the number of sets not in $\mathcal{D}$ over the number of sets in $\mathcal{D}$ covered by S. In other words

$$f_+(S) \equiv \frac{|\mathcal{P}(S) \setminus \mathcal{D}|}{|\mathcal{P}(S) \cap \mathcal{D}|}.$$

Notice that the false-positive ratio of a set S can always be defined since at least the empty set belongs simultaneously in $\mathcal{D}$ and in $\mathcal{P}(S)$. For a set S, $f_+(S) = 0$ is equivalent to $S \in \mathcal{D}$.

A collection of candidate spanners can now be defined using the notion of false-positive ratio. Given a *false-positive-threshold* value α, we define the collection of candidates $\mathcal{D}_\alpha$ to be

$$\mathcal{D}_\alpha \equiv \{S \mid f_+(S) < \alpha\}.$$

That is, sets in $\mathcal{D}_\alpha$ introduce at most a fraction of α false positives over the sets in $\mathcal{D}$ that they cover. The threshold α can take any positive value, however, as we will see shortly, we are particularly interested in $\mathcal{D}_1$, i.e., candidates whose false-positive ratio is smaller than 1.

We will first show that the collection $\mathcal{D}_\alpha$ is downwards closed, and thus it can be computed using a simple APriori-type algorithm.

Lemma 1. *For any threshold value $\alpha > 0$, the collection $\mathcal{D}_\alpha$ is downwards closed.*

Proof. Assume that there exists a collection $\mathcal{D}$ and a value α for which the hypothesis of the lemma is not true. Then there should exist two sets X and Y such that $f_+(X) \geq \alpha$, $f_+(Y) < \alpha$, and $Y = X \cup \{e\}$, that is, Y extends X by just one element.

We partition the powerset $\mathcal{P}(Y)$ into four disjoint collections: $\mathcal{C}_1 = \{S \in \mathcal{P}(Y) \mid e \notin S \text{ and } S \in \mathcal{D}\}$, $\mathcal{C}_2 = \{S \in \mathcal{P}(Y) \mid e \notin S \text{ and } S \notin \mathcal{D}\}$, $\mathcal{C}_3 = \{S \in \mathcal{P}(Y) \mid e \in S \text{ and } S \in \mathcal{D}\}$, and $\mathcal{C}_4 = \{S \in \mathcal{P}(Y) \mid e \in S \text{ and } S \notin \mathcal{D}\}$. Define $s_i = |\mathcal{C}_i|$, for $i = 1, 2, 3, 4$, and observe that

$$f_+(X) = \frac{s_2}{s_1}, \quad \text{and} \quad f_+(Y) = \frac{s_2 + s_4}{s_1 + s_3}.$$

Finally, define $\mathcal{C}_{\bar{e}} = \mathcal{C}_1 \cup \mathcal{C}_2$ (which, in fact, is $\mathcal{P}(X)$) and $\mathcal{C}_e = \mathcal{C}_3 \cup \mathcal{C}_4$ (that is, $\mathcal{P}(Y) \setminus \mathcal{P}(X)$). For a visualization aid of the above concepts and definitions see Figure 1.

Notice that given our definitions, each set A in the collection $\mathcal{C}_{\bar{e}}$ has a "copy" set $A \cup \{e\}$ in the collection $\mathcal{C}_e$, and vice versa. This one-to-one mapping of $\mathcal{C}_{\bar{e}}$ to $\mathcal{C}_e$ implies that $s_1 + s_2 = s_3 + s_4$. The crucial observation for the proof of the lemma is that since $\mathcal{D}$ is downwards closed, for each set in $\mathcal{C}_e$ that also belongs to $\mathcal{C}_3$, its "copy" in $\mathcal{C}_{\bar{e}}$ should belong to $\mathcal{C}_1$. In other words, the "copies" of the sets in $\mathcal{C}_3$ is a subset of $\mathcal{C}_1$, which implies that $s_1 \geq s_3$. Combining the facts $s_1 + s_2 = s_3 + s_4$ and $s_1 \geq s_3$ we obtain $s_2 \leq s_4$, and therefore

$$\frac{s_2 + s_4}{s_1 + s_3} \geq \frac{2s_2}{2s_1} = f_+(X) \geq \alpha.$$

However, the above conclusion contradicts with our initial assumption that $\frac{s_2+s_4}{s_1+s_3} = f_+(Y) < \alpha$. □

One potential obstacle in using the definition of $\mathcal{D}_\alpha$, is that, although it is intuitive, it does not provide us with an obvious upper bound on the number of candidates to be used. However, we next show how to overcome this problem and obtain such a bound for the special case of false-positive threshold value $\alpha = 1$. Our bound is based on the rather interesting containment property of $\mathcal{D}_1$.

Lemma 2. *Any set in $\mathcal{D}_1$ can be expressed as the union of two sets in $\mathcal{D}$, that is,*

$$\mathcal{D}_1 \subseteq \{X \cup Y \mid X, Y \in \mathcal{D}\}.$$

Proof. Consider a set Z for which there are not exist two sets $X, Y \in \mathcal{D}$ such that $Z = X \cup Y$. We will show that $f_+(Z) \geq 1$, and so $Z \notin \mathcal{D}_1$. Define the partition of $\mathcal{P}(Z)$ into the disjoint collections $\mathcal{D}_+(Z) = \mathcal{P}(Z) \cap \mathcal{D}$ and $\mathcal{D}_-(Z) = \mathcal{P}(Z) \setminus \mathcal{D}$. Notice that $f_+(Z) = |\mathcal{D}_-(Z)|/|\mathcal{D}_+(Z)|$. Let X be any set in $\mathcal{D}_+(Z)$. The complement of X with respect to Z (i.e., the set $Z \setminus X$) should belong to $\mathcal{D}_-(Z)$, otherwise the assumption for Z would be violated. Therefore, i.e., by complementation with respect to Z, we correspond each set from $\mathcal{D}_+(Z)$ to a set in $\mathcal{D}_-(Z)$, and no two sets from $\mathcal{D}_+(Z)$ correspond to the same set in $\mathcal{D}_-(Z)$. Thus $|\mathcal{D}_-(Z)| \geq |\mathcal{D}_+(Z)|$ or $f_+(Z) \geq 1$. □

Corollary 1. *We have $|\mathcal{D}_1| = O(|\mathcal{D}|^2)$.*

Using the fact that the collections $\mathcal{D}_\alpha$ are downwards closed, it is clear to see that $\mathcal{D}_\alpha \subseteq \mathcal{D}_\beta$ for $\alpha \leq \beta$. Therefore, the same upper bound of Corollary 1 can be used for all values $\alpha < 1$, that is $|\mathcal{D}_\alpha| = O(|\mathcal{D}|^2)$. For small values of α the bound might be crude, but nevertheless polynomial. Furthermore, the algorithm will perform much better in practice than the bound suggests (the running time depends on the actual size of $\mathcal{D}_\alpha$). An empirical estimation of the real bound for $\alpha < 1$ is discussed in Section 5. Also notice that Lemma 2 sheds some light in understanding the structure of $\mathcal{D}_\alpha$. For example, if $\mathcal{D}$ is spanned by only one set, i.e., $\mathcal{D} = \mathcal{P}(X)$, then we get $\mathcal{D}_1 = \mathcal{D}$, which can also be verified by the definition of false-positive ratio. We now combine all of the above steps and obtain the main result of this section.

THEOREM 3. *Consider* APRX-COLLECTION*, as defined in Problem 2. For a given false-positive threshold value α, we write C_α to denote the coverage measure of approximating $\mathcal{D}$ when the collection of candidate spanners allowed to be used is the collection $\mathcal{D}_\alpha$. Then, for any $\alpha \leq 1$, we can find in polynomial time a collection $\mathcal{A}$ spanned by k sets such that $\mathrm{C}_\alpha(\mathcal{A}, \mathcal{D}) \geq \left(1 - \frac{1}{e}\right) \mathrm{C}_\alpha(\mathcal{A}^*, \mathcal{D})$.*

PROOF. From Corollary 1, we know that the size of the candidate collection $\mathcal{D}_\alpha$ is quadratic in $|\mathcal{D}|$. Using Lemma 1, we can compute $\mathcal{D}_\alpha$ in an APriori fashion, and the running time is polynomial in $|\mathcal{D}|$. Now, we use the greedy algorithm with candidate sets restricted in $\mathcal{B}_+(\mathcal{D}_\alpha)$. The overall running time is clearly polynomial. Finally, the analysis of the greedy guarantees that the approximation ratio is at least $\left(1 - \frac{1}{e}\right)$. □

4. THE BORDER AS INPUT

In this section we explain how one can use the greedy algorithm in order to deal with the case that only the border is specified as input to the problem. Our main contribution is to show that we are able to obtain a result almost identical to the one presented in Section 3.1—the price of specifying $\mathcal{D}$ with no redundancy is only an ϵ loss in the approximation factor, for any $\epsilon > 0$. We start by explaining where the difficulty lies in using the greedy algorithm of Section 3.1, and then we describe the necessary remedy for the algorithm to run in polynomial time.

As we already mentioned in Section 2, the size of $\mathcal{D}$ can be exponentially large in the size of $\mathcal{B}_+(\mathcal{D})$. The greedy algorithm actually utilizes resources polynomial in $|\mathcal{D}|$, since at each step it evaluates how many new elements of $\mathcal{D}$ are covered by a potential candidate set to be added in the solution $\mathcal{A}$. Assume now that we apply the greedy algorithm in the case that only the border is specified as input. The first set S_1 to be added in the solution is the set in $\mathcal{B}_+(\mathcal{D})$ that covers the most sets in $\mathcal{D}$. A base set on t items covers exactly 2^t itemsets, therefore the first set S_1 will be the set with maximum cardinality in $\mathcal{B}_+(\mathcal{D})$ (breaking ties arbitrarily). The second set S_2 will be the one that maximizes $|S_1 \cup S_2|$ given S_1, which can be computed using the formula $|S_1 \cup S_2| = |S_1| + |S_2| - |S_1 \cap S_2|$.

In general, in order to find at each step of the greedy algorithm the set in the border that covers the most uncovered sets in $\mathcal{D}$, we need to compute the size of the the union $S_1 \cup \ldots \cup S_m$. Resorting to the *inclusion-exclusion formula* [5], as we did for S_2, is a possibility but not an efficient method, since the number of terms in the formula is exponential in m.

The first idea for computing the size of the union $S_1 \cup \ldots \cup S_m$ is to use a *Monte Carlo sampling* method: Denote by $S^{(m)}$ the union $S_1 \cup \ldots \cup S_m$. To estimate $|S^{(m)}|$ sample n sets uniformly at random from $\mathscr{L}_U$ and count how many of them belong in $S^{(m)}$. Let this count be x. Then the ratio $\frac{x}{n}$ is a good *estimator* for the ratio $|S^{(m)}|/|\mathscr{L}_U|$, and since we know that $|\mathscr{L}_U| = 2^N$ we can estimate $S^{(m)}$ as $\frac{x}{n} \cdot 2^N$. In particular, using the Chernoff bounds we can show the following.

FACT 1. *The sampling method described above provides and ϵ-approximation to $|S^{(m)}|$ with probability at least $1 - \delta$, provided that*

$$n \geq \frac{2^N}{\epsilon^2 |S^{(m)}|} \log \frac{2}{\delta}.$$

Unfortunately, the idea of uniform sampling is not good enough. The reason is that if $|S^{(m)}|$ is small compared to 2^N, then we need to sample many sets from $\mathscr{L}_U$—not to mention that to obtain the required number of samples requires knowledge of $|S^{(m)}|$, which is precisely what we are trying to compute.

Fortunately, the problem posed by uniform sampling can be overcome by resorting to the technique of *importance sampling*. Here we give a short overview of the method, more details can be found in [12, Section 11.2.2]. Recall that our goal is to compute to compute $|S^{(m)}| = |S_1 \cup \ldots \cup S_m|$, where each S_i is a subset of $\mathscr{L}_U$. For simplifying the notation, denote $V = \mathscr{L}_U$, so each S_i contains elements from the universe V. Also assume that we are given small positive numbers ϵ and δ. Given the above setting, the method of importance sampling provides an ϵ-accurate estimate for $|S^{(m)}|$ with probability at least $1 - \delta$, provided that the following three conditions are satisfied.

(i) For all i, we can compute $|S_i|$.

(ii) We can sample uniformly at random from each S_i.

(iii) For all $v \in V$, we can verify efficiently if $v \in S_i$.

In our case, all of the above conditions are satisfied: For (i) and (ii) we used the fact that S_i are downwards closed sets. In particular, for (i), notice that if the base set of S_i contains t elements then $|S_i| = 2^t$. For (ii), let R be a set that contains each element of the base items of S_i with probability 1/2. Then, it is easy to see that R is a uniform sample from the itemsets in S_i. Finally, for (iii), given an element $v \in V$ we can trivially verify if it is also an element of S_i: just check if the base set of v is a subset of the base set of S_i.

The importance sampling method considers the multiset $M^{(m)} = S_1 \uplus \ldots \uplus S_m$, where the elements of $M^{(m)}$ are ordered pairs of the form (v, i) corresponding to $v \in S_i$. In other words, the elements of $M^{(m)}$ are the elements of $S^{(m)}$ appended with an index that indicates due to which S_i they appear in $S^{(m)}$. Notice that the size of $M^{(m)}$ can be trivially computed as $|M^{(m)}| = \sum_i^m |S_i|$.

The multiset $M^{(m)}$ is then divided into equivalent classes, where each class contains all pairs (v, i) that correspond to the same element $v \in S^{(m)}$. That is, each equivalent class corresponds to an element of $v \in S^{(m)}$ and contains all indices i for which $v \in S_i$. For each equivalent class one pair (v, i) is defined to be the *canonical representation* for the class. Now $|S^{(m)}|$ can be approximated by generating random elements in $M^{(m)}$ and estimating the fraction of those that correspond to a canonical representation of an equivalent class. The intuition is that instead of sampling from the whole space $V = \mathscr{L}_U$, we sample only from the set $M^{(m)}$. The problem that appeared before with the uniform sampling disappears now because each element in $S^{(m)}$ can contribute at most m elements in $M^{(m)}$, and therefore the ratio $\frac{|S^{(m)}|}{|M^{(m)}|}$ is bounded from below by $1/m$. Now by applying a sampling result similar to the one given in Fact 1, we can estimate the ratio $\frac{|S^{(m)}|}{|M^{(m)}|}$ using just $\frac{m}{\epsilon^2}\log\frac{2}{\delta}$ samples, i.e., the number of samples required is polynomial in m. After estimating $\frac{|S^{(m)}|}{|M^{(m)}|}$ and since the value of $|M^{(m)}|$ is known we can also estimate $|S^{(m)}|$.

Now we can use the above approximation scheme as part of the greedy algorithm. The idea is to approximate the value of the coverage measure for each candidate solution $\mathcal{A}$ by the method of importance sampling, and then select the set that maximizes the estimated coverage. By the approximation result, the coverage C of each set is estimated within the range $[(1-\epsilon)C, (1+\epsilon)C]$, with probability at least $1-\delta$. Thus, in each iteration of the greedy algorithm, we can find a set whose coverage is at least a $(1-\epsilon)$ fraction of the largest coverage, and therefore the quality of approximation of the greedy is multiplied by a factor of $(1-\epsilon)$. Notice that the greedy calls the importance-sampling approximation scheme a polynomial number of times, therefore, in order to obtain a high probability result we need to set $\frac{1}{\delta} = \Omega(\text{poly}(\mathcal{B}_+(\mathcal{D})))$. However, this setting for δ does not create a serious overhead in the algorithm, since the complexity of the importance-sampling approximation scheme is only logarithmic in $\frac{1}{\delta}$. Summarizing the above discussion, we have shown the following.

Theorem 4. *For* Aprx-Border *as defined in Problem 2 and for any $\epsilon > 0$, we can find with high probability a k-spanned collection $\mathcal{A}$ such that*

$$\mathrm{C}(\mathcal{A},\mathcal{D}) \geq \left(1-\frac{1}{e}-\epsilon\right)\mathrm{C}(\mathcal{A}^*,\mathcal{D}).$$

Notice that the NP-hardness of Aprx-Border can also be established; the same reduction as in Theorem 2 can be used.

In fact, the technique described in this section can be used as a fully polynomial randomized approximation scheme (FPRAS) for the problem of estimating the size of a frequent itemset collection given the border of the collection.

5. EXPERIMENTAL EVALUATION

To verify the applicability of the problem studied in this paper, we implemented the proposed algorithms and we tested their behavior on three different real sets of data.

The first data set, **Mushroom**, was obtained from the machine learning repository of UC Irvine. A support threshold of 25% was used to obtain a collection of 6624 item sets. The number of sets in the border was 98 and the average number of items for the border sets was 6.6. The second set, **Course**, is from anonymized student/course registration data in the Department of Computer Science at the University of Helsinki. Frequent course sets were obtained using a support threshold of 2.2%, yielding a collection of size 1637. The size of the border for the second data set was 268 and the average number of items per border set was 4. Finally, our third data set, **BMS**, is owned by Blue Martini™ and it has been made available by Ronny Kohavi [10]. The data set contains click-stream data of a small company, and it was used at the KDD CUP 2000. Applying a support threshold of 0.1% we obtained a collection of 8192 item sets with border size 3546 and average item size for the border sets equal to 3. The three data sets, in the order described above, can be characterized from "narrow" (small border consisting of large item sets) to "wide" (large border consisting of small sets). Thus, we believe that our experiments capture a fairly large range of typical cases.

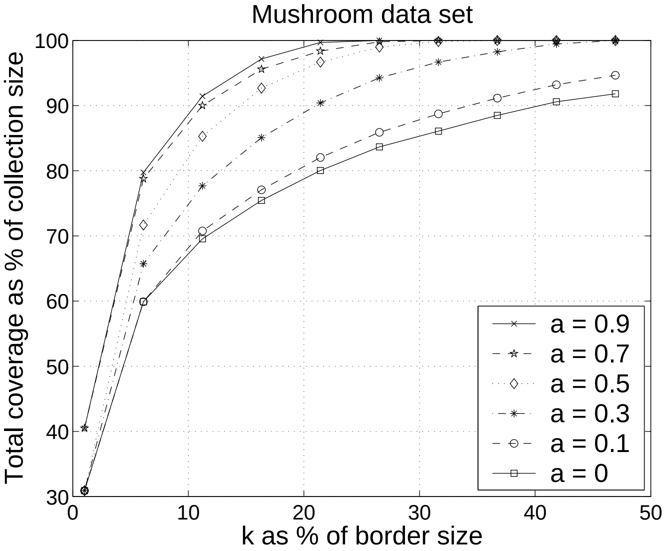

Figure 2: Coverage of frequent set with using up to $k = 50$ sets for Mushroom dataset.

For extracting frequent sets from the above data sets we used implementation of frequent itemset mining algorithms available by Bart Goethals [7].

We run the greedy algorithm with value of k up to 50, and values of the false-positive threshold parameter α in the range $(0, 1)$. Notice that the value $k = 50$ corresponds to about 50%, 18% and 1.4% of the border size, for the data sets Mushroom, Course, and BMS, respectively. The results for the data sets **Mushroom**, **Course**, and **BMS** are shown in Figures 2, 3, and 4, respectively. In all cases we see that with "few" sets we can cover a "large" part of the collection. For instance, for the **mushroom** data set, only 20 out of 98 item-sets in the border can cover about 80% of the collection without introducing any false positives, whereas if we allow a percentage of at most .3 false positives, then 20 sets cover 90% and 30 sets cover 97% of the collection. Obviously, increasing α corresponds to better coverage (with a single exception(!) in the Course data set for values $\alpha = 0.7$ and $\alpha = 0.9$; this is probably due to fact that the greedy is an approximate algorithm). Also, as one can expect, the more "narrow" the data set, the better the coverage achieved for the same (absolute) value of k.

Next we measure the number of candidates introduced and the size of the border of the candidates, as a function of α. This is shown in Figures 5 and 6. As we mentioned in Section 3.2, the quadratic upper bound used from the case $\alpha = 1$ is expected to be rather crude for smaller values of α. In practice, the number of the candidates can be much smaller than quadratic, e.g., for $\alpha = .3$ our experiments show that the number of candidates is at most $|\mathcal{D}|^{1.25}$. Assuming that the candidate size $|\mathcal{D}_\alpha|$ has polynomial dependency on $|\mathcal{D}|$, the exponent of the polynomial can be estimated from the ratio $\log|\mathcal{D}_\alpha| / \log|\mathcal{D}|$, which is computed using our data sets and plotted in Figure 5 for various values of α. From Figure 5, a reasonable conjecture is that the true exponent

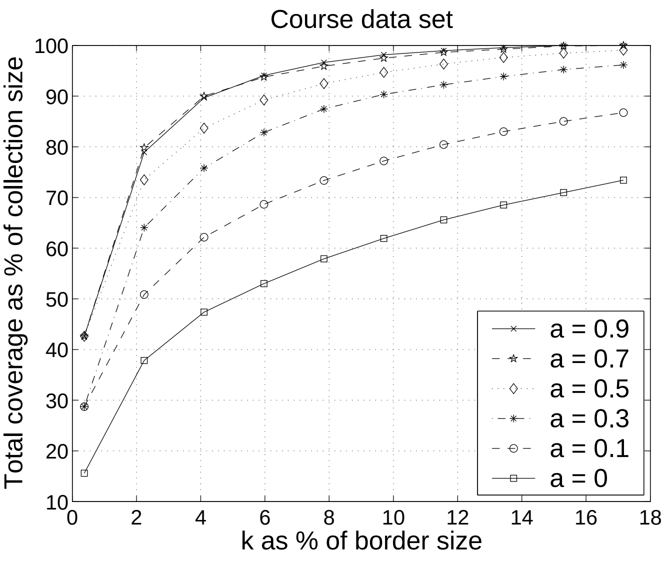

Figure 3: Coverage of frequent set with using up to $k = 50$ sets for Course dataset.

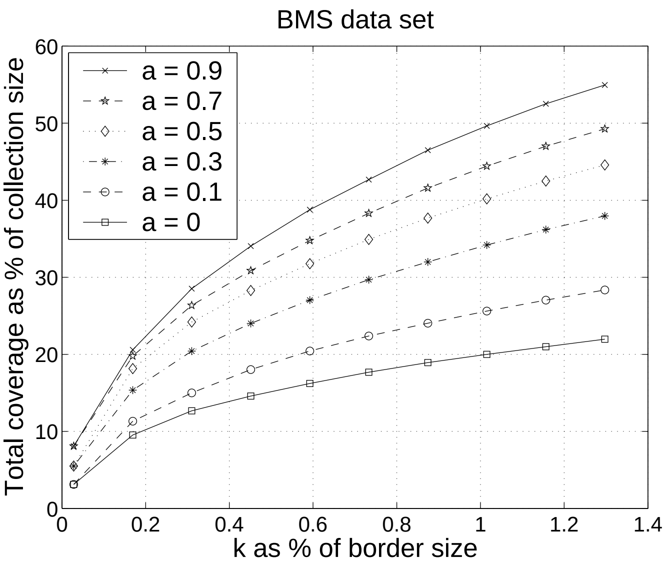

Figure 4: Coverage of frequent set with using up to $k = 50$ sets for BMS dataset.

is $1 + \alpha$. Also one can see that less candidates are generated for the more "narrow" data sets. Finally, in Figure 6, we show a similar kind of dependency between the border size of the candidate collection and the border size of the input collection, that is, the ratio $\log |\mathcal{B}_+(\mathcal{D}_\alpha)| / \log |\mathcal{B}_+(\mathcal{D})|$ is plotted as a function of α. In this case, we see that the "narrowness" of the data set does not influence the exponent of growth as much.

The results indicate that using as few as 20–50 sets in the approximation gives often quite a good approximation of the original collection. This implies that there is hope to obtain good succinct representations of large 0-1 datasets.

6. RELATED WORK

Related to our paper in the respect of attempting to reduce the size of the output of frequent item-set patterns is the work of Han et al. [8] on mining top-k frequent closed patterns, as well as work on closed frequent item-sets and condensed frequent item sets, see for example, Pei et al. [14], Pasquier et al. [13], and Calders and Goethals [3]. In [8] the goal is to find the k most frequent sets containing at least `min_l` items. This goal, however, is different from our setting where we ask for the k sets that best approximate the frequent item-set collection in the sense of set coverage. The work on frequent closed item sets attempts to compress the collection of frequent sets in a lossless manner, while for the condensed frequent item sets the idea is to be able to reduce the output size by allowing a small error on the support of the frequent item sets.

7. CONCLUDING REMARKS

We have considered the problem of approximating a collection of frequent sets. We showed that if the whole collection of frequent sets or its border are given as input, the best collection of sets spanned by k sets can be approximated to within $\left(1 - \frac{1}{e}\right)$. We also showed that the same results hold when the sets used to span the collection are from the collections $\mathcal{D}_\alpha$. The results used the greedy approximation algorithm, importance sampling, and a lemma bounding the size of $\mathcal{D}_\alpha$. The results can also be generalized to any setting of frequent pattern discovery, provided some mild computability conditions hold. The empirical results show that the methods work well in practice.

Several open problems remain. First of all, the properties of different measures of approximation merit further study. The measure we use counts the number of positive examples covered, and the negative examples are bounded by the choice of the collection $\mathcal{D}'$. Another measure of approximation quality would simply be the number of positive examples covered minus the number of negative examples covered. It turns out, however, that in this case the greedy algorithm performs arbitrarily bad.

Along the lines of using more elaborate measures is the idea of taking into account the support of the itemsets in the covered area, as well as the support of the false positives. One conceptual difficulty is how exactly to integrate the support information. In our current formulation we count the number of itemsets covered by a collection of spanners, however, extending this formulation to simply adding the supports of the itemsets is perhaps less intuitive.

The case when the input is the original database is perhaps the most interesting open algorithmic question. This case presents significant difficulties. First, computing the border in time polynomial to its size is a main open problem. Furthermore, the size of the border can be exponential in the size of the database, and therefore one cannot afford looking at the whole search space—some kind of sampling method needs to be applied. One can try to form an approximation of the border of the collection of frequent sets by random walks on the subset lattice. However, for such a "random walk" sampling method we were able to construct "bad" cases, i.e., cases in which the probability of finding any set that covers a not so small fraction of what the best set covers is not sufficiently large. Notice, however, that given the transactional database one can always compute

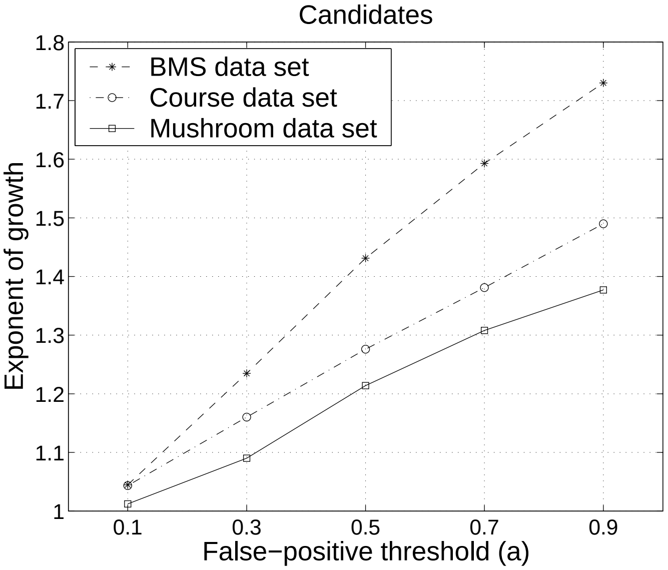

Figure 5: Size of the candidate collection.

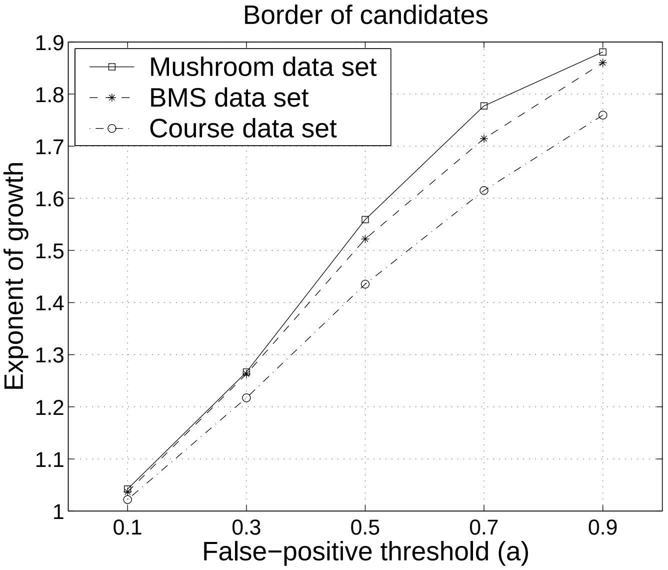

Figure 6: Size of the border of candidate collection.

the border of $\mathcal{D}$ using one of the many algorithms in the data-mining literature and then apply our technique for the BORDER case. We expect this method to work quite well in practice.

Finally, from the practical point of view, user interface techniques that use approximation of frequent set collections might be interesting.

8. ACKNOWLEDGMENTS

We would like to thank Floris Geerts, Bart Goethals, Taneli Mielikäinen, and Jouni Seppänen for many useful comments.

9. REFERENCES

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining associations between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Proceedings of the IEEE International Conference on Data Engineering*, pages 3–14, 1995.

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

[4] Min Fang, Narayanan Shivakumar, Hector Garcia-Molina, Rajeev Motwani, and Jeffrey D. Ullman. Computing iceberg queries efficiently. In *Proceedings of the 24th International Conference on Very Large Data Bases*, pages 299–310, New York City, USA, 1998.

[5] William Feller. *An introduction to probability theory and its applications*. John Wiley & Sons, 1968.

[6] M.R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[7] Bart Goethals. Frequent itemset mining implementations. `http://www.cs.helsinki.fi/u/goethals/software/`.

[8] Jiawei Han, Jianyong Wang, Ying Lu, and Petre Tzvetkov. Mining top-k frequent closed patterns without minimum support. In *Proceedings of the IEEE International Conference on Data Mining*, pages 211–218, 2002.

[9] Dorit Hochbaum, editor. *Approximation algorithms for NP-hard problems*. PWS Publishing Company, 1997.

[10] Ron Kohavi, Carla Brodley, Brian Frasca, Llew Mason, and Zijian Zheng. KDD-Cup 2000 Organizers' Report: Peeling the Onion. *SIGKDD Explorations*, 2(2):86–98, 2000. `http://www.ecn.purdue.edu/KDDCUP/`.

[11] Heikki Mannila, Hannu Toivonen, and Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289, 1997.

[12] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[13] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Discovering frequent closed itemsets for association rules. In *7th International Conference on Database Theory*, pages 398–416, 1999.

[14] Jian Pei, Guozhu Dong, Wei Zou, and Jiawei Han. On computing condensed frequent pattern bases. In *Proceedings of the IEEE International Conference on Data Mining*, 2002.