On Approximating Minimum Infrequent and Maximum Frequent Sets

Mario Boley

Fraunhofer IAIS, Schloss Birlinghoven, Sankt Augustin, Germany
mario.boley@iais.fraunhofer.de

Abstract. The maximum cardinality of a frequent set as well as the minimum cardinality of an infrequent set are important characteristic numbers in frequent (item) set mining. Gunopulos et al. [10] have shown that finding a maximum frequent set is **NP**-hard. In this paper I show that the minimization problem is also **NP**-hard. As a next step I investigate whether these problems can be approximated. While a simple greedy algorithm turns out to approximate a minimum infrequent set within a logarithmic factor one can show that there is no such algorithm for the maximization problem.

1 Introduction

Finding sets of items that appear concurrently in at least a specified number of records in a given database is an important task in data mining. This so-called frequency criterion for sets is used as an additional condition for different interestingness predicates. Examples are association rules [2], correlations [5], or emerging patterns [7].

Algorithms usually perform an exhaustive enumeration of the family of frequent sets or of a reduced family like closed frequent sets or maximal frequent sets. Such an exhaustive enumeration tends to be very time-consuming because both, the search space and the output size, can be exponential in the size of the input database. The running time as well as the semantic significance of the produced output depend on the user-specified frequency parameter. Thus it is of great value to know as much as possible about the results of an exponential time pattern mining algorithm prior to its application. This knowledge can be used to readjust the frequency parameter and thus improve performance and semantic value of the mining algorithm.

For that purpose frequent sets of maximum cardinality resp. infrequent sets of minimum cardinality can be used. Many mining algorithms tend to run exponentially long in the cardinality of a longest pattern, i.e. the size of a maximum frequent set and for level-wise algorithms the size of a minimum infrequent set determines the level where pruning starts. So knowing either of the two would allow to upper bound the running time resp. skip initial search levels. In terms of result quality both indicate whether the chosen frequency threshold provides a significant gain of information for the resulting patterns. If for instance the

V. Corruble, M. Takeda, and E. Suzuki (Eds.): DS 2007, LNAI 4755, pp. 68–77, 2007.

minimum cardinality of an infrequent set is 18 in a database containing 20 items this is an indication for a weak parameter choice.

On the one hand, both optimization problems are **NP**-hard. For the maximization problem this was shown by Gunopulos et al. in [10]. For the minimization problem this is shown in Section 3 of this paper. On the other hand, computing approximate solutions would suffice for the described motivations. In this paper I show that not even a reasonable approximation algorithm for a maximum frequent set is likely to exist based on recent results from computational complexity [12], while for a minimum infrequent set a simple greedy algorithm reaches a logarithmic approximation factor. By another recent complexity result [8] this factor cannot be improved substantially. Note that in contrast to approaches that aim at approximating the set of *all* frequent sets (like in [1]) we consider different problems each aiming to compute only *one* set. To the best of my knowledge this is the first investigation on the approximability of these problems.

The rest of the paper is organized as follows: Section 2 introduces basic definitions and notations. In Section 3, the two optimization problems are defined formally and their **NP**-hardness is discussed. Section 4 points out the hardness of approximating the maximization problem, while Section 5 proves the logarithmic performance of the greedy algorithm for the minimization problem. Finally, Section 6 concludes with a summary and ideas for possible future work.

2 Preliminaries

A **hypergraph** is a triple $(V, \mathcal{H}, \mu)$ with V a finite set called **ground set**, $\mathcal{H} \subseteq 2^V$ a family whose elements are called **hyperedges**, and $\mu : \mathcal{H} \to \mathbb{N}$ a mapping representing the multiplicity of each hyperedge. So $\mathcal{H}$ can be seen as a multiset, and thus we mean by its cardinality $|\mathcal{H}|$ the sum $\sum_{H \in \mathcal{H}} \mu(H)$. For the purpose of computational problems we assume a hypergraph to be given as incidence matrix, and thus define $\text{size}((V, \mathcal{H}, \mu)) = |V||\mathcal{H}|$ as the input size. If $\mu(H) = 1$ for all $H \in \mathcal{H}$ we omit μ and $(V, \mathcal{H})$ is called proper.

A **graph** is a hypergraph $G = (V, E)$ with $|e| = 2$ for all $e \in E$. The elements of V are called **vertices**, and the elements of E are called **edges**[1]. G is called **bipartite** if V can be partitioned into V_1, V_2 such that all edges are of the form $\{v, w\}$ with $v \in V_1$ and $w \in V_2$. A graph of this form is denoted by (V_1, V_2, E). A set of vertices $X = X_1 \cup X_2$ with $X_1 \subseteq V_1, X_2 \subseteq V_2$ is denoted by (X_1, X_2) and is called a **bipartite clique** if for all $x_1 \in X_1$ and all $x_2 \in X_2$ there is an edge $\{x_1, x_2\} \in E$. It is called **balanced** if $|X_1| = |X_2|$. The size of a balanced bipartite clique (X_1, X_2) is $|X_1| = |X_2|$.

An **optimization problem** is a computational problem formally given by a 4-tuple $P = (X, (S_x)_{x \in X}, c, \text{goal})$ with a set of instances X, a set of feasible solutions S_x for all instances, a target function $c : \bigcup_{x \in X} S_x \to \mathbb{N}$, and goal $\in \{\min, \max\}$. The task is then, given an instance $x \in X$, compute a feasible

[1] In this paper we do not consider graphs with parallel edges or loops, i.e. edges with only one element.

solution $y \in S_x$ with $c(y) = \text{goal}\{c(y') : y' \in S_x\}$. If goal $=$ min, P is called a **minimization problem**. If goal $=$ max, P is called a **maximization problem**.

As examples consider the following two well known **NP**-hard optimization problems (see [9]):

MAX BALANCED CLIQUE is the following maximization problem: *Given* a bipartite graph G, *compute* a balanced bipartite clique in G of maximum cardinality. Here the instances are bipartite graphs, the feasible solutions for a graph G are balanced bipartite cliques in G, and the target function maps a balanced bipartite clique (X, Y) to its size $|X|$.

MIN SET COVER is the following minimization problem: *Given* a hypergraph $(V, \mathcal{H})$ with $\bigcup \mathcal{H} = V$, *compute* a family $\mathcal{H}' \subseteq \mathcal{H}$ of minimum cardinality covering V, i.e., $\bigcup \mathcal{H}' = V$.

Let $P = (X, (S_x)_{x \in X}, c, \text{goal})$ be an optimization problem. A deterministic algorithm $\mathcal{A}$ for P can be thought of as a mapping from the instances X to the set of all possible outputs $\bigcup_{x \in X} S_x$. Then $\mathcal{A}$ is called an **α-approximation algorithm** for P with $\alpha : X \to \mathbb{R}_{\geq 1}$ if for all $x \in X$ with $\text{goal}\{c(y) : y \in S_x\} = \text{OPT}$ it holds that $\mathcal{A}(x) \in S_x$, i.e., the algorithm produces only feasible solutions, $\mathcal{A}$ runs in polynomial time, and

$$\frac{1}{\alpha(x)}\text{OPT} \leq c(\mathcal{A}(x)) \leq \alpha(x)\text{OPT} \ .$$

For such an algorithm we say that $\mathcal{A}$ approximates P within a factor of α. If $\alpha(x) \equiv 1$, $\mathcal{A}$ solves the problem exactly. Note that the first inequality applies only to maximization problems, while the second applies only to minimization problems. Since we require $\mathcal{A}$ to produce always feasible solutions, it holds that $\mathcal{A}(x) \leq \text{OPT}$ in case goal=max and $\text{OPT} \leq \mathcal{A}(x)$ in case goal=min.

In *frequent set mining* (or frequent itemset mining) [2] the input is a hypergraph $D = (I, \mathcal{T}, \mu)$ called dataset and a positive integer $t \in \{1, \ldots, |\mathcal{T}|\}$ called **frequency threshold**. Sometimes the elements of I are called items and the elements of $\mathcal{T}$ are called transactions. For $X \subseteq I$ the **support set** of X is defined as

$$\mathcal{T}[X] = \{T \in \mathcal{T} \colon X \subseteq T\} \ .$$

X is called **t-frequent** in D if $|\mathcal{T}[X]| \geq t$.

3 Problems and Hardness of Exact Solutions

We are now ready to give a formal definition of the problems of interest: *Given* a hypergraph $(I, \mathcal{T}, \mu)$ and a frequency threshold $t \in \{1, \ldots, |\mathcal{T}|\}$ we define

MAX FREQUENT SET as the maximization problem to *compute* a t-frequent set $X \subseteq I$ of maximum cardinality and

MIN INFREQUENT SET as the minimization problem to *compute* a set $X \subseteq I$ of minimum cardinality that is not t-frequent.

Remark 1. In Section 1 we only discussed the use of the maximum resp. the minimum *cardinality* of a frequent resp. infrequent set. Here we require the *construction* of an actual set in each problem. However, these two tasks are polynomially equivalent. In particular a maximum frequent set can be constructed by iteratively trying to remove an element and then checking whether the maximum cardinality has changed.

Next we recall the construction used in [10] to prove hardness of MAX FREQUENT SET. In Section 4 we will reuse this construction, which is a transformation from the **NP**-hard MAX BALANCED CLIQUE problem to MAX FREQUENT SET that uses a canonical correspondence between hypergraphs and bipartite graphs:

For a given bipartite graph $G = (V, U, E)$ construct a hypergraph $D = (V, \mathcal{T}, \mu)$ with

$$\mathcal{T} = \{\Gamma(u) : u \in U\}$$
$$\mu \colon T \mapsto |\{u \in U \colon \Gamma(u) = T\}|$$

where $\Gamma(u)$ denotes the set of all neighbors of u, i.e., $\Gamma(u) = \{v \in V \colon \{v, u\} \in E\}$. Note that size$(D) \leq$ size(G). Furthermore, the maximum cardinality of a balanced bipartite clique in G is the maximum t such that there is a t-frequent set X in D with $|X| \geq t$, which can easily be computed from D with an algorithm solving MAX FREQUENT SET. This implies:

Theorem 1 (Gunopulos et al. [10]). MAX FREQUENT SET *is* **NP**-*hard.*

To analyze MIN INFREQUENT SET we define the following generalized version of MIN SET COVER:

MIN GENERAL SET COVER is the following minimization problem: *Given* a hypergraph $(V, \mathcal{H})$ and a positive integer $p \in \{0, \ldots, |V| - 1\}$, *compute* a minimum family of hyperedges $\mathcal{H}'$ covering at least $|V| - p$ elements of V, i.e., $|V \setminus \bigcup \mathcal{H}'| \leq p$.

MIN GENERAL SET COVER contains the **NP**-hard problem MIN SET COVER as a special case ($p = 0$), and thus it is itself **NP**-hard. Moreover, we have the following equivalence:

Theorem 2. MIN INFREQUENT SET *is polynomially equivalent to* MIN GENERAL SET COVER.

Proof. Construct a polynomial transformation f from MIN GENERAL SET COVER to MIN INFREQUENT SET by transposing the given incidence matrix and changing 0-entries to 1-entries and vice versa. The frequency parameter t is set to $p + 1$. Note that because of the parameter ranges of t and p this mapping is bijective. For an instance $((V, \mathcal{H}), p)$ this results in:

$$f : ((V, \mathcal{H}), p) \mapsto ((\mathcal{H}, \overline{\mathcal{V}}, \mu), p + 1)$$
$$\overline{\mathcal{V}} = \{\mathcal{H} \setminus \mathcal{H}[\{v\}] \colon v \in V\}$$
$$\mu : \mathcal{H}' \mapsto |\{v \in V \colon \mathcal{H}' = \mathcal{H} \setminus \mathcal{H}[\{v\}]\}| \ .$$

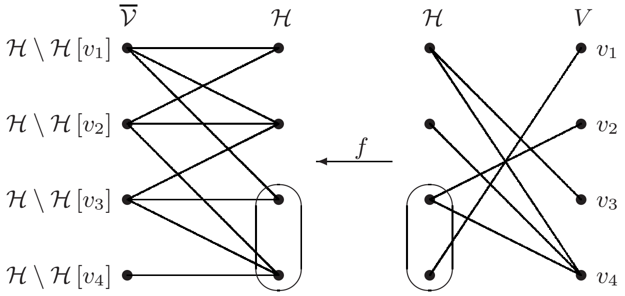
$\overline{\mathcal{V}}$
$\mathcal{H}$
$\mathcal{H}$
V
$\mathcal{H} \setminus \mathcal{H}[v_1]$
$\mathcal{H} \setminus \mathcal{H}[v_2]$
$\mathcal{H} \setminus \mathcal{H}[v_3]$
$\mathcal{H} \setminus \mathcal{H}[v_4]$
f
v_1
v_2
v_3
v_4

Fig. 1. Construction used to proof Theorem 2. Here the hypergraphs are drawn as bipartite graphs (see proof of Theorem 1) with hyperedges on the left side and ground set on the right side. The marked set is 2-infrequent in the left hypergraph and covers all but 1 element in the right.

So the original hyperedges act as items and every element v of the original ground set becomes a hyperedge, that contains all the sets, which v is not an element of (in the original MIN GENERAL SET COVER instance). Now we claim that an $(p+1)$-infrequent set in $(\mathcal{H}, \overline{\mathcal{V}})$ corresponds to a set of hyperedges covering at least all but p elements of $(V, \mathcal{H})$ and vice versa (see Fig. 1). For a subset $\mathcal{H}' \subseteq \mathcal{H}$ it holds that

$$\begin{aligned} \mathcal{H}' \ (p+1)\text{-infrequent in } (\mathcal{H}, \overline{\mathcal{V}}) &\Leftrightarrow |\overline{\mathcal{V}}[\mathcal{H}']| < p+1 \\ &\Leftrightarrow |\{v \in V \colon \mathcal{H} \setminus \mathcal{H}[\{v\}] \supseteq \mathcal{H}'\}| < p+1 \\ &\Leftrightarrow |\{v \in V \colon \forall H \in \mathcal{H}' \ v \notin H\}| < p+1 \\ &\Leftrightarrow |V \setminus \bigcup \mathcal{H}'| < p+1 \\ &\Leftrightarrow \mathcal{H}' \text{ covers at least all but } p \text{ elements} \ . \end{aligned}$$

So an infrequent set of size k corresponds to a subfamily of size k covering sufficient many elements and vice versa. Furthermore, f is a bijection implying polynomial equivalence. □

This implies the main result of this section completing our problem introduction:

Corollary 3. MIN INFREQUENT SET *is* **NP**-*hard.*

4 Hardness of Approximating a Maximum Frequent Set

Since MAX FREQUENT SET is **NP**-hard, the next step is to ask for an approximation algorithm. Proving negative results for the approximation of hard problems has been very successful in recent years. New results have in common that they use so called 'probabilistically checkable proofs' [4] as a characterization of **NP**. As indicated by the proof of Theorem 1 the following result proved by Khot [12] for MAX BALANCED CLIQUE is of particular importance for our purpose:

Unless there are probabilistic algorithms with an arbitrary small exponential time complexity for all problems in **NP** there is no polynomial approximation scheme for MAX BALANCED CLIQUE, i.e., the infimum of all constants k such that there is a k-approximation algorithm for MAX BALANCED CLIQUE is bounded away from 1. It was known before that such a result, once achieved, can be boosted via derandomized graph products (introduced in [3]). So that the result of Khot implies in fact:

Theorem 4 (Khot [12]). *Unless for all $\epsilon > 0$ and all decision problems in* **NP** *there is a probabilistic algorithm $\mathcal{A}$ accepting a YES-instance resp. rejecting a NO-instance of size n with probability at least $2/3$ in time 2^{n^ϵ} the following holds: There is a constant $\delta_{BC} > 0$ such that there is no algorithm approximating* MAX BALANCED CLIQUE *within a factor of size$(x)^{\delta_{BC}}$ for instances x.*

Now suppose there is an algorithm $\mathcal{A}$ approximating MAX FREQUENT SET within a factor of $\alpha(\text{size}(x))$ for instances x. Then one can construct a hypergraph D from a given bipartite graph G as for Theorem 1 and find t_{APX} the maximum $t \in \{1, \ldots, |\mathcal{T}|\}$ for which $|\mathcal{A}(D,t)| \geq t$ by running $\mathcal{A}$ at most $|\mathcal{T}|$ times. Let (X, Y) be a maximum balanced bipartite clique in $G = (V, U, E)$ with size t_{OPT}. Any set of transactions corresponding to a subset $Y' \subseteq Y$ contains the t_{OPT} items corresponding to X—in particular those with $|Y'| = t_{\text{OPT}}/\alpha(\text{size}(D)) = t^*$. This implies for the maximum cardinality of a t^*-frequent set in D, denoted as $\text{mfs}(D, t^*)$,

$$\text{mfs}(D, t^*) \geq t_{\text{OPT}} \Rightarrow |\mathcal{A}(D, t^*)| \geq t_{\text{OPT}}/\alpha(\text{size}(D)) = t^* \ .$$

But then $t_{\text{APX}} \geq t^* = t_{\text{OPT}}/\alpha(\text{size}(D)) \geq t_{\text{OPT}}/\alpha(\text{size}(G))$, because the transformed instance is of equal or smaller size. Since all necessary computations can be performed in polynomial time, we have a polynomial algorithm approximating MAX BALANCED CLIQUE within a factor of $\alpha(\text{size}(x))$ for instances x and hence

Corollary 5. *Under the same assumptions as in Theorem 4 with the same constant $\delta_{BC} > 0$ there is no algorithm approximating* MAX FREQUENT SET *within a factor of size$(x)^{\delta_{BC}}$ for instances x.*

Although stronger than $\mathbf{P} \neq \mathbf{NP}$ the stated complexity assumption is still widely believed and thus we have a strong indication that there is no algorithm for MAX FREQUENT SET with a reasonable approximation factor.

5 Greedy Approximation of a Minimum Infrequent Set

The transformation in Theorem 2 maps instances of MIN GENERAL SET COVER to instances of MIN INFREQUENT SET with the same optimum value and vice versa and there is also a bijection between feasible solutions. So an approximation algorithm for either one of the two problems will grant the same approximation factor for the other. To analyze the approximability of the two problems we will use another related coverage problem:

MAX COVERAGE is the following maximization problem: *Given* a hypergraph $(V,\mathcal{H})$ and a positive integer k, *compute* a family $\mathcal{H}' \subseteq \mathcal{H}$ of k hyperedges covering a maximum number of elements.

Using the known fact that the approximation ratio of the greedy algorithm for this problem is $(1-e^{-1})$ (see for instance [6]), one can analyze the approximation performance of the greedy approach for MIN GENERAL SET COVER.

Theorem 6. MIN GENERAL SET COVER *can be approximated in polynomial time within a factor of* $\lceil \ln(|V|-p)\rceil+1$ *for instances* $((V,\mathcal{H}),p)$.

Proof. The following algorithm uses the greedy algorithm $\mathcal{G}$ for MAX COVERAGE, to achieve the desired approximation rate for MIN GENERAL SET COVER. Denote with n the number of elements $|V|$ and with $\text{gsc}(V,\mathcal{H},p)$ the minimum cardinality of a hyperedge set covering at least $n-p$ elements.

```
1. $i \leftarrow 1,\ \mathcal{S} \leftarrow \emptyset,\ V_1 \leftarrow V,\ \mathcal{H}_1 \leftarrow \mathcal{H}$
2. while $|V_i| > p$ do
3.    $k_i \leftarrow \min\{j : |\bigcup \mathcal{G}(V_i,\mathcal{H}_i,j)| \geq e^{1-i}(1-\frac{1}{e})(|V|-p)\}$
4.    $\mathcal{H}_\Delta \leftarrow \mathcal{G}(V_i,\mathcal{H}_i,k_i)$
      $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{H}_\Delta,\ \mathcal{H}_{i+1} \leftarrow \mathcal{H}_i \setminus \mathcal{H}_\Delta,\ V_{i+1} \leftarrow V_i \setminus \bigcup \mathcal{H}_\Delta$
5.    $i \leftarrow i+1$
6. return $\mathcal{S}$
```

Obviously $\mathcal{S}$ covers at least $n-p$ elements after termination. We claim that also $|\mathcal{S}| \leq (\lceil \ln(n-p)\rceil+1)\,\text{gsc}(V,\mathcal{H},p)$. To see this, we first analyze the number of iterations and then the number of hyperedges added to the $\mathcal{S}$ in every iteration.

(i) The algorithm terminates after at most $\lceil \ln(n-p)\rceil+1$ iterations.
Proof of (i):
First show $|V_i| \leq p+e^{1-i}(n-p)$ by induction on i. For $i=1$ this is true, because $|V|=|V_1|=n$. Now assume that $|V_i| \leq p+e^{1-i}(n-p)$ for a given i. In line 3 k_i is chosen such that at least $e^{1-i}(1-e^{-1})(n-p)$ elements will be covered. So

$$|V_{i+1}| \leq p + \frac{1}{e^{i-1}}(n-p) - \frac{e-1}{e^i}(n-p) = p + \frac{1}{e^i}(n-p) \ .$$

Since the algorithm terminates when $|V_i| \leq p$ (and $|V_i|$ cannot be fractional), it is for the number of iterations t:

$$t \leq \min\{i \in \mathbb{N} : e^i > n-p\} = \lceil \ln(n-p)\rceil+1$$

$(i)\Box$

(ii)For all iterations i it is $k_i \leq \text{gsc}(V,\mathcal{H},p)$.
Proof of (ii):
By definition there is an optimum cover $\mathcal{O} \subseteq \mathcal{H}$ with

$$|\bigcup \mathcal{O}| \geq n-p \text{ and } |\mathcal{O}| = \text{gsc}(V,\mathcal{H},p) \ .$$

So $\mathcal{O}$ covers all but p elements. Let $\text{mc}(V',\mathcal{H}',k)$ denote the maximum number of elements one can cover with k hyperedges in $(V',\mathcal{H}')$. Since in iteration i it is

Algorithm 1. $(\lceil \ln(|\mathcal{T}| - t)\rceil + 1)$-approximation for MIN INFREQUENT SET

Require: Dataset $D = (I, \mathcal{T}, \mu)$ and frequency threshold t
Ensure: X infrequent and $|X| \leq (\lceil \ln(|\mathcal{T}| - t)\rceil + 1)\mathrm{OPT}$, with OPT the minimum cardinality of a set that is not t-frequent in D

```
1. $X \leftarrow \emptyset$
2. while $|\mathcal{T}| \geq t$ do
3.    $i \leftarrow i \in I$ with $|\mathcal{T}[\{i\}]| = \min\{|\mathcal{T}[\{i'\}]| \colon i' \in I\}$
4.    $X \leftarrow X \cup \{i\}$
5.    $I \leftarrow I \setminus \{i\}$
6.    $\mathcal{T} \leftarrow \mathcal{T}[\{i\}]$
7. return $X$
```

$|V_i| \leq p + e^{1-i}(n-p)$, $|\mathcal{O}|$ elements can still cover at least $e^{1-i}(n-p)$ elements. It follows

$$\mathrm{mc}(V_i, \mathcal{H}_i, \mathrm{gsc}(V, \mathcal{H}, p)) \geq e^{1-i}(n-p)$$
$$\Rightarrow |\bigcup \mathcal{G}(V_i, \mathcal{H}_i, \mathrm{gsc}(V, \mathcal{H}, p))| \geq (1 - \frac{1}{e})e^{1-i}(n-p)$$

and because k_i is selected in line 3 as the minimum number satisfying this

$$\Rightarrow k_i \leq \mathrm{gsc}(V, \mathcal{H}, p) \ .$$

$(ii)\square$

Since k_i sets are added to $\mathcal{S}$ in every iteration i, it follows from (i) and (ii) that $|\mathcal{S}| \leq (\lceil \ln(n-p)\rceil + 1)\mathrm{gsc}(V, \mathcal{H}, p)$. The polynomial running time is obvious, because the polynomial time greedy algorithm is called in every iteration at most $|\mathcal{H}|$ times. $\square$

Remark 2. The formulation of the algorithm in the above proof was tailor-made for the surrounding analysis. In fact it only selects remaining hyperedges covering a maximum number of remaining elements and thus the simple greedy strategy stopping, when all but p elements are covered, will select the same hyperedges or possibly even some less.

Algorithm 1 takes this into account and incorporates the transformation between MIN INFREQUENT SET and MIN GENERAL SET COVER. Note that this transformation switches the roles of ground set and hyperedges so that the resulting approximation factor does not depend on the number of items but on the number of transactions. This constitutes the following result:

Corollary 7. MIN INFREQUENT SET *can be approximated within a factor of* $\lceil \ln(|\mathcal{T}| - t)\rceil + 1$ *for instances* $(I, \mathcal{T}, \mu)$, t.

The approximation ratio achieved above is close to optimal. Otherwise, since MIN GENERAL SET COVER contains MIN SET COVER as a special case for $p = 0$, a better ratio would imply the existence of subexponential time algorithms with extremely small exponents for every problem in **NP** by the following theorem:

Theorem 8 (Feige [8]). *For all $\epsilon > 0$ there is no algorithm approximating* MIN SET COVER *within a factor of $(1-\epsilon)\ln|V|$ for instances $(V, \mathcal{H})$, unless for all problems in* **NP** *there is an algorithm running in time $n^{O(\log\log n)}$ for instances of size n.*

6 Discussion

In this paper, we have analyzed the algorithmical tasks to approximate a maximum frequent resp. a minimum infrequent set. This investigation is motivated by the need for an efficient parameter evaluation procedure that can be applied before a possibly exponential time pattern mining algorithm. We turned to approximation algorithms because both problems are **NP**-hard. In case of the maximization problem this was well-known. In case of the minimization problem we proved this hardness by showing it to be equivalent to a generalized version of the MIN SET COVER problem.

Using recent results from computational complexity we have argued that a non-trivial approximation algorithm for MAX FREQUENT SET is unlikely to exist. For MIN INFREQUENT SET we gave a polynomial time greedy algorithm, which was proven to compute an infrequent set of cardinality smaller than $\lceil\ln(m-t)\rceil+1$ times the minimum cardinality of an infrequent set for instances with frequency threshold t and m transactions. Slavík proved in [14] that the approximation ratio of the greedy algorithm for MIN SET COVER can in fact be bounded by $\ln n - \ln\ln n + 0.79$. It is likely that his tight analysis can be transfered to MIN GENERAL SET COVER, which is a task for possible future work. The fact that the approximation factor depends on the number of transactions and not on the number of items indicates that the algorithm is useful for gene expression data [13], which can contain up to 100,000 items but typically only about 1000 transactions. In general, knowing the approximation factor allows valuable conclusions. If the cardinality of the returned set is c this implies that all sets of cardinality smaller than $c/(\lceil\ln(m-t)\rceil+1)$ are frequent. In turn, this provides a lower bound on the number of frequent sets and for level-wise algorithms determines an earliest level where pruning can occur so that search need not to be started before this level.

Other important characteristics that can be used for parameter evaluation are the number of frequent resp. closed or maximal frequent sets resulting from a given parameter, all of which are hard *counting* problems [15,10]. It is an interesting question whether the positive results from computing the permanent of a 0-1 matrix can be transfered to those problems. For 0-1-PERMANENT the existence of a fully polynomial randomized approximation scheme has been shown [11]. Another question is, how quick parameter evaluation can be done in other domains with similar problems as frequent set mining (exponential output size and even greater search space). Examples for such domains are pattern mining tasks with structured data like sequences or graphs.

Acknowledgments. I would like to thank Prof. Bhaskar DasGupta who pointed me towards the maximum coverage approach.

References

1. Afrati, F., Gionis, A., Mannila, H.: Approximating a collection of frequent sets. In: ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (2004)
2. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Advances in Knowledge Discovery and Data Mining, pp. 307–328. AAAI/MIT Press, Cambridge (1996)
3. Alon, N., Feige, U., Wigderson, A., Zuckerman, D.: Derandomized graph products. Computational Complexity 5(1), 60–75 (1995)
4. Arora, S., Safra, S.: Probabilistic checking of proofs: A new characterization of NP. J. ACM 45(1), 70–122 (1998)
5. Brin, S., Motwani, R., Silverstein, C.: Beyond market baskets: Generalizing association rules to correlations. In: SIGMOD Conference, pp. 265–276 (1997)
6. Cornuejols, G., Fisher, M., Nemhauser, G.: Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms. Management Science (23), 789–810 (1977)
7. Dong, G., Li, J.: Efficient mining of emerging patterns: Discovering trends and differences. In: KDD, pp. 43–52 (1999)
8. Feige, U.: A threshold of ln n for approximating set cover. J. ACM 45(4), 634–652 (1998)
9. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, (1979) ISBN 0-7167-1044-7
10. Gunopulos, D., Khardon, R., Mannila, H., Saluja, S., Toivonen, H., Sharm, R.S.: Discovering all most specific sentences. ACM Trans. Database Syst. 28(2), 140–174 (2003)
11. Jerrum, M., Sinclair, A.: Approximating the permanent. SIAM J. Comput. 18(6), 1149–1178 (1989)
12. Khot, S.: Ruling out ptas for graph min-bisection, densest subgraph and bipartite clique. In: FOCS, pp. 136–145 (2004)
13. Pan, F., Cong, G., Tung, A.K.H., Yang, J., Zaki, M.J.: Carpenter: Finding closed patterns in long biological datasets. In: ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, ACM Press, New York (2003)
14. Slavík, P.: A tight analysis of the greedy algorithm for set cover. Journal of Algorithms 25, 237–254 (1997)
15. Valiant, L.G.: The complexity of computing the permanent. Theor. Comput. Sci. 8, 189–201 (1979)