Minimal k-Free Representations of Frequent Sets

Toon Calders[1] and Bart Goethals[2]

[1] University of Antwerp, Belgium
[2] Helsinki Institute for Information Technology, Finland

Abstract. Due to the potentially immense amount of frequent sets that can be generated from transactional databases, recent studies have demonstrated the need for concise representations of all frequent sets. These studies resulted in several successful algorithms that only generate a lossless subset of the frequent sets. In this paper, we present a unifying framework encapsulating most known concise representations. Because of the deeper understanding of the different proposals thus obtained, we are able to provide new, provably more concise, representations. These theoretical results are supported by several experiments showing the practical applicability.

1 Introduction

The frequent itemset mining problem is by now well known [1]. We are given a set of items $\mathcal{I}$ and a database $\mathcal{D}$ of subsets of $\mathcal{I}$. The elements of $\mathcal{D}$ are called transactions. An *itemset* $I \subseteq \mathcal{I}$ is some set of items; its *support* in $\mathcal{D}$, denoted $support(I, \mathcal{D})$, is defined as the number of transactions in $\mathcal{D}$ that contain all items of I. An itemset is called *s-frequent* in $\mathcal{D}$ if its support in $\mathcal{D}$ exceeds s. The database $\mathcal{D}$ and the minimal support s are omitted when they are clear from the context. The goal is now, given a minimal support threshold and a database, to find all frequent itemsets. The set of all frequent itemsets is denoted $\mathcal{F}(\mathcal{D}, s)$, the set of infrequent sets is denoted $\overline{\mathcal{F}}(\mathcal{D}, s)$.

Recent studies on frequent itemset mining algorithms resulted in significant performance improvements. However, if the minimal support threshold is set too low, or the data is highly correlated, the number of frequent itemsets itself can be extremely large. To overcome this problem, recently several proposals have been made to construct a concise representation [13] of the frequent itemsets, instead of mining all frequent itemsets: *Closed sets* [2,4,14,15,16], *Free sets* [5], *Disjunction-Free Sets* [6,10], *Generalized Disjunction-Free Generators* [12,11], and *Non-Derivable Itemsets* [8].

A *Concise Representation of frequent sets* is a subset of all frequent sets with their supports that contains enough information to construct all frequent sets with their support. Therefore, based on the representation, for each itemset I, we must be able to (a) decide whether I is frequent, and (b) if I is frequent, produce its support.

Mannila et al. [13] introduced the notion of a concise representation in a more general context. Our definition resembles theirs, but for reasons of simplicity we only concentrate on representations that are exact, and for frequent itemsets.

N. Lavrač et al. (Eds.): PKDD 2003, LNAI 2838, pp. 71–82, 2003.

For representations the term *concise* will refer to their space-efficiency; that is, $\mathcal{R}$ is called *more concise* than $\mathcal{R}'$ if for every database $\mathcal{D}$ and support threshold s, $\mathcal{R}(\mathcal{D}, s)$ is smaller than or equal to $\mathcal{R}'(\mathcal{D}, s)$.

We introduce new representations based on the deduction rules for support presented in [8]. Many of the proposals in the literature, such as the *free sets* [5], the *disjunction-free sets* [6,10], the *generalized disjunction-free sets* [12,11], the *disjunction-free generators* [10], the *generalized disjunction-free generators* [11,12], and the *non-derivable itemsets* [8] representations, will be shown to be manifestations of this method. As such, the proposed method serves as a unifying framework for these representations.

The organization of the paper is as follows. In Section 2 we briefly describe different concise representations in the literature. Section 3 revisits the deduction rules introduced in [8]. In Section 4, a unifying framework for different concise representations is given, based on the deduction rules. Also new, minimal, representations are introduced. In Section 5 we present the results of experiments concerning the size of the different representations.

2 Related Work

Closed Sets. The first successful concise representation was the closed set representation introduced by Pasquier et al. [14]. In short, a *closed set* is an itemset such that its frequency does not equal the frequency of any of its supersets. The collection of the frequent closed sets together with their supports is a concise representation. This representation will be denoted *ClosedRep*.

Generalized Disjunction-Free Sets. [11,12] Let X, Y be two disjunct itemsets. The *disjunctive rule* $X \rightarrow \bigvee Y$ is said to *hold in the database* $\mathcal{D}$, if every transaction in $\mathcal{D}$ that contains X, also contains at least one item of Y. A set I is called *generalized disjunction-free* if there do not exist disjunct subsets X, Y of I such that $X \rightarrow \bigvee Y$ holds. The set of all generalized disjunction free sets is denoted *GDFree*.

In [12], a representation based on the frequent generalized disjunction-free sets is introduced. On the one hand, based on the supports of all subsets of a set I (including I), it can be decided whether I is generalized disjunction-free or not. On the other hand, if a disjunctive rule $X \rightarrow \bigvee Y$ holds, the support of every superset I of $X \cup Y$ can be constructed from the supports of its subsets. For example, $a \rightarrow b \vee c$ holds if and only if for every superset X of abc,

$$supp(X) = supp(X - b) + supp(X - c) - supp(X - bc) \ .$$

Hence, if we know that a rule $X \rightarrow \bigvee Y$ holds, there is no need to store supersets of $X \cup Y$ in the representation.

However, the set of frequent generalized disjunction-free sets *FGDFree* is not a representation. We illustrate this with an example. Suppose that *FGDFree* completed with the supports is $\{(\emptyset, 10), (a, 5), (b, 4), (c, 3), (ab, 3)\}$. What conclusion should be taken for the set ac? There can be two reasons for ac to be left

out of the representation: (a) because ac is infrequent, or (b) because ac is not generalized disjunction-free. Furthermore, suppose that ac was left out because it is not generalized disjunction-free. Since we have no clue which disjunctive rule holds for ac, we cannot produce its support. Hence, *FGDFree* completed with the supports of the sets clearly is not a representation. This problem is resolved in [12] by adding a part of the *border* of the set *FGDFree* to the representation.

Definition 1. *Let S be a set of itemsets.* $\mathcal{B}(S) = \{J \mid J \notin S, \forall J' \subset J : J' \in S\}$.

Suppose that we also store the sets in $\mathcal{B}(FGDFree)$ in the representation. Let I be a set not in $FGDFree \cup \mathcal{B}(FGDFree)$. There exists a set $J \subset I$ in $\mathcal{B}(FGDFree)$. The set J is either infrequent, or not generalized disjunction-free. If J is infrequent, then I is as well. If J is not generalized disjunction-free, then the supports of all subsets of J (including the support of J) allow for determining the rule $X \to \bigvee Y$ that holds for J. Hence, we know a rule $X \to \bigvee Y$ that holds for I $(X, Y \subseteq J \subset I)$. Therefore, from the supports of all strict subsets of I, we can derive the support of I using this rule. Using induction on the cardinality of I, it can easily be proven that $FGDFree \cup \mathcal{B}(FGDFree)$ completed with the supports is a representation. For the details, we refer to [11,12].

It is also remarked in [12] that it is not necessary to store the complete border $\mathcal{B}(FGDFree)$. For example, we could decide to leave out the infrequent sets. When reconstructing the complete set of frequent itemsets, we will be able to recognize these infrequent sets in the border because they are the only sets that have all their strict subsets in *FGDFree*, but that are not in the representation themselves. Other alternatives are the *generalized disjunction-free generators representation* (*GDFreeGenRep*) [12] and the representations in Section 4.

Free and Disjunction-Free Sets. [5,6,10] Free and disjunction-free sets are special cases of generalized disjunction-free sets. For free sets, the righthand side of the rules $X \to \bigvee Y$ is restricted to singletons, for disjunction free sets to singletons and pairs. Hence, a set I is free if and only if there does not exist a rule $X \to a$ that holds with $X \cup \{a\} \subseteq I$, and I is disjunction-free if there does not exists a rule $X \to a \vee b$ that holds with $X \cup \{a, b\} \subseteq I$. The free and disjunction-free sets are denoted respectively by *Free* and *DFree*, the frequent free and frequent disjunction-free sets by *FFree* and *FDFree*.

Again, neither *FFree* nor *FDFree* completed with the supports form a concise representation. The reasons are the same as explained for the generalized disjunction-free sets above. Hence, for the representations based on the free sets and the disjunction-free sets, (parts of) the border must be stored as well. Which parts of the border are stored can have a significant influence on the size of the representations, since the border is often very large, sometimes even larger than the total number of frequent itemsets.

However, the parts of the border that are stored in the representations presented in [5,6,10,11,12] are often far from optimal. In this paper we describe a unifying framework for these disjunctive-rule based representations. This framework is based on the deduction rules for support presented in [8] and revisited in Section 3. The framework allows a neat description of the different strategies

used in the free, disjunction-free and generalized disjunction-free based representations. Due to the deeper understanding of the problem resulting from the unifying framework, we are able to find new and more concise representations that drastically reduce the number of sets to be stored.

3 Deduction Rules

In this section we review the deduction rules introduced in [8]. These rules derive bounds on the support of an itemset I if the supports of all strict subsets of I are known. In [7], it is shown that these rules are sound and complete; that is, they compute the best possible bounds.

Let a *generalized itemset* be a conjunction of items and negations of items. For example, $G = \{a, b, \overline{c}, d\}$ is a generalized itemset. A transaction T *contains a general itemset* $G = X \cup \overline{Y}$ if $X \subseteq T$ and $T \cap Y = \emptyset$. The *support of a generalized itemset* G *in a database* $\mathcal{D}$ is the number of transactions of $\mathcal{D}$ that contain G.

We say that a general itemset $G = X \cup \overline{Y}$ is *based* on itemset I if $I = X \cup Y$. From the well known inclusion-exclusion principle [9], we know that for a given general itemset $G = X \cup \overline{Y}$ based on I,

$$support(G) = \sum_{X \subseteq J \subseteq I} (-1)^{|J \setminus X|} support(J) \ .$$

Since $supp(G)$ is always larger than or equal to 0, we derive

$$\sum_{X \subseteq J \subseteq I} (-1)^{|J \setminus X|} support(J) \geq 0$$

If we isolate $supp(I)$ in this inequality, we obtain the following bound on the support of I:

$$supp(I) \leq \sum_{X \subseteq J \subset I} (-1)^{|I \setminus J|+1} supp(J) \qquad \text{If } |I \setminus J| \text{ odd}$$
$$supp(I) \geq \sum_{X \subseteq J \subset I} (-1)^{|I \setminus J|+1} supp(J) \qquad \text{If } |I \setminus J| \text{ even}$$

This rule will be denoted $\mathcal{R}_I(X)$. Depending of the sign of the coefficient of $supp(I)$, the bound is a lower or an upper bound. If $|I \setminus X|$ is odd, $\mathcal{R}_I(X)$ is an upper bound, otherwise it is a lower bound. Thus, given the supports of all subsets of an itemset I, we can derive lower and upper bounds on the support of I with the rules $\mathcal{R}_I(X)$ for all $G = X \cup \overline{Y}$ based on I.

We denote the greatest lower bound on I by $LB(I)$ and the least upper bound by $UB(I)$. The complexity of the rules $\mathcal{R}_I(X)$ increases exponentially with the cardinality of $I \setminus X$. The number $|I \setminus X|$ is called the *depth* of rule $\mathcal{R}_I(X)$. Since calculating all rules is often tedious, we sometimes restrict ourselves to only rules of limited depth. More specifically, we denote the greatest lower and least upper bounds on the support of I resulting from evaluation of rules up to depth k by $LB_k(I)$ and $UB_k(I)$. Hence, the interval $[LB_k(I), UB_k(I)]$ are the bounds calculated by the rules $\{\mathcal{R}_I(X) \mid X \subseteq I, |I \setminus X| \leq k\}$.

Example 1. Consider the following database:

TID	Items
1	a
2	b
3	c
4	a, b
5	a, c
6	b, c
7	a, b, c

$$\begin{array}{lcl} supp(abc) & \geq & 0 \\ & \leq & s_{ab} = 2 \\ & \leq & s_{ac} = 2 \\ & \leq & s_{bc} = 2 \\ & \geq & s_{ab} + s_{ac} - s_a = 0 \\ & \geq & s_{ab} + s_{bc} - s_b = 0 \\ & \geq & s_{ac} + s_{bc} - s_c = 0 \\ & \leq & s_{ab} + s_{ac} + s_{bc} - s_a - s_b - s_c + s_\emptyset = 1 \end{array}$$

The rules above are the rules $\mathcal{R}_{abc}(X)$ for X respectively $abc, ab, ac, bc, a, b, c, \emptyset$. The first rule has depth 0, the following three rules depth 1, the next three rules depth 2, and the last rule has depth 3. Hence, $LB_0(abc) = 0$, $LB_2(abc) = 0$, $UB_1(abc) = 2$, $UB_3(abc) = 1$. □

Links Between $\mathcal{R}_I(X)$, the support of $X \cup \overline{Y}$, and $X \rightarrow \bigvee Y$. Let I be an itemset, and $G = X \cup \overline{Y}$ a generalized itemset based on I. From the derivation of the rule $\mathcal{R}_I(X)$, it can be seen that the difference between the bound calculated by it, and the actual support of I equals the support of $X \cup \overline{Y}$. Hence, the bound calculated by $\mathcal{R}_I(X)$ equals $supp(I)$ if and only if $supp(X \cup \overline{Y}) = 0$. It is also true that the disjunctive rule $X \rightarrow \bigvee Y$ holds if and only if $supp(X \cup \overline{Y}) = 0$. Indeed, if $supp(X \cup \overline{Y})$ is 0, then there are no transactions that contain X but do not contain any of the items in Y. Therefore, we obtain the following theorem.

Theorem 1. *Let I be an itemset, and $G = X \cup \overline{Y}$ a generalized itemset based on I. The following are equivalent:*
(a) The bound calculated by $\mathcal{R}_I(X)$ equals the support of I,
(b) $supp(G) = 0$, and
(c) The disjunctive rule $X \rightarrow \bigvee Y$ holds. □

Example 2. We continue Example 1. Since the bound 1 calculated by $\mathcal{R}_{abc}(\emptyset)$ equals $supp(abc)$, $supp(\overline{abc})$ must be 0. Indeed, there is no transaction that contains none of a,b, or c. Hence, the disjunctive rule $\emptyset \rightarrow a \vee b \vee c$ holds. On the other hand, the difference between the bound calculated by $\mathcal{R}_{abc}(a)$ and the actual support of abc is 1. Hence, $supp(a \cup \overline{bc}) = 1$. □

4 Unifying Framework

In [8] we introduced the NDI representation based on the deduction rules which we repeated in Section 3. The NDI-representation was defined as follows:

$$NDIRep(\mathcal{D}, s) \ =_{def} \ \{(I, supp(I, \mathcal{D})) \mid supp(I, \mathcal{D}) \geq s, LB(I) \neq UB(I)\}$$

Hence, if a set I is not in the representation, then either $LB(I) = UB(I)$, and hence the support of I is determined uniquely by the deduction rules, or I is

infrequent. A set I with $LB(I) = UB(I)$ is called a *derivable itemset* (DI), otherwise it is called a *non-derivable itemset* (NDI). Derivability is anti-monotone, which allows an Apriori-like algorithm [8].

NDIRep is the only representation that is based on logical implication. For every set I not in the representation, I is either infrequent in every database consistent with the supports in *NDIRep*, or every such database gives the same support to I. All other representations are based on additional assumptions. For example, in the disjunction-free generators representation there is an explicit assumption that all sets in the border of *FGDFree* that are not in the representation, are not free. Such assumptions make it possible to reduce the size of the representations.

In this section, we add similar assumptions to the NDI-based representations. In order to do this, we identify different groups of itemsets: itemsets that are frequent versus those that are infrequent, sets that have support equal to the lower bound, equal to the upper bound, etc. Based on these groups a similar strategy as for the free, the disjunction-free, and the generalized disjunction-free representations will be followed. We identify minimal sets of groups that need to be stored in order to obtain a representation.

4.1 *k*-Free Sets

The k-free sets will be a key tool in the unified framework.

Definition 2.
A set I is said to be k-free*, if $supp(I) \neq LB_k(I)$ and $supp(I) \neq UB_k(I)$.*
A set I is said to be ∞-free, if $supp(I) \neq LB(I)$, and $supp(I) \neq UB(I)$.
The set of all k-free (∞-free) sets is denoted $Free_k$ ($Free_\infty$). □

As the next lemma states, these definitions cover freeness, disjunction-freeness, and generalized disjunction-freeness. The proof is based on Theorem 1, but is omitted because of space restrictions.

Lemma 1. *Let I be an itemset.*

- *I is free if and only if I is 1-free*
- *I is disjunction free if and only if I is 2-free.*
- *I is generalized disjunction-free if and only if I is ∞-free.*

k-freeness is anti-monotone; if a set I is k-free, then all its subsets are k-free as well. Moreover, if $supp(J) = LB_k(J)$ ($supp(J) = UB_k(J)$), then also $supp(I) = LB_k(I)$ ($supp(I) = UB_k(I)$), for all $J \subseteq I$.

4.2 Groups in the Border

Let now $FFree_k$ be the frequent k-free sets. As we argued in Section 2 for the generalized disjunction-free representations, $FFree_k$ is not a representation. Indeed, if a set I is not in the representation, there is no way to know whether I was left out of the representation because I is infrequent, or because $supp(I) = LB_k(I)$,

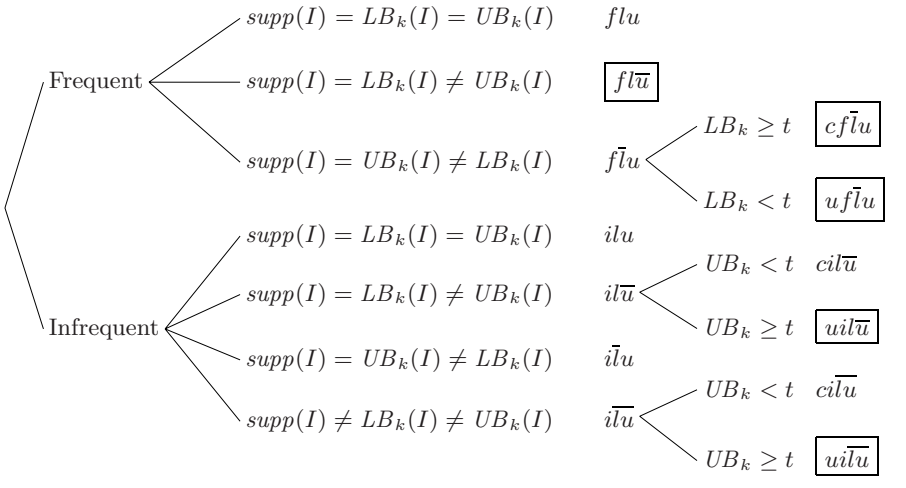


Fig. 1. This tree classifies every set in $\mathcal{B}(FFree_k)$ in the right group. Only the groups that are in a rectangle need to be stored in a representation.

or because $supp(I) = UB_k(I)$. To resolve this problem, parts of the border $\mathcal{B}(FFree_k)$ have to be stored as well. If we can restore the border exactly, then also the other frequent sets can be determined. This can be seen as follows: if a set I is not in $\mathcal{B}(FFree_k)$, and not in $FFree_k$, then it has a subset J in the border. If this set J is infrequent, then so is I. If $supp(J) = LB_k(J)$, then also $supp(I) = LB_k(I)$, and, if $supp(J) = UB_k(J)$, then also $supp(I) = UB_k(I)$ (Lemma 1). Hence, if we can restore the complete border, then we can restore all necessary information.

The sets in $\mathcal{B}(FFree_k)$ can be divided in different groups, depending on whether they are frequent or not, have frequency equal to the lower bound or not, and have frequency equal to the upper bound or not. In order to make the discussion easier, we introduce a 3-letter notation to denote the different groups in the border. The first letter denotes whether the sets in the group are frequent: f is frequent, i is infrequent. The second letter is l if the sets I in the group have $supp(I) = LB_k(I)$, otherwise it is $\bar{l}$. The third letter is u for groups with $supp(I) = UB_k(I)$, and $\overline{u}$ otherwise. The rule depth k is indicated as a subscript to the notation. For example, $f\bar{l}u_k$ denotes the group

$$f\bar{l}u_k \;=_{def}\; \mathcal{B}(FFree_k) \cap \mathcal{F} \cap \{I \mid supp(I) \neq LB_k(I)\} \cap \{I \mid supp(I) = UB_k(I)\} \;,$$

and $il\overline{u}_k$ denotes the group

$$il\overline{u}_k \;=_{def}\; \mathcal{B}(FFree_k) \cap \overline{\mathcal{F}} \cap \{I \mid supp(I) = LB_k(I)\} \cap \{I \mid supp(I) \neq UB_k(I)\} \;.$$

We split some of the groups even further, based on whether or not the bounds $LB_k(I)$, and $UB_k(I)$ allow to conclude that a set is certainly frequent or certainly

infrequent. For example, in the group $f\bar{l}u$, we distinguish between sets I such that the bounds allow to derive that I is frequent, and the other sets. That is, $cf\bar{l}u$ (c of certain), is the set

$$cf\bar{l}u_k =_{def} \mathcal{B}(FFree_k) \cap \mathcal{F} \cap \{I \mid supp(I) \neq LB_k(I)\} \cap \{I \mid supp(I) = UB_k(I)\} \cap \{I \mid LB_k(I) \geq s\}.$$

The other sets of $f\bar{l}u$ are in $uf\bar{l}u$ (u of uncertain). Thus, $uf\bar{l}u$ is the set

$$uf\bar{l}u_k =_{def} \mathcal{B}(FFree_k) \cap \mathcal{F} \cap \{I \mid supp(I) \neq LB_k(I)\} \cap \{I \mid supp(I) = UB_k(I)\} \cap \{I \mid LB_k(I) < s\}.$$

Some of the groups only contain certain or uncertain sets, such as $fl\overline{u}$. Since $fl\overline{u}$ only contains frequent sets I with $supp(I) = LB_k(I)$, automatically the condition $LB_k(I) \geq s$ is fulfilled. The different groups are depicted in Figure 1.

The tree in this figure indicates to which group a set $I \in \mathcal{B}(FFree_k)$ belongs. For example, a frequent set with $supp(I) = LB_k(I)$, and $supp(I) \neq UB_k(I)$, takes the upper branch at the first split, since it is frequent, and the second branch in the second split. Notice that there are no groups with code $f\overline{l\overline{u}}$, because sets that are frequent and have a frequency that equals neither the lower, nor the upper bound, must be in $FFree_k$ and hence cannot be in $\mathcal{B}(FFree_k)$. To make notations more concise, we will sometimes leave out some of the letters. For example, fl_k denotes the union $flu_k \cup fl\overline{u}_k$, and $i\bar{l}_k$ denotes $i\bar{l}u_k \cup ci\overline{lu}_k \cup ui\overline{lu}_k$.

4.3 Representations Expressed with $FFree_k$ and the Groups

We can express many of the existing representations in function of $FFree_k$ for a certain k, and a list of groups in the border of $FFree_k$. Table 1 describes different existing representations in this way. The correctness of this table is proven in [7]. The first line of the table for example, states that the free sets representation actually is

$$(\{(I, supp(I)) \mid I \in FFree_1\}, fl\overline{u}_1, cil\overline{u}_1, uil\overline{u}_1, ci\overline{lu}_1, ui\overline{lu}_1) \ .$$

We do not differentiate between a representation that stores the different groups separately, or in one set; that is, storing the one set $flu \cup f\bar{l}u$ is considered the same as storing the pair of sets $(flu, f\bar{l}u)$. The reason for this is that for space usage the difference between the two is not significant.

The notation $f\overline{u}_{\infty,1}$ and $i\overline{u}_{\infty,1}$ for the generalized disjunction-free generators representation indicates that in this representation, $FFree_\infty$ is used as basis, but for pruning the border $\mathcal{B}(FFree_\infty)$, only rules up to depth 1 are used. In the experiments however, we will use the other rules for pruning the border as well, and hence we report a slightly better size for this representation.

4.4 Minimal Representations

We can not distinguish between two itemsets within the same group if we only use comparisons between their lower and upper bound, their support, and the

Table 1. Representations in function of $FFree_k$ and the groups in $\mathcal{B}(FFree_k)$. *DFreeGenRep* denotes the disjunction-free generators representation, *GDFreeGenRep* the generalized disjunction-free generators representation.

Representation	Base	with frequency	without frequency
FreeRep	$FFree_1$		$\overline{u}_1$
DFreeRep	$FFree_2$	complete border	
DFreeGenRep	$FFree_2$	$fl\overline{u}_2$	$i\overline{u}_2$
GDFreeRep	$FFree_\infty$	complete border	
GDFreeGenRep	$FFree_\infty$	$f\overline{u}_{\infty,1}$	$i\overline{u}_{\infty,1}$
NDIRep	$FFree_\infty$	$f\bar{l}u_\infty, fl\overline{u}_\infty$	

minimal support threshold. Hence, we can think of the different groups as being equivalence classes. We will now concentrate on which of these classes have to be stored to get a minimal representation.

Instead of storing the complete border in a representation, we can restrict ourselves to only some of the groups. It is, for example, not necessary to store the groups flu and ilu, because every set I in these two groups has $supp(I) = LB_k(I) = UB_k(I)$, and thus, its support is derivable. Furthermore, it is not necessary to store the sets in $i\bar{l}u$, $cil\overline{u}$, and $ci\overline{lu}$, because these sets have $UB_k(I) < s$ and thus are certainly infrequent. In Figure 1, the groups which cannot be excluded directly are indicated with boxes. The other groups can always be reconstructed, based on $FFree_k$.

Notice that for all these groups, there is no need to store the supports of the sets in it. For example, for $fl\overline{u}_k$ all sets I in $fl\overline{u}_k$ have $supp(I) = LB_k(I)$. Hence, we can derive the support of a set I if we know that I is in $fl\overline{u}_k$. Similar observations hold for the other groups as well. In the proposed representations, each group is stored separately.

We can reduce the number of groups even more. For some subsets $\mathcal{G} = \{g_1, \ldots, g_n\}$ of the remaining groups $\{fl\overline{u}_k, cf\bar{l}u_k, uf\bar{l}u_k, uil\overline{u}_k, ui\overline{lu}_k\}$, the structure

$$(\{(I, supp(I)) \mid I \in FFree_k\}, g_1, \ldots, g_n)$$

will be a representation, and for some groups $\mathcal{G}$ will not. We denote the structure associated with $\mathcal{G}$ and rules up to depth k with $\mathcal{S}_k(\mathcal{G})$.

The structure $\mathcal{S}_k(\{fl\overline{u}_k, cf\bar{l}u_k\})$ is a representation for every k, but neither $\mathcal{S}_k(\{fl\overline{u}_k\})$, nor $\mathcal{S}_k(\{cf\bar{l}u_k\})$ are. Hence, $\mathcal{S}_k(\{fl\overline{u}_k, cf\bar{l}u_k\})$ is a minimal representation among the representations $\mathcal{S}_k(\mathcal{G})$. The only minimal sets of groups $\mathcal{G}$ such that the associated structures are representations are:

$$\mathcal{G}_1 = \{fl\overline{u}, uf\bar{l}u\}\ , \qquad \mathcal{G}_2 = \{cf\bar{l}u, uf\bar{l}u\}\ ,$$
$$\mathcal{G}_3 = \{fl\overline{u}, uil\overline{u}, ui\overline{lu}\}\ , \text{ and} \qquad \mathcal{G}_4 = \{cf\bar{l}u, uil\overline{u}, ui\overline{lu}\}\ .$$

Theorem 2. [7] *Let $\mathcal{G} \subseteq \{fl\overline{u}, cf\bar{l}u, uf\bar{l}u, uil\overline{u}, ui\overline{lu}\}$. $\mathcal{S}_k(\mathcal{G})$ is a representation if and only if either $\mathcal{G}_1 \subseteq \mathcal{G}$, or $\mathcal{G}_2 \subseteq \mathcal{G}$, or $\mathcal{G}_3 \subseteq \mathcal{G}$, or $\mathcal{G}_4 \subseteq \mathcal{G}$.*

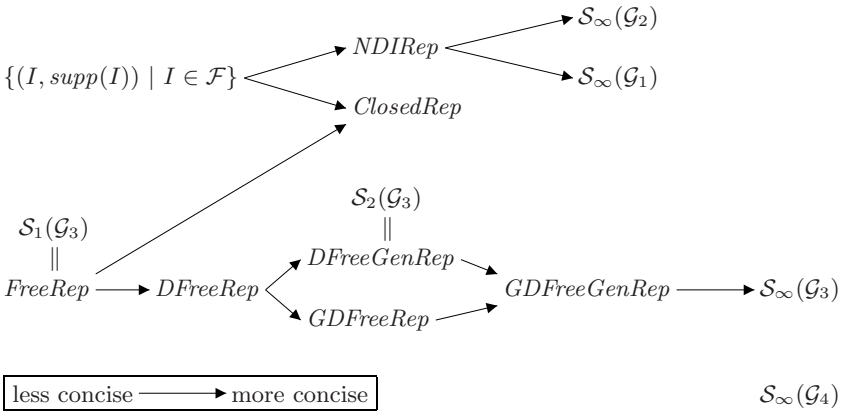


Fig. 2. Relation between the different representations.

For the proof we refer to [7]. The theorem implies that representations $\mathcal{S}_\infty(G_1)$, $\mathcal{S}_\infty(G_2)$,$\mathcal{S}_\infty(G_3)$, and $\mathcal{S}_\infty(G_4)$ are minimal. Thus, all representations in Table 1, have at least one $\mathcal{S}_k(\mathcal{G})$ that is more concise. The relations between the different representations are given in Figure 2. For proofs of the relations see [7].

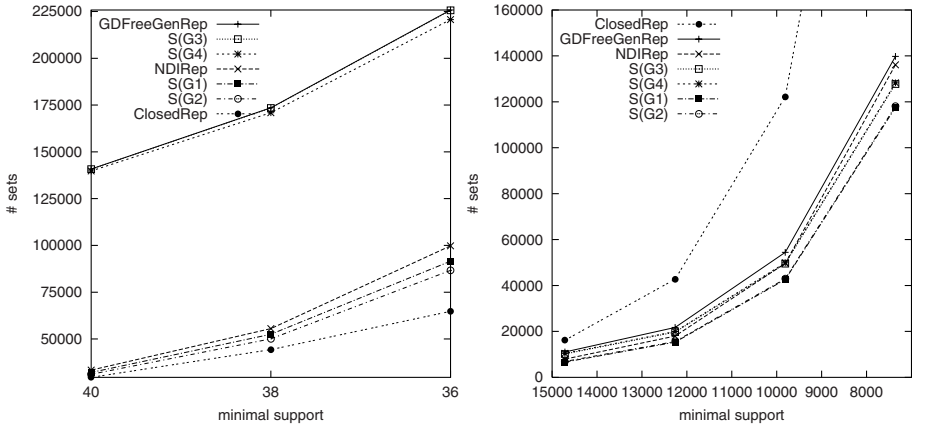
5 Experiments

To empirically evaluate the newly proposed concise representations, we experimented with several database benchmarks used in [16]. Due to space limitations, we only report results for the BMS-Webview-1 dataset, containing 59 602 transactions, created from click-stream data from a small dot-com company which no longer exists [17], and the pumsb* dataset, containing 100 000 transactions from census data from which items that occur more frequently than 80% are removed [3]. Each experiment finished within minutes (mostly seconds) on a 1GHz Pentium IV PC with 1GB of main memory.

Figure 3 shows the total number of itemsets that is stored for each of the four new representations, together with the previously known minimal representations, i.e., the non-derivable itemsets, the closed itemsets, and the generalized disjunction-free generators.

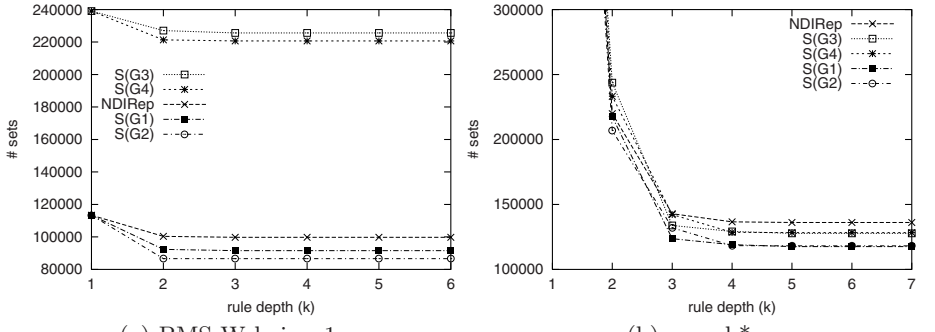
In both experiments, the representations $\mathcal{S}_\infty(\mathcal{G}_1)$ and $\mathcal{S}_\infty(\mathcal{G}_2)$ have more or less the same size. This is not very surprising, since the parts of the border these two representations store have a big overlap. Also the representations $\mathcal{S}_\infty(\mathcal{G}_3)$ and $\mathcal{S}_\infty(\mathcal{G}_4)$ are almost equal in size. Again we see that $\mathcal{G}_3$ and $\mathcal{G}_4$ are almost equal.

Notice also that for BMS-Webview-1 the representations *GDFreeGenRep* and $\mathcal{S}_\infty(\mathcal{G}_3)$ have the same size. The reason for this can be found in Figure 2. In this figure we see that the size of *GDFreeGenRep* is between the sizes of $\mathcal{S}_2(\mathcal{G}_3)$ and $\mathcal{S}_\infty(\mathcal{G}_3)$. Therefore, the fewer rules of depth more than 2 that need to evaluated in order to get optimal bounds, the closer *GDFreeGenRep* will be to $\mathcal{S}_\infty(\mathcal{G}_3)$. In



(a) BMS-Webview-1 (b) pumsb*

Fig. 3. Number of sets in concise representations for varying minimal support.



(a) BMS-Webview-1 (b) pumsb*

Fig. 4. Number of sets in concise representations of BMS-Webview-1 for varying rule depth.

Figure 4, the effect of varying rule depth is given. The plot shows the sizes of the representations $\mathcal{S}_k(\mathcal{G}_i)$ for different values of k. For the BMS-Webview-1 dataset, evaluating rules of depth greater than 2 does not give any additional gain. In the pumsb* dataset, some gain is still achieved with rules of depth 3. Hence, in the BMS-Webview-1 dataset, *GDFreeGenRep* and $\mathcal{S}_\infty(\mathcal{G}_3)$ have similar size, and in the pumsb*-dataset, there is a slight difference in the part of the border that is stored. In the BMS-Webview-1 dataset, the total number of sets in representations $\mathcal{S}_\infty(\mathcal{G}_1)$ and $\mathcal{S}_\infty(\mathcal{G}_2)$ is smaller than all other representations, except for the closed sets. However, in the pumsb* dataset, the closed set representation is much larger than all others. As can be seen, $\mathcal{S}_\infty(\mathcal{G}_3)$ and $\mathcal{S}_\infty(\mathcal{G}_4)$ sometimes contain more sets, which was expected since these representations also include infrequent sets.

Additionally, to get these results, only rules up to depth 3 were needed to be evaluated. This is illustrated in Figure 4, in which we plotted the size of the condensed representation for varying rule depth.

Also, for all other experiments almost no additional gain resulted from evaluating rules of depth larger than 3. As a consequence, the additional effort to evaluate only these rules is almost negligible during the candidate generation of the frequent set mining algorithm. Indeed, for every itemset I, at most $\binom{|I|}{3}$ rules need to be evaluated, each containing at most three terms.

References

1. R. Agrawal, T. Imilienski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. ACM SIGMOD*, pages 207–216, Washington, D.C., 1993.
2. Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, and L. Lakhal. Mining frequent patterns with counting inference. *SIGKDD Explorations*, 2(2):66–75, 2000.
3. C.L. Blake and C.J. Merz. *UCI Repository of machine learning databases*. University of California, Irvine, Dept. of Information and Computer Sciences, `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 1998.
4. J.-F. Boulicaut and A. Bykowski. Frequent closures as a concise representation for binary data mining. In *Proc. PaKDD*, pages 62–73, 2000.
5. J.-F. Boulicaut, A. Bykowski, and C. Rigotti. Approximation of frequency queries by means of free-sets. In *Proc. PKDD*, pages 75–85, 2000.
6. A. Bykowski and C. Rigotti. A condensed representation to find frequent patterns. In *Proc. PODS*, 2001.
7. T. Calders. *Axiomatization and Deduction Rules for the Frequency of Itemsets*. PhD thesis, University of Antwerp, Belgium, `http://win-www.ruca.ua.ac.be/u/calders/download/thesis.pdf`, May 2003.
8. T. Calders and B. Goethals. Mining all non-derivable frequent itemsets. In *Proc. PKDD*, pages 74–85. Springer, 2002.
9. J. Galambos and I. Simonelli. *Bonferroni-type Inequalities with Applications*. Springer, 1996.
10. M. Kryszkiewicz. Concise representation of frequent patterns based on disjunction-free generators. In *Proc. ICDM*, pages 305–312, 2001.
11. M. Kryszkiewicz and M. Gajek. Concise representation of frequent patterns based on generalized disjunction-free generators. In *Proc. PaKDD*, pages 159–171, 2002.
12. M. Kryszkiewicz and M. Gajek. Why to apply generalized disjunction-free generators representation of frequent patterns? In *Proc. ISMIS*, pages 382–392, 2002.
13. H. Mannila and H. Toivonen. Multiple uses of frequent sets and condensed representations. In *Proc. KDD*, 1996.
14. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Proc. ICDT*, pages 398–416, 1999.
15. J. Pei, J. Han, and R. Mao. Closet: An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD Workshop DMKD*, Dallas, TX, 2000.
16. M.J. Zaki and C. Hsiao. ChARM: An efficient algorithm for closed association rule mining. In *TR 99-10, Computer Science, Rensselaer Polytechnic Institute*, 1999.
17. Z. Zheng, R. Kohavi, and L. Mason. Real world performance of association rule algorithms. In *Proc. KDD*, pages 401–406, 2001.