Approximation of Frequency Queries by Means of Free-Sets

Jean-François Boulicaut, Artur Bykowski, and Christophe Rigotti

Laboratoire d'Ingénierie des Systèmes d'Information
INSA Lyon, Bâtiment 501
F-69621 Villeurbanne Cedex, France
{Jean-François.Boulicaut,Artur.Bykowski,Christophe.Rigotti}@insa-lyon.fr

Abstract. Given a large collection of transactions containing items, a basic common data mining problem is to extract the so-called frequent itemsets (i.e., set of items appearing in at least a given number of transactions). In this paper, we propose a structure called free-sets, from which we can approximate any itemset support (i.e., the number of transactions containing the itemset) and we formalize this notion in the framework of ϵ-adequate representation [10].We show that frequent free-sets can be efficiently extracted using pruning strategies developed for frequent itemset discovery, and that they can be used to approximate the support of any frequent itemset. Experiments run on real dense data sets show a significant reduction of the size of the output when compared with standard frequent itemsets extraction. Furthermore, the experiments show that the extraction of frequent free-sets is still possible when the extraction of frequent itemsets becomes intractable. Finally, we show that the error made when approximating frequent itemset support remains very low in practice.

1 Introduction

Several data mining tasks (e.g., association rule mining [1]) are based on the evaluation of frequency queries to determine how often a particular pattern occurs in a large data set. We consider the problem of frequency query evaluation, when patterns are itemsets, in dense data sets[1] like, for instance in the context of census data analysis [4] or log analysis [8]. In these important but difficult cases, there is a combinatorial explosion of the number of frequent itemsets and computing the frequency of all of them turns out to be intractable. In this paper, we present an efficient technique to approximate closely the result of the frequency queries, and formalize it within the ϵ-adequate representation framework [10]. Intuitively an ϵ-adequate representation is a representation of data that can be substituted to another representation to answer the same kind of queries, but eventually with some lost of precision (bound by the ϵ parameter). First evidences of the practical interest of such representations has been given in [10,5].

[1] e.g., data sets containing many strong correlations.

D.A. Zighed, J. Komorowski, and J. Żytkow (Eds.): PKDD 2000, LNAI 1910, pp. 75–85, 2000.

In this paper, we propose a new ϵ-adequate representation for the frequency queries. This representation called *free-sets*, is more condensed than the ϵ-adequate representation based on itemsets [10]. The key intuition of the free-set representation is the following. Let A, B, C, D represent binary attributes in a database. If we know that the association rule $A, B, C \Rightarrow D$ is nearly an exact rule (i.e., it has only a few exceptions), then we can approximate the frequency of itemset $\{A, B, C, D\}$ using the frequency of $\{A, B, C\}$. Moreover, we can approximate the frequency of any itemset X such that $\{A, B, C, D\} \subseteq X$ by the frequency of $X \setminus \{D\}$. We call free-set an itemset Y such that the items in Y can not be used to form a nearly exact association rule. We show that *frequent* free-sets are an ϵ-adequate representation for frequency queries that can be extracted efficiently, even on dense data sets. We also show that the error made when approximating itemset frequency using frequent free-sets remains very low in practice.

Organization of the Paper. In the next section we introduce preliminary definitions used in this paper. In Section 3, we present the notion of free-set, and show that it can be used as an ϵ-adequate representation for the frequency queries. Due to space limitation proofs are omitted. In section 4, we present an algorithm to extract the frequent free-sets. In Section 5, we give practical evidence that frequent free-sets can be extracted efficiently and that the estimation of the supports of frequent itemsets using frequent free-sets leads in practice to very low errors. We review related work in Section 6. Finally, we conclude with a summary and directions for future work.

2 Preliminary Definitions

When applicable, we use the notational conventions and definitions from [10,11].

2.1 Frequent Sets and Association Rules

In this section we recall standard definitions.

Definition 1 (binary database). Let R be a set of symbols called items. A *row* (also called *transaction*) is a subset of R. A *binary database r over R* is a multiset of transactions.

Definition 2 (support and frequency). We note $\mathcal{M}(r, X) = \{t \in r | X \subseteq t\}$ the multiset of rows matched by the itemset X and $Sup(r, X) = |\mathcal{M}(r, X)|$ the *support* of X in r, i.e., the number of rows matched by X. The *frequency* of X in r is $Sup(r, X)/|r|$. Let σ be a frequency threshold, $Freq(r, \sigma) = \{X | X \subseteq R$ and $Sup(r, X)/|r| \geq \sigma\}$ is the set of all σ-frequent itemsets in r.

For notational convenience, we also need the following specific definition.

Definition 3 (frequent sets). $FreqSup(r, \sigma)$ is the set of all pairs containing a frequent itemset and its support, i.e., $FreqSup(r, \sigma) = \{\langle X, Sup(r, X)\rangle | X \subseteq R$ and $Sup(r, X)/|r| \geq \sigma\}$.

2.2 ϵ-Adequate Representation

Definition 4 (ϵ-adequate representation [10]). Let $\mathcal{S}$ be a class of structures. Let $\mathcal{Q}$ be a class of queries for $\mathcal{S}$. The value of a query $Q \in \mathcal{Q}$ on a structure $s \in \mathcal{S}$ is assumed to be a real number in $[0,1]$ and is denoted $Q(s)$. An ϵ-adequate representation for $\mathcal{S}$ w.r.t. a class of queries $\mathcal{Q}$, is a class of structures $\mathcal{C}$, a representation mapping $rep : \mathcal{S} \to \mathcal{C}$ and a query evaluation function $m : \mathcal{Q} \times \mathcal{C} \to [0,1]$ such that $\forall Q \in \mathcal{Q}, \forall s \in \mathcal{S}, |Q(s) - m(Q, rep(s))| \leq \epsilon$.

Example 1. An example of a class of structures is the set noted $\mathcal{DB}_R$ of all possible binary databases over a set of items R. An interesting query class is $\mathcal{Q}_R$, the set of all queries retrieving the frequency of an itemset $\subseteq R$. If we denote Q_X the query in $\mathcal{Q}_R$ asking for the frequency of itemset X then $\mathcal{Q}_R = \{Q_X | X \subseteq R\}$ and the value of Q_X on a database instance $r \in \mathcal{DB}_R$ is defined by $Q_X(r) = Sup(r, X)/|r|$.
An example of ϵ-adequate representation for $\mathcal{DB}_R$ w.r.t. $\mathcal{Q}_R$ is the representation of $r \in \mathcal{DB}_R$ by means of $Freq(r, \epsilon)$. The corresponding $rep, \mathcal{C}$ and m are as follows. $\forall r \in \mathcal{DB}_R, rep(r) = FreqSup(r, \epsilon)$, $\mathcal{C} = \{rep(r) | r \in \mathcal{DB}_R\}$, $\forall Q_X \in \mathcal{Q}_R, \forall c \in \mathcal{C}$, if $\exists \langle X, \alpha \rangle \in rep(r)$ then $m(Q_X, c) = \alpha/|r|$ else $m(Q_X, c) = 0$. It is straightforward to see that this is an ϵ-adequate representation for $\mathcal{DB}_R$ w.r.t. $\mathcal{Q}_R$ since $\forall Q_X \in \mathcal{Q}_R, \forall r \in \mathcal{DB}_R, |Q_X(r) - m(Q_X, rep(r))| \leq \epsilon$.

Interesting ϵ-adequate representations are *condensed representations*, i.e., ϵ-adequate representations where structures have a smaller size than the original structures.

3 The Free-Sets as a Condensed Representation

Even though this paper do not concerned directly with *association rules*, we need the following definitions to introduce the concept of free-set in a concise way.

Definition 5 (δ-strong rule). Let R be a set of items, an *association rule* based on R is an expression of the form $X \Rightarrow Y$, where $X, Y \subseteq R$ and $X \cap Y = \emptyset$. A *δ-strong rule*[2] in a binary database r over R is an association rule $X \Rightarrow Y$ such that $Sup(r, X) - Sup(r, X \cup Y) \leq \delta$, i.e., the rule is violated in no more than δ rows.

In this definition, δ is supposed to have a small value, so a δ-strong rule is intended to be a rule with very few exceptions.

[2] Stemming from the notion of *strong rule* of [15]

3.1 Free-Sets

Definition 6 (δ-free-set). Let r be a binary database over R, $X \subseteq R$ is a *δ-free-set* w.r.t. r if and only if there is no δ-strong rule based on X in r. The set of all δ-free-sets w.r.t. r is noted $Free(r, \delta)$.

Since δ is supposed to be rather small, informally, a free-set is a set of items such that its subsets (seen as conjunction of properties) are not related by any very strong positive correlation.
One of the most interesting properties of *freeness* is its *anti-monotonicity* w.r.t. itemset inclusion.

Definition 7 (anti-monotonicity). A property ρ is *anti-monotone* if and only if for all itemsets X and Y, $\rho(X)$ and $Y \subseteq X$ implies $\rho(Y)$.

The anti-monotonicity has been identified as a key property for efficient pattern mining [11,12,6], since it is the formal basis of a safe pruning criterion. Indeed, efficient frequent set mining algorithms like Apriori [1] make use of the (anti-monotone) property *"to be frequent"* for pruning.
The anti-monotonicity of freeness follows directly from the definition of free-set and is stated by the following theorem.

Theorem 1. *Let X be an itemset. For all $Y \subseteq X$ if $X \in Free(r, \delta)$ then $Y \in Free(r, \delta)$.*

3.2 Free-Sets as an ϵ-Adequate Representation

We show now that δ-free-sets can be used to answer frequency queries with a bounded error. The following lemma states that the support of any itemset can be approximated using the support of one of the free-sets.

Lemma 1. *Let r be a binary database over a set of items R, $X \subseteq R$ and $\delta \in [0, |r|]$, then there exists $Y \subseteq X$ such that $Y \in Free(r, \delta)$ and $Sup(r, Y) \geq Sup(r, X) \geq Sup(r, Y) - \delta|X|$.*

This lemma states that the support of an itemset X can be approximated using the support of one of the free-sets, but it remains to determine which free-set to use. We now show that this can be done by simply choosing among the free-sets included in X any free-set with a minimal support value. This is stated more formally by the following theorem.

Theorem 2. *Let r be a binary database over a set of items R, $X \subseteq R$ and $\delta \in [0, |r|]$, then for any $Y \subseteq X$ such that $Y \in Free(r, \delta)$ and $Sup(r, Y) = min(\{Sup(r, Z) | Z \subseteq X$ and $Z \in Free(r, \delta)\})$ we have $Sup(r, Y) \geq Sup(r, X) \geq Sup(r, Y) - \delta|X|$.*

In practice, computing the whole collection of δ-free-sets is often intractable. We show now that such an exhaustive mining can be avoided since an ϵ-adequate representation to answer frequency queries can be obtained if we extract only *frequent* free-sets together with a subset of the corresponding negative border [11].

Definition 8 (frequent free-set). Let r be a binary database over a set of items R, we noted $\mathit{FreqFree}(r, \sigma, \delta) = \mathit{Freq}(r, \sigma) \cap \mathit{Free}(r, \delta)$ the set of σ-frequent δ-free-sets w.r.t. r.

Let us adapt the concept of negative border from [11] to our context.

Definition 9 (negative border of frequent free-sets). Let r be a binary database over a set of items R, the negative border of $\mathit{FreqFree}(r, \sigma, \delta)$ is noted $\mathcal{B}d^-(r, \sigma, \delta)$ and is defined as follows: $\mathcal{B}d^-(r, \sigma, \delta) = \{X | X \subseteq R, X \notin \mathit{FreqFree}(r, \sigma, \delta) \wedge (\forall Y \subset X, Y \in \mathit{FreqFree}(r, \sigma, \delta))\}$.

Informally, the negative border $\mathcal{B}d^-(r, \sigma, \delta)$ consists of the smallest itemsets (w.r.t. set inclusion) that are not σ-frequent δ-free. Our approximation technique only needs a subset of the negative border $\mathcal{B}d^-(r, \sigma, \delta)$. This subset, noted $\mathit{Free}\mathcal{B}d^-(r, \sigma, \delta)$, is the set of all free-sets in $\mathcal{B}d^-(r, \sigma, \delta)$.

Definition 10. $\mathit{Free}\mathcal{B}d^-(r, \sigma, \delta) = \mathcal{B}d^-(r, \sigma, \delta) \cap \mathit{Free}(r, \delta)$

As in the case of an ϵ-adequate representation for $\mathcal{DB}_R$ w.r.t. $\mathcal{Q}_R$ using frequent itemsets (see Section 2.2), we need the free-sets and their supports.

Definition 11. $\mathit{FreqFreeSup}(r, \sigma, \delta)$ is the set of all pairs containing a frequent free-set and its support, i.e., $\mathit{FreqFreeSup}(r, \sigma, \delta) = \{\langle X, \mathit{Sup}(r, X)\rangle | X \in \mathit{FreqFree}(r, \sigma, \delta)\}$.

We can now define the ϵ-adequate representation w.r.t. the frequency queries.

Definition 12. The *frequent free-sets representation* w.r.t. σ, δ and a query class $\mathcal{Q} \subseteq \mathcal{Q}_R$, is defined by a class of structures $\mathcal{C}$, a representation mapping *rep* and a query evaluation function m, where $\forall r \in \mathcal{DB}_R$,
$\mathit{rep}(r) = \langle \mathit{FreqFreeSup}(r, \sigma, \delta), \mathit{Free}\mathcal{B}d^-(r, \sigma, \delta)\rangle$,
$\mathcal{C} = \{\mathit{rep}(r) | r \in \mathcal{DB}_R\}$,
$\forall Q_X \in \mathcal{Q}, \forall c \in \mathcal{C}$, if $\exists Y \in \mathit{Free}\mathcal{B}d^-(r, \sigma, \delta), Y \subseteq X$ then $m(Q_X, c) = 0$ else $m(Q_X, c) = \mathit{min}(\{\alpha | \exists Z \subseteq X, \langle Z, \alpha\rangle \in \mathit{FreqFreeSup}(r, \sigma, \delta)\})/|r|$.

Using this representation, the frequency of an itemset X is approximated as follows. If X has a subset Y which is free but not frequent then the frequency of X is considered to be 0. Otherwise we take the smallest support value among the supports of the subsets of X that are free and frequent.
We now establish that this representation is an ϵ-adequate representation for the following database class and query class.

Definition 13. $\mathcal{DB}_{R,s} = \{r | r \in \mathcal{DB}_R \text{ and } |r| \leq s\}$, i.e., the set of all binary databases having no more than s rows. $\mathcal{Q}_{R,n} = \{Q_X | X \subseteq R \text{ and } |X| \leq n\}$, i.e., the set of frequency queries on itemsets having no more than n items.

Theorem 3.
A frequent free-sets representation *w.r.t. σ, δ and a query class $\mathcal{Q}_{R,n}$ is an ϵ-adequate representation for $\mathcal{DB}_{R,s}$ w.r.t. $\mathcal{Q}_{R,n}$ where $\epsilon = \mathit{max}(\sigma, n\delta/s)$.*

4 Discovering All Frequent Free-Sets

In this section, we describe an algorithm, called MinEx , generating all frequent free-sets. For clarity, we omit the fact that it outputs their supports as well. Implementation issues are presented in Section 4.2.

4.1 The Algorithm – An Abstract Version

MinEx can be seen as an instance of the levelwise search algorithm presented in [11]. It explores the itemset lattice (w.r.t. set inclusion) levelwise, starting from singletons and stopping at the level of the largest frequent free-sets. More precisely, the collection of candidates is initialized to the collection of all sets of size 1 and then the algorithm iterates on candidate evaluation and larger candidate generation. At the i^{th} iteration of this loop, it scans the database to find out which candidates are frequent free-sets. Then, it generates candidates for the $i+1^{th}$ iteration, taking every set of size $i+1$ such that all proper subsets are frequent free-sets. The algorithm finishes when there is no more candidates.

Algorithm 1
Input: *r a binary database over a set of items R, σ and δ two thresholds.*
Output: $FreqFree(r, \sigma, \delta)$

```
1.  $\mathcal{C}_1 := \{\{A\} | A \in R\}$;
2.  $i := 1$;
3.  while $\mathcal{C}_i \neq \emptyset$ do
4.      $\mathcal{F}req\mathcal{F}ree_i := \{X | X \in \mathcal{C}_i$ and $X$ is a $\sigma$-frequent $\delta$-free-set$\}$;
5.      $\mathcal{C}_{i+1} := \{X | X \subseteq R$ and $\forall\, Y \subset X, Y \in \bigcup_{j \leq i} \mathcal{F}req\mathcal{F}ree_j\} \setminus \bigcup_{j \leq i} \mathcal{C}_j$;
6.      $i := i + 1$;
7.  od;
8.  output $\bigcup_{j<i} \mathcal{F}req\mathcal{F}ree_j$;
```

Using the correctness result of the levelwise search algorithm given in [11] the following theorem is straightforward.

Theorem 4 (Correctness). Algorithm MinEx computes the sets of all σ-frequent δ-free-sets.

4.2 Implementation Issues

We used techniques similar to the ones described in [2] for frequent itemsets mining. The candidate generation is made using a join-based function, and the itemset support counters are updated w.r.t. a row of the database using a *prefix-tree* data structure.

The key point that needs a new specific technique is the freeness test in step 4 of the algorithm. An efficient computation of this test can be done, based on the

following remark: Z is not a δ-free-set iff there exists $A \in Z$ and $X = Z \backslash \{A\}$ such that X is not δ-free or X is δ-free and $X \Rightarrow \{A\}$ is a δ-strong rule. Furthermore, the step 5 of the algorithm guarantees that if Z is a candidate, X, which is a subset of Z, must be δ-free. Therefore, during the i^{th} iteration, we might first compute the δ-strong rules of the form $X \Rightarrow \{A\}$ where $X \in \mathcal{F}req\mathcal{F}ree_i$ and $A \in R \setminus X$ and then use them to remove candidates in $\mathcal{C}_{i+1}$ that are not δ-free. Thus, at the begining of an iteration, only free-sets are candidates.
This is incorporated in the algorithm by replacing steps 4 and 5 with the following steps:

```
4.1    $\mathcal{F}req\mathcal{F}ree_i := \{X | X \in \mathcal{C}_i$ and $X$ is a $\sigma$-frequent$\}$;
4.2    $\mathcal{N}ot\mathcal{F}ree_{i+1} := \{Z | Z = X \cup \{A\}$ where $X \in \mathcal{F}req\mathcal{F}ree_i, A \in R \setminus X$
                 and $X \Rightarrow \{A\}$ is a $\delta$-strong rule $\}$ ;
5.1    $\mathcal{C}^g_{i+1} := \{X | X \subseteq R$ and $\forall\ Y \subset X, Y \in \bigcup_{j \leq i} \mathcal{F}req\mathcal{F}ree_j\} \setminus \bigcup_{j \leq i} \mathcal{C}_j$;
5.2    $\mathcal{C}_{i+1} := \mathcal{C}^g_{i+1} \setminus \mathcal{N}ot\mathcal{F}ree_{i+1}$;
```

and the step 1 with the following initialization step:

```
1.   $\mathcal{C}_1 := \{\{A\} | A \in R$ and $\emptyset \Rightarrow \{A\}$ is not a $\delta$-strong rule $\}$;
```

More details on this technique can be found in [7], where it is shown in particular that steps 4.1 and 4.2 can be computed efficiently within the same database scan.

5 Experiments

The running prototype is implemented in C++. We use a PC with 512 MB of memory and a 500 MHz Pentium III processor under Linux operating system.
For an experimental evaluation, we chose the pumsb* data set, a PUMS census data set[3] preprocessed by researchers from IBM Almaden Research Center. The particularity of PUMS data sets is that they are very dense and make the mining of all frequent itemsets together with their supports intractable for low frequency thresholds, because of the combinatorial explosion of the number of frequent itemsets [4].

5.1 Frequent Free-Sets vs. Frequent Sets Condensation

Table 1 shows a comparison of the extraction of frequent sets and frequent free-sets for different frequency thresholds and different values of δ. The collections $FreqFree(r, \sigma, \delta)$ are significantly smaller than the corresponding $Freq(r, \sigma)$. For frequency thresholds of 15% and 20% $Freq(r, \sigma)$ is so large that it is clearly impossible to provide it on our platform, while the extraction of $FreqFree(r, \sigma, \delta)$ remains tractable. For this two frequency thresholds of 15% and 20%, we use lower-bound estimations of $|Freq(r, \sigma)|$. These lower-bounds are computed using the δ-strong rules collected by MinEx (see Section 4.2) to find the size of the

[3] http://www.almaden.ibm.com/cs/quest/data/long_patterns.bin.tar

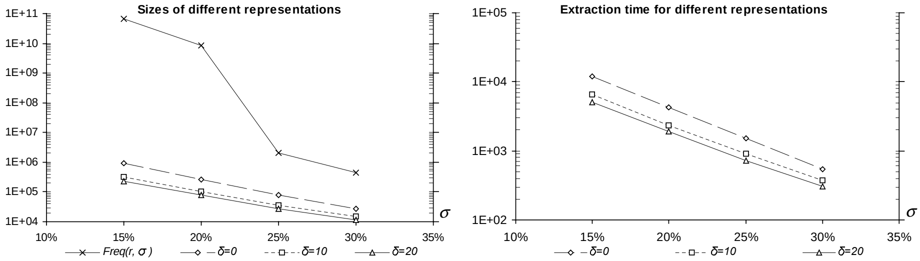

Fig. 1. Extraction time and sizes of different representations.

largest frequent itemset. If this size is m then there is a least 2^m frequent itemsets. Figure 1 (left) emphasizes, using logarithmically scaled axes, the difference of the size of the various representations.

Using also logarithmically scaled axes, Figure 1 (right) shows that the extraction time for MinEx grows up exponentially when the frequency threshold is reduced. This is due to the combinatorial explosion of the number of frequent free-sets. Apriori-based algorithms have the same global exponential evolution of the extraction time, due in this case to the combinatorial explosion of the number of frequent sets.

Table 1. Comparison of different representations at various frequency thresholds.

σ	15%			20%			25%			30%		
δ	0	10	20	0	10	20	0	10	20	0	10	20
Max frequent free-set size (≈MIN-EX DB scans)	12	11	10	12	10	9	11	9	9	10	9	8
|*FreqFree(r, σ, δ)*|	909 806	324 743	232 887	253 107	105 615	76 413	78 220	36 310	27 137	26 972	14 631	11 079
FreqFree(r, σ, δ) extraction time (sec.)	11 977	6 590	5 126	4 233	2 342	1 890	1 540	905	731	533	373	302
Max frequent set size (≈APRIORI DB scans)	35			32			18			16		
|*Freq(r, σ)*|	$>2^{35}$			$>2^{32}$			2 064 946			432 699		

5.2 Scale-Up Experiment

On figure 2 we report the extraction time (for $\sigma = 20\%$) when changing the number of rows or the number of items in the data set. We observe an exponential complexity w.r.t. the number of items and a linear complexity w.r.t. number of rows in the data set if the value of δ follows the number of tuples (e.g., if we double the number of rows then we double the value of δ). This is emphasized by a superimposed straight line on figure 2 (left).

5.3 Approximation Error in Practice

In this experiment, we report the practical error made on σ-frequent itemset supports when using the approximation based on σ-frequent δ-free-sets. The data

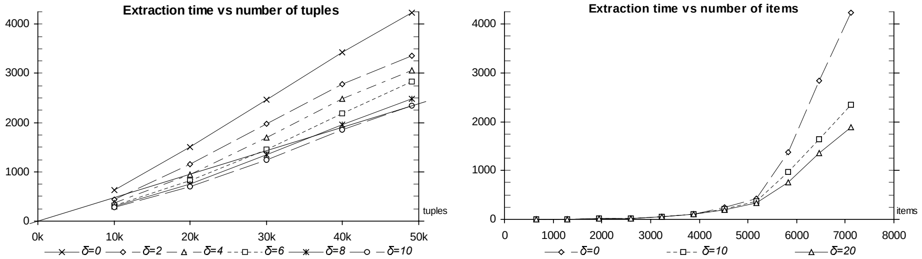

Fig. 2. Behavior of MinEx w.r.t. the number of rows and the number of items.

set is a PUMS data set of Kansas state[4]. We use a version of this data set that has been preprocessed at the University of Clermont-Ferrand (France) in Prof. L. Lakhal's research group. We have reduced this data set to 10000 rows and 317 items to be able to extract all σ-frequent itemsets at a low frequency threshold. For $\sigma = 0.05$ (500 rows), there are 90755 σ-frequent sets and the largest has $n = 13$ items. As a condensed representation, we computed $FreqFreeSup(r, 0.05, 6)$ which contains 4174 elements.

Theoretical error bounds on the frequent set support can be determined using Theorem 2 as follows. In this experiment, the maximal absolute support error is $\delta * n = 6 * 13 = 78$ rows. The maximal relative support error can be obtained assuming that the maximal theoretical absolute error occurs on the σ-frequent set of minimal frequency (i.e. σ). So, the maximal relative support error is $\delta * n/(N * \sigma) = 15.6\%$ ($N = 10000$ rows in the experiment).

The support of each of the 90755 σ-frequent itemsets is approximated using the collection $FreqFreeSup(r, 0.05, 6)$ and Theorem 2 and then compared to the exact support. The maximal absolute support error is 18 rows, and the maximal relative support error is 3.1%. The average absolute support error is 2.12 rows and the average relative support error is 0.28%. Table 2 shows that this error remains very low even for frequent sets containing a lot of items.

Table 2. Error observed on σ-frequent itemset supports by itemset size.

itemset size	1	2	3	4	5	6	7	8	9	10	11	12	13
average abs. sup. error	0	0.24	0.65	1.10	1.53	1.92	2.31	2.75	3.28	3.9	4.58	5.2	5.5
average rel. sup. error	0	0.03%	0.07%	0.13%	0.18%	0.24%	0.31%	0.38%	0.47%	0.58%	0.71%	0.83%	0.88%

[4] ftp://ftp2.cc.ukans.edu/pub/ippbr/census/pums/pums90ks.zip

6 Related Work

Using incomplete information about itemset frequencies for some mining task, e.g., rule mining, has been proposed in [10], and formalized in the general framework of ϵ-adequate representations. Probabilistic approaches to the problem of frequency queries have also been investigated (see [14]).
Several search space reductions based on nearly exact (or exact) association rules have been proposed. The use of the nearly exact association rules to estimate the confidence of other rules and then to prune the search space has been suggested in [3] but not investigated nor experimented. Efficient mining of nearly exact rules (more specifically rules with at most δ exceptions) with a single attribute in both the left and the right hand sides has been proposed in [9]. Search space pruning using exact association rules has been experimented in [3] in the context of rule mining and developed independently in the context of frequent itemsets mining in [13]. [13] implicitely proposes a kind of condensed representation called *closed* itemsets which is strongly related to the notion of δ-free-set when $\delta = 0$.

7 Conclusion and Future Work

We proposed a structure called free-sets that can be extracted efficiently, even on dense data sets, and that can be used to approximate closely the support of frequent itemsets. We formalized this approximation in the framework of ϵ-adequate representations [10] and gave a correct extraction algorithm formulated as an instance of the levelwise search algorithm presented in [11].
We reported experiments showing that frequent free-sets can be extracted even when the extraction of frequent itemsets turns out to be intractable. The experiments also show that the error made when approximating the support of frequent itemsets using the support of frequent free-sets remains very low in practice.
Interesting future work includes applications of the notion of free-set to the discovery of association rules with approximated confidence and support, and to the approximation of boolean formula support as investigated in [10].

References

1. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI Press, 1996.
2. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proc. VLDB'94*, pages 487 – 499, 1994.
3. R. J. Bayardo. Brute-force mining of high-confidence classification rules. In *Proceedings KDD'97*, pages 123–126, 1997.

4. R. J. Bayardo. Efficiently mining long patterns from databases. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 85–93. ACM Press, 1998.
5. J.-F. Boulicaut and A. Bykowski. Frequent closures as a concise representation for binary data mining. In *Proc. PAKDD'00*, volume 1805 of *LNAI*, pages 62–73, Kyoto, JP, 2000. Springer-Verlag.
6. J.-F. Boulicaut and B. Jeudy. Using constraints during itemset mining: a generic approach. Technical Report 2000-01, INSA Lyon, LISI, F-69621 Villeurbanne, Mar. 2000.
7. A. Bykowski. Frequent set discovery in highly-correlated data. Technical Report July 1999, Master of Science thesis, INSA Lyon, LISI, F-69621 Villeurbanne, 1999.
8. A. Bykowski and L. Gomez-Chantada. Frequent itemset extraction in highly-correlated data: a web usage mining application. In *Proc. WKDDM'00*, pages 27–42, Kyoto, JP, Apr. 2000.
9. S. Fujiwara, J. D. Ullman, and R. Motwani. Dynamic miss-counting algorithms: Finding implication and similarity rules with confidence pruning. In *Proc. ICDE'00*, pages 501–511, San Diego, USA, 2000.
10. H. Mannila and H. Toivonen. Multiple uses of frequent sets and condensed representations. In *Proceedings KDD'96*, pages 189–194, Portland, USA, 1996.
11. H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.
12. R. Ng, L. V. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning optimization of constrained association rules. In *Proc. ACM SIGMOD'98*, pages 13–24, Seattle, USA, 1998.
13. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1):25–46, 1999.
14. D. Pavlov, H. Mannila, and P. Smyth. Probalistic models for query approximation with large data sets. Technical Report 2000-07, Univsersity of California, Department of Information and Computer Science, Irvine, CA-92697-3425, Feb. 2000.
15. G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, pages 229 – 248. AAAI Press, Menlo Park, CA, 1991.