Theoretical Bounds on the Size of Condensed Representations

Nele Dexters and Toon Calders

University of Antwerp, Belgium

Abstract. Recent studies demonstrate the usefulness of condensed representations as a semantic compression technique for the frequent itemsets. Especially in inductive databases, condensed representations are a useful tool as an intermediate format to support exploration of the itemset space. In this paper we establish theoretical upper bounds on the maximal size of an itemset in different condensed representations. A central notion in the development of the bounds are the l-free sets, that form the basis of many well-known representations. We will bound the maximal cardinality of an l-free set based on the size of the database. More concrete, we compute a lower bound for the size of the database in terms of the size of the l-free set, and when the database size is smaller than this lower bound, we know that the set cannot be l-free. An efficient method for calculating the exact value of the bound, based on combinatorial identities of partial row sums, is presented.

1 Introduction

Mining frequent itemsets [1] is a core operation of many data mining algorithms. During the last decade, hundreds of algorithms have been proposed to find frequent itemsets when a database and a user-defined support threshold are given. However, when this minimal support threshold is set too low or when the data are highly correlated, the process of mining frequent itemsets can result in an immense amount of frequent sets. Even the most efficient mining algorithms cannot cope with this combinatorial blow-up. To overcome this problem, condensed representations can be used. Condensed representations were introduced in [13] in the slightly different context of arbitrary Boolean rules. Intuitively, a condensed representation can be seen as a compact view on the data that allows for answering user queries more efficiently than directly from the original data. In [13], for example, the collection of frequent sets is considered as a condensed representation that allows to speed up frequency counts of arbitrary Boolean expressions over the items. In this paper we concentrate on condensed representations for the collection of frequent itemsets itself, since this collection can already be far too large to store. A condensed representation in the context of the frequent itemsets can be a sub-collection of all frequent itemsets that still contains all information to construct the frequent sets with their corresponding supports. The best-known example of a condensed representation is the *closed itemsets representation* [14]. Other examples are the *Free Sets* [2], the

Disjunction Free Sets [3], the *Generalized Disjunction Free Sets* [12], and the *Non-Derivable Sets* [7].

Especially in inductive databases, condensed representations are a useful tool as an intermediate format to support exploration of the itemset space. In fact, the role of a condensed representation in an inductive database is comparable to a partly materialized view in a data warehouse: materializing all frequent itemsets off-line would speed-up the exploration enormously, but is infeasible because of the gigantic number of them. Instead, the condensed representation is materialized. This representation is much smaller, but, at the same time, contains enough information to speed up ad-hoc querying in the inductive database. When the user asks a query concerning the frequencies of itemsets, these frequencies can be computed more efficiently from the condensed representation than directly from the database. Depending on time and space constraints, the type of condensed representation can be chosen. For example, the free sets representation is less compact than the disjunction free representation, but allows for faster computation of frequency queries.

An important question now is: how concise is a condensed representation; do we have guarantees about the maximal size of a representation? The usefulness of a condensed representation relies critically on its size. In this paper we establish theoretical upper bounds on the maximal size of an itemset for all representations that are based on l-free sets [6]. These representations include the *Free Sets* [2], the *Disjunction Free Sets* [3], the *Generalized Disjunction Free Sets* [12], the *Non-Derivable Sets* [7], and all the variants of these representations, such as the disjunction free and generalized disjunction free generators representations [9, 11]. Hence, based on the size of the database, we present worst-case bounds on the size of the largest sets in these representations.

A central notion in the development of the bounds are thus the l-free sets. Each of the aforementioned representations can be expressed in terms of l-freeness. It was shown in [6], that these representations can be expressed as the collection of frequent l-free sets together with a part of the border, for different values of l. The border of the collection of the frequent l-free sets are the itemsets that are not frequent l-free themselves, but all their subsets are frequent l-free. For example, the free sets representation of [2], corresponds to the collection of the frequent 1-free sets plus the sets in the border that are infrequent. For more details about the connection between the l-free sets and existing condensed representations, we refer to [6].

In this paper, we will bound the maximal cardinality of an l-free set based on the size of the database. More concrete, we compute a lower bound on the size of the database in terms of the size of the l-free set, and when the database size is smaller than this lower bound, we know that the set cannot be l-free. In this paper, we thus give general results relating l-freeness of a set I with a bound on the size of the database $\mathcal{D}$ in terms of the size of I. The results for a particular l can be generalized to the case where l equals the size of I, yielding a connection between ∞-freeness and a bound on the size of $\mathcal{D}$ in terms of the size of I, and can also be extended to NDIs. Because the aforementioned representations can be expressed as the collection of frequent l-free sets plus some sets in the border, the maximal size of a set in the representations is the maximal size of a frequent l-free set plus 1, since the sets in the border can be at most 1 item larger than the largest frequent l-free set. In this way, we extend results of [7] and of [10] that relate the database size to the maximal length of respectively the non-derivable itemsets and the generalized disjunction free sets. Hence, even though we concentrate on a bound on the l-free sets, the main goal of the paper is to establish a bound on the condensed representations that are based on the l-free sets.

An efficient method, the *sum-of-binomials triangle*, for calculating the exact value of the bound based on combinatorial identities of partial row sums is presented and an approximation that is easy to compute is given. From this triangle, we can conclude interesting facts concerning the size of the database.

The organization of the paper is as follows. Section 2 revisits the notions of deduction rules. In Section 3 the bounds on the size of the database that are related to the l-freeness of a set are introduced. Section 4 gives an approximation of this bound, while Section 5 discusses an efficient method to compute the exact bound. In Section 6, our work is related to other papers. Section 7 concludes the paper and gives further research directions.

2 Deduction Rules Revisited

In this section we refresh the deduction rules introduced in [4]. The deduction rules allow for deriving a lower and an upper bound on the support of an itemset, based on the support of its subsets. For example, for the itemset abc we can find the following lower bound on the support:

$$supp(abc) \geq supp(ab) + supp(ac) - supp(a) \ .$$

We first give a complete collection of deduction rules in general. Then, the depth of a rule is defined and only rules up to a certain depth are considered. Next, the notion of an l-free set is introduced. The l-free sets are an important concept, since in [6], it was shown that many condensed representations can easily be expressed in a uniform way using the l-free sets. We will not go into detail about this uniform framework, but only give the intuition behind it. For the exact details, we refer the reader to [6].

2.1 General Concept of Deduction Rules

We start from a database $\mathcal{D}$ with $|\mathcal{D}| = m$ transactions, based on $|\mathcal{I}| = n$ items. We consider an itemset $I \subseteq \mathcal{I}$ with k elements ($|I| = k$) and we are interested in the support of I in the database $\mathcal{D}$: $supp(I, \mathcal{D})$. In [6] (in a somewhat different form), the following relation between the support of I and its subsets was shown:

Theorem 1. *Let $\delta_X(I, \mathcal{D})$ denote the following sum ($X \subseteq I$):*

$$\delta_X(I, \mathcal{D}) = \sum_{X \subseteq J \subset I} (-1)^{|I \setminus J|+1} supp(J) \ .$$

Upper/Lower Bounds $\boldsymbol{\delta_X(I,\mathcal{D})}$	$\boldsymbol{X}$	$\boldsymbol{\lvert I \setminus X \rvert}$	$\boldsymbol{X \cup \overline{Y}}$
$supp(I,\mathcal{D}) \leq s_{ab} + s_{ac} + s_{bc} - s_a - s_b - s_c + s_{\{\}}$	$\{\}$	3	$\overline{abc}$
$supp(I,\mathcal{D}) \geq s_{ab} + s_{ac} - s_a$	a	2	$a\overline{bc}$
$supp(I,\mathcal{D}) \geq s_{ab} + s_{bc} - s_b$	b	2	$\overline{a}b\overline{c}$
$supp(I,\mathcal{D}) \geq s_{ac} + s_{bc} - s_c$	c	2	$\overline{ab}c$
$supp(I,\mathcal{D}) \leq s_{ab}$	ab	1	$ab\overline{c}$
$supp(I,\mathcal{D}) \leq s_{ac}$	ac	1	$a\overline{b}c$
$supp(I,\mathcal{D}) \leq s_{bc}$	bc	1	$\overline{a}bc$
$supp(I,\mathcal{D}) \geq 0$	abc	0	abc
		Depth $\boldsymbol{l}$	

$\boldsymbol{X \cup \overline{Y}}$	**Equalities**
$\overline{abc}$	$supp(I,\mathcal{D}) = s_{ab} + s_{ac} + s_{bc} - s_a - s_b - s_c + s_{\{\}} - s_{\overline{abc}}$
$a\overline{bc}$	$supp(I,\mathcal{D}) = s_{ab} + s_{ac} - s_a + s_{a\overline{bc}}$
$\overline{a}b\overline{c}$	$supp(I,\mathcal{D}) = s_{ab} + s_{ac} - s_a + s_{\overline{a}b\overline{c}}$
$\overline{ab}c$	$supp(I,\mathcal{D}) = s_{ac} + s_{bc} - s_c + s_{\overline{ab}c}$
$ab\overline{c}$	$supp(I,\mathcal{D}) = s_{ab} - s_{ab\overline{c}}$
$a\overline{b}c$	$supp(I,\mathcal{D}) = s_{ac} - s_{a\overline{b}c}$
$\overline{a}bc$	$supp(I,\mathcal{D}) = s_{bc} - s_{\overline{a}bc}$
abc	$supp(I,\mathcal{D}) = s_{abc}$

Table 1. Deduction rules for the set *abc*. s_J denotes $supp(J)$.

Then, $supp(I,\mathcal{D}) = \delta_X(I,\mathcal{D}) + (-1)^{|Y|} supp(X \cup \overline{Y}, \mathcal{D})$ *where* $Y = I \setminus X$*, and* $supp(X \cup \overline{Y}, \mathcal{D})$ *denotes the number of transactions in* $\mathcal{D}$ *that contains all items in* X*, and none of the items in* Y*.*

Hence, for all $X \subseteq I$, depending on the sign of $|Y|$, $\delta_X(I,\mathcal{D})$ is an upper ($|Y|$ odd), or a lower ($|Y|$ even) bound on the support of I. The set $X \cup \overline{Y}$ in Theorem 1, is called a *generalized itemset based on* I. For the complete set of rules for the example where $I = \{a,b,c\}$, see Table 1. With these rules we can compute a lower and upper bound on the support of I when we assume that the supports of all its strict subsets are known. The lower bound is denoted by $LB(I,\mathcal{D})$ and the upper bound by $UB(I,\mathcal{D})$. That is:

$$
\begin{aligned}
LB(I,\mathcal{D}) &= \max\{\delta_X(I,\mathcal{D}) \mid X \subseteq I, |I \setminus X| \text{ even}\} \\
UB(I,\mathcal{D}) &= \min\{\delta_X(I,\mathcal{D}) \mid X \subseteq I, |I \setminus X| \text{ odd}\}
\end{aligned}
$$

Notice that the complexity of the sum $\delta_X(I,\mathcal{D})$ depends on the cardinality of $Y = I \setminus X$. This number $|Y|$ is called the *depth* of the rule $\delta_X(I,\mathcal{D})$. Hence, the deeper a rule is, the more complex it is (see Table 1). Therefore, it is often interesting to only consider rules up to a fixed depth l. The lower and upper bounds calculated with rules up to depth l will be denoted $LB_l(I,\mathcal{D})$ and $UB_l(I,\mathcal{D})$. That is:

$$
\begin{aligned}
LB_l(I,\mathcal{D}) &= \max\{\delta_X(I,\mathcal{D}) \mid X \subseteq I, |I \setminus X| \text{ even}, |I \setminus X| \leq l\} \\
UB_l(I,\mathcal{D}) &= \min\{\delta_X(I,\mathcal{D}) \mid X \subseteq I, |I \setminus X| \text{ odd}, |I \setminus X| \leq l\}
\end{aligned}
$$

When it is clear from the context we do not explicitly write down $\mathcal{D}$ in the formulas.

Example 1. Consider the following database:

TID	Items
1	a, b, c, d
2	a, b, c
3	a, b, d, e
4	c, e
5	b, d, e
6	a, b, e
7	a, c, e
8	a, d, e
9	b, c, e
10	b, d, e

In this database, the following supports hold:

$$\begin{array}{llll} supp(\{\}) = 10 & supp(a) \ = 6 & supp(b) \ = 7 & supp(c) = 5 \\ supp(ab) = 4 & supp(ac) = 3 & supp(bc) = 3 & \end{array}$$

The deduction rules for abc up to level 2 are the following (see Table 1):

$$\begin{array}{ll} supp(abc) \geq \delta_a(abc) = 1 & supp(abc) \leq \delta_{ab}(abc) = 4 \\ supp(abc) \geq \delta_b(abc) = 0 & supp(abc) \leq \delta_{ac}(abc) = 3 \\ supp(abc) \geq \delta_c(abc) = 1 & supp(abc) \leq \delta_{bc}(abc) = 3 \\ & supp(abc) \geq \delta_{abc}(abc) = 0 \end{array}$$

Hence, based on the supports of the subsets of abc, we can deduce that $LB_1(abc) = 0$, $UB_1(abc) = 3$, $LB_2(abc) = 1$ and $UB_2(abc) = 3$.

2.2 *l*-Freeness of an Itemset

A very important notion in the context of a unifying framework for the condensed representations is l-freeness:

Definition 1. *Let l be a positive integer. A set I is l-free, if $supp(I, \mathcal{D}) \neq LB_l(I, \mathcal{D})$, and $supp(I, \mathcal{D}) \neq UB_l(I, \mathcal{D})$. A set I is ∞-free, if $supp(I, \mathcal{D}) \neq LB(I, \mathcal{D})$, and $supp(I, \mathcal{D}) \neq UB(I, \mathcal{D})$.*

In [6], the following properties of l-freeness were shown: l-freeness is anti-monotone; that is, every subset of an l-free itemset is also l-free, and every superset of an itemset that is not l-free is also not l-free. l-freeness is interesting in the context of condensed representations, because the support of any non-l-free set can be derived as follows: if $supp(I, \mathcal{D}) = LB_l(I, \mathcal{D})$, then for all $I \subseteq J$, $supp(J, \mathcal{D}) = LB_l(J, \mathcal{D})$. Hence, if we observe the fact $supp(I, \mathcal{D}) = LB_l(I, \mathcal{D})$, there is no need to store any of the supersets of I in a condensed representation. The representations that rely on l-freeness hence store the frequent l-free sets, and some sets that are "on the border." For a detailed description we refer to [6].

From Theorem 1, the following lemma easily follows:

Lemma 1. *Let l be a positive integer, I an itemset, $X \subseteq I$. I is l-free if and only if $supp(X \cup \overline{Y}) \neq 0$ for all generalized itemsets $X \cup \overline{Y}$ that are based on I, with $|Y| \leq l$.*

2.3 Link Between *l*-Freeness and Condensed Representations

The following proposition from [6], links the *free sets* [2], the *disjunction free sets* [3, 9], and the *generalized disjunction free sets* [12, 11] with l-freeness, for different values of l.

Proposition 1. *Link between l-freeness with other condensed representations.*

- I is free $\Leftrightarrow I$ is $1-$ free
- I is disjunction free $\Leftrightarrow I$ is $2-$ free
- I is generalized disjunction free $\Leftrightarrow I$ is $\infty-$ free
- I is NDI $\Rightarrow$ every strict subset of I is $\infty-$ free

From the unified framework introduced in [6] the following proposition making the connection between the size of an l-free set and the different condensed representations is immediate (recall from the introduction that the different representations can be expressed as l-free sets *plus the border*. Hence, the representations can contain sets that are 1 item larger than the largest l-free set):

Proposition 2. *Let $max(l, \mathcal{D})$ be the length of the largest l-free set in $\mathcal{D}$.*

- *Every set in the free sets representation [2] has length at most $max(1, \mathcal{D})+1$.*
- *Every set in the disjunction free sets representation [3, 9] has length at most $max(2, \mathcal{D}) + 1$.*
- *Every set in the generalized disjunction free sets representation [12, 11] has length at most $max(\infty, \mathcal{D}) + 1$.*
- *Every set in the non-derivable itemsets representation [7] has length at most $max(\infty, \mathcal{D}) + 1$.*

Hence, because of Proposition 2, a theoretical bound on the size of the l-free sets immediately leads to a bound on many condensed representations.

3 Bounds on the Size of the Database

In this section we present the theoretical lower bound $d_l(k)$ on the size of the database in terms of the size k of the largest l-free set. Hence, if $\mathcal{D}$ contains an l-free set of size k, then the cardinality of $\mathcal{D}$ must be at least $d_l(k)$. This result then allows for deriving the maximal cardinality of an l-free set based on the size of a database. Indeed; the maximal size k of an l-free set is the largest integer k such that $|\mathcal{D}| \geq d_l(k)$.

3.1 Bounds for *l*-Free Sets

We illustrate the principle of the bound with an example. Let $I = abcd$ be a 2-free set. We will show how the lower bound $d_2(4)$ on the size of $\mathcal{D}$ can be derived. Because I is 2-free, by definition, $supp(I, \mathcal{D}) \neq LB_2(I, \mathcal{D})$ and $supp(I, \mathcal{D}) \neq UB_2(I, \mathcal{D})$. Because of Lemma 1, for all generalized itemsets $X \cup \overline{Y}$ based on I, with $|Y| \leq 2$, $supp(X \cup \overline{Y}, \mathcal{D}) \neq 0$. In the case of $abcd$, this means that

$$\begin{array}{l} supp(\overline{ab}cd) > 0,\, supp(\overline{a}b\overline{c}d) > 0,\, supp(\overline{a}bc\overline{d}) > 0, \\ supp(a\overline{bc}d) > 0,\, supp(a\overline{b}c\overline{d}) > 0,\, supp(ab\overline{cd}) > 0, \\ supp(\overline{a}bcd) > 0,\, supp(a\overline{b}cd) > 0,\, supp(ab\overline{c}d) > 0, \\ supp(abc\overline{d}) > 0,\, supp(abcd) > 0 \;\; . \end{array}$$

Every transaction can make only one of these conditions true. Indeed; suppose that a transaction T supports *both* $\overline{ab}cd$ and $\overline{a}b\overline{c}d$. Then, T must at the same time not contain b ($\overline{ab}cd$) and contain b ($\overline{a}b\overline{c}d$), and that is clearly impossible. Hence, a database $\mathcal{D}$ in which *abcd* is 2-free, must contain at least one transaction for each generalized itemset $X \cup \overline{Y}$ based on *abcd* with $|Y| \leq 2$. Hence, to get the lower bound on the size of the database, we we have to count the number of generalized itemsets consisting of 4 items with at most two negated items. There are $\binom{4}{2} = 6$ generalized itemsets consisting of 4 elements with exactly two elements negated and $\binom{4}{1} = 4$ generalized itemsets of size 4 with exactly 1 item negated. There exists only 1 generalized itemset of size 4 with no items negated. Hence, every database in which *abcd* is 2-free needs to have at least $d_2(4) = 6 + 4 + 1 = 11$ transactions.

In general, let I be l-free with $|I| = k$. Then, for every generalized itemset $X \cup \overline{Y}$ based on I with $|Y| \leq l$, there needs to be at least one supporting transaction. For each generalized itemset we thus have k items and at most l of them can be negated. We now count all the possibilities with no item of the k items negated, with 1 item negated, ..., up to when l items out of k are negated. Hence, in general, we have:

$$d_l(k) \;=\; \binom{k}{0} + \binom{k}{1} + \ldots + \binom{k}{l} \;=\; \sum_{i=0}^{l} \binom{k}{i}$$

This reasoning leads directly to the following theorem:

Theorem 2.

$$I \text{ is } 1-\text{free} \Rightarrow |\mathcal{D}| \geq \sum_{i=0}^{l} \binom{k}{i} \tag{1}$$

$$|\mathcal{D}| < \sum_{i=0}^{l} \binom{k}{i} \Rightarrow I \text{ is not } l-\text{free} \tag{2}$$

3.2 Bounds For ∞-Free Sets

If we take l equal to k, the size of I, we use *all* the deduction rules to derive the support of I. Based on (1) and (2) we now have the following results:

$$I \text{ is } \infty-\text{free} \Rightarrow |\mathcal{D}| \geq \sum_{i=0}^{k} \binom{k}{i} = 2^k \tag{3}$$

$$|\mathcal{D}| < 2^k \Rightarrow I \text{ is not } \infty-\text{free} \tag{4}$$

From eqs. (3) and (4), it follows that

Theorem 3. I is $\infty-$free $\Rightarrow |I| \leq \log_2(|\mathcal{D}|)$
Hence, $|I| > \log_2(|\mathcal{D}|) \Rightarrow I$ is not $\infty-$free

4 Approximation of the Bound

As already mentioned before, the main focus of this paper is to derive the maximal cardinality of an l-free set, based on the size of the database. The maximal size k of an l-free set is the largest integer k such that $|\mathcal{D}| \geq d_l(k)$. This lower bound on the size of the database is an incomplete binomial sum in terms of l and k and it is difficult to rewrite it to an expression for k in terms of l and $|\mathcal{D}|$. Therefore, we try to find a lower bound for $d_l(k)$ yielding a simple expression that is easy convertible in a result for k in terms of l and $|\mathcal{D}|$. We use the following approximation:

$$\binom{k}{l} \leq \sum_{i=0}^{l} \binom{k}{i}.$$

It is known that $\dfrac{(k-l)^l}{l!} \leq \dbinom{k}{l} = \dfrac{k(k-1)\cdots(k-l+1)}{l!} \leq \dfrac{k^l}{l!}$.

If we now require that $|\mathcal{D}| < \frac{(k-l)^l}{l!}$ we have for sure that the left side of the implication (2) is satisfied and therefore that I is not l-free. Even more, we find the following condition for k:

Proposition 3. $k > \sqrt[l]{l!|\mathcal{D}|} + l \Rightarrow I \text{ is not } l\text{−free}$

5 Exact Computation of the Bound

In Section 3.1 we derived Theorem 2. A crucial part in the equations (1) and (2) is the incomplete binomial sum that is completely determined by l and k:

$$d_l(k) =_{\text{def}} \sum_{i=0}^{l} \binom{k}{i}$$

This is the exact amount of generalized itemsets that is needed to to make a set I of size k, l-free. We can now find a recursion relation between the different $d_l(k)$'s. We illustrate the relation with an example. Suppose we want to know the value of $d_2(4)$. $d_2(4)$ corresponds to the number of generalized disjunction free sets of size 4 with at most 2 negations. Let *abcd* be the base of the generalized disjunction free sets. The disjunction free sets based on *abcd* can be divided into two groups: the ones with d, and the ones with $\overline{d}$. Let $X \cup \overline{Y}$ be a generalized itemset of the first type. Then, $X \setminus \{d\} \cup \overline{Y}$ is a generalized itemset based on *abc* with at most 2 negations. Similarly, if we take $\overline{d}$ out of a generalized itemset based on *abcd* of the second type, we get a generalized itemset base on *abc* with at most 1 negation. Hence, there are $d_2(3)$ generalized itemsets of the first kind, and $d_1(3)$ of the second type. Hence, $d_2(4) = d_2(3) + d_1(3)$. An illustration of this example can be found in Table 2.

For general l and k, we get the following recursive relation:

$$d_l(k) = d_l(k-1) + d_{l-1}(k-1) \tag{5}$$

$d_2(3)$	
$\sum_{i=0}^{2}\binom{3}{i}=7$	
$a\overline{bc}$	$\binom{3}{2}=3$
$\overline{a}b\overline{c}$	
$\overline{ab}c$	
$ab\overline{c}$	$\binom{3}{1}=3$
$a\overline{b}c$	
$\overline{a}bc$	
abc	$\binom{3}{0}=1$

→

$d_2(4)$		$d_2(3)$	$d_1(3)$
$\sum_{i=0}^{2}\binom{4}{i}=11$		$\sum_{i=0}^{2}\binom{3}{i}=7$	$\sum_{i=0}^{1}\binom{3}{i}=4$
$ab\overline{cd}$	$\binom{4}{2}=6$		$ab\overline{c\boldsymbol{d}}$
$a\overline{b}c\overline{d}$			$a\overline{b}\,\overline{c\boldsymbol{d}}$
$a\overline{bc}d$		$a\overline{bc}\boldsymbol{d}$	
$\overline{a}bc\overline{d}$			$\overline{a}bc\overline{\boldsymbol{d}}$
$\overline{a}b\overline{c}d$		$\overline{a}b\overline{c}\boldsymbol{d}$	
$\overline{ab}cd$		$\overline{ab}c\boldsymbol{d}$	
$abc\overline{d}$	$\binom{4}{1}=4$		$abc\overline{\boldsymbol{d}}$
$ab\overline{c}d$		$ab\overline{c}\boldsymbol{d}$	
$a\overline{b}cd$		$a\overline{b}c\boldsymbol{d}$	
$\overline{a}bcd$		$\overline{a}bc\boldsymbol{d}$	
$abcd$	$\binom{4}{0}=1$	$abc\boldsymbol{d}$	

Table 2. Total amount of generalized itemsets for a set of size $k = 4$, consisting of the items a, b, c and d, for level 2 based on a subset of size $k = 3$.

When we rewrite this relation with the partial binomial sums, we get:

$$\sum_{i=0}^{l}\binom{k}{i}=\sum_{i=0}^{l}\binom{k-1}{i}+\sum_{i=0}^{l-1}\binom{k-1}{i}$$

This relation is also known as *Pascal's 6th Identity of Partial Row Sum Rules*.

Because $l \leq k$, we only need to know the diagonal elements $d_k(k)$ and the base-elements $d_1(k)$ to use the above recurrence relation (5). With this knowledge, we can construct a triangle with the incomplete binomial sums $d_l(k)$.

This *sum-of-binomials* triangle has several interesting properties (see Fig. 1):

- The diagonal defined by $l = k$ is easy to compute because $\sum_{i=0}^{k}\binom{k}{i}=2^k$.
- The bottom line for $l = 0$ is always 1.
- The base line for $l = 1$ is always $k + 1$. For a set of size k to be 1-free, $\binom{k}{0}+\binom{k}{1}=1+k$ generalized itemsets are needed.
- The line under the diagonal defined by $l = k - 1$ is always $2^k - 1$. $d_k(k) = \sum_{i=0}^{k}\binom{k}{i}=2^k$ and $d_{k-1}(k)=\sum_{i=0}^{k-1}\binom{k}{i}=2^k-\binom{k}{k}=2^k-1$.
- Parallellogramrule: the entry $d_{l'}(k')$ for a certain couple (k', l') can be calculated using (5) and therefore needs all the other entries in the parallelogram that can be constructed starting in that couple (k', l') and drawing lines parallel with the diagonal and the horizontal axes. The parallellogram is then

$\boldsymbol{l}$	1	2	4	8	16	32	64	128	256	.	.	$\mathbf{2}^{\boldsymbol{k}}$	.	.
.	.	.	.	.	.	.	.	.	.	.	↗	$\mathbf{2}^{\boldsymbol{k}} - \mathbf{1}$		.
.	.	.	.	.	.	.	.	.	.	↗	↗	.	.	.
8	1	2	4	8	16	32	64	128	**256**	↗	.	.	.	.
7	1	2	4	8	16	32	64	**128**	255	.	.	.	.	.
6	1	2	4	8	16	32	**64**	127	297	.	.	.	.	.
5	1	2	4	8	16	**32**	63	120	219	.	.	.	.	.
4	1	2	4	8	**16**	31	***57***	99	163	.	.	.	.	.
3	*1*	*2*	*4*	***8***	*15*	*26*	42	64	93	.	.	.	⟶	$\frac{\boldsymbol{n}^{\mathbf{3}}}{\mathbf{3!}}$
2	1	2	**4**	7	11	16	22	29	37	.	.	.	⟶	$\frac{\boldsymbol{n}^{\mathbf{2}}}{\mathbf{2!}}$
1	1	**2**	3	4	5	6	7	8	9	→	→	**k+1**	⟶	$\boldsymbol{n}$
0	**1**	1	1	1	1	1	1	1	1	→	→	**1**	⟶	**1**
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	.	.	$\boldsymbol{k}$	.	**Limit**.

Fig. 1. Sum-of-binomials triangle

bounded by the diagonal defined by $l = k$, the horizonthal base line defined by $l = 1$ and the lines parallel with these basic axes defined by $l = l'$ and $l = k - (k' - l')$. For an example, take $k' = 6$ and $l' = 4$.

- Sumrule: the entry $d_{l'}(k')$ for a certain couple (k', l') can also be computed by taking $\left(\sum_{i=0}^{k'-1} d_{l'-1}(i)\right) + 1$. This is taking the sum of all the entries on the lower line, one step shifted to the left, plus 1. For example, if we take $k' = 6$ and $l' = 4$ we see that $57 = (1 + 2 + 4 + 8 + 15 + 26) + 1$.

To give an idea of the complete(d) diagram, see Figure 1. When $l = 2$ we find $d_2(k) = \dfrac{k^2}{2} + o(k)$. When $l = 3$ we find $d_3(k) = \dfrac{k^3}{2 \cdot 3} + o(k^2)$. In general, $d_l(k) = \dfrac{k^l}{l!} + o(k^{l-1})$.

With the use of Proposition 2 and these bounds, we also find bounds for the various condensed representations.

6 Related Work

In [5], the $\log_2(|\mathcal{D}|) + 1$-bound on the NDI-representation was already proven. In [10], it is showed, using a very similar technique, the upper bound $\log_2(|\mathcal{D}|)+1$ on the cardinality of generalized disjunction free set. Notice that this claim is less strong than our claim that the largest ∞-free set is at most $\log_2(|\mathcal{D}|)$, and that ∞-free and generalized disjunction free is the same. This discrepancy comes partially from a slight difference in definition between generalized disjunction free sets in [12], and in [6]. The results in [10] are based on the definitions in [12], while the results in this paper are based on the definitions in [6]. We next explain the difference and motivate our choice to follow the definition of [6].

The original definition of generalized disjunction free sets in [12] relies on the notion of disjunctive rules $I \setminus Y \to \bigvee Y$. A rule $I \setminus Y \to \bigvee Y$ holds in a transaction database if and only if every transaction that contains all items in $I \setminus Y$ also contains at least one item in Y. A set I is said to be generalized disjunction free if for all *non-empty* $Y \subset I$, the rule $I \setminus Y \rightarrow \bigvee Y$ does not hold. Notice that the rule $I \setminus Y \rightarrow \bigvee Y$ holds if and only if $supp((I \setminus Y) \cup \overline{Y}) = 0$ [6]. Since $Y = \emptyset$ is not considered, a set is generalized disjunction free according to [12] if and only if $supp((I \setminus Y) \cup \overline{Y}) \neq 0$ for all non-empty subsets Y of I. In [6], however, also the rule $I \rightarrow \emptyset$ is considered. The rule $I \rightarrow \emptyset$ only holds for sets with support equal to 0, since the right-hand side is the empty disjunction, which is always false. Therefore, the only sets for which there is a difference, are sets with support equal to 0. There are situations in which a set with support equal to 0 is generalized disjunction free in the definition of [12], while it is not generalized disjunction free in the definition of [6]. In our opinion, it is reasonable to call every itemset with support 0 generalized disjunction free, since the support of all its supersets can trivially be derived to be equal to 0. Therefore, in this paper, we used the definitions from [6]. This difference explains the difference between the bound of [10] and ours, as for a set I of size k to be generalized disjunction free, there must not be 2^k transactions, but $2^k - 1$; $supp(I)$ itself can be 0. Therefore, I can only be generalized disjunction free according to [12] if $|\mathcal{D}| \geq 2^k - 1$. This gives the bound $\log_2(|\mathcal{D}| + 1)$, which still improves the bound $\log_2(|\mathcal{D}|) + 1$ given in [10].

Notice incidently that our bound on the l-free sets can easily be extended to a bound on frequent l-free sets, using a similar technique as in [10]. Let σ be the frequency threshold. A set I is frequent if at least σ transactions in the database contain all items of I. Therefore, for a set I of cardinality k to be frequent l-free, there need to be at least σ transactions containing all items of I, and 1 transaction for every other generalized itemset $X \cup \overline{Y}$ based on I. Hence,

$$I \text{ is } \sigma - \text{frequent } 1 - \text{free} \Rightarrow |\mathcal{D}| \geq \textstyle\sum_{i=0}^{l} \begin{pmatrix} k \\ i \end{pmatrix} + (\sigma - 1)$$

Another interesting link exists with [8]. In [8], the following question in the context of mining frequent itemsets with a standard levelwise approach is studied: given the current level and the current frequent itemsets, what is the maximal number of candidate itemsets that can be generated on the next level? The method described in [8] can be used at run-time to get ever better estimates on the size of the largest possible frequent itemset. Furthermore, the method also works for any collection of itemsets that is subset-closed. Hence, the results in [8] can also be used to get a run-time bound on the number and maximal cardinality of the l-free sets.

7 Conclusion and Future Work

In this paper, an upper bound on the size of the database $\mathcal{D}$, in terms of the size of the set I is found, indicating that whenever that bound is not exceeded, the set I is no l-free set. For the case that $l = k$ the bound simplifies yielding a expression in terms of $|I|$ and $|\mathcal{D}|$.

The aim of this work was to find a simple expression based only on the size of the set I and the number of transactions, to bound the cardinality of l-free sets. We found a reasonable approximation for the combinatorial bound that is easy to compute and useful for making conclusions. Because of the link between l-freeness and the other representations (freeness, disjunction freeness and generalized disjunction freeness) we can extend our results based on Proposition 1 and also make conclusions for these cases.

$$|\mathcal{D}| < k+1 \Rightarrow I \text{ is not free}$$
$$|\mathcal{D}| < \frac{k^2+k+2}{2} \Rightarrow I \text{ is not disjunction free}$$
$$|\mathcal{D}| < 2^k \Rightarrow I \text{ is not generalized disjunction free}$$

Interesting future work includes trying to find a statistical bound that, when we assume that all the items are independent and have probability p, tells us what the probability is that all combinations of generalized itemsets occur. Another interesting topic is when there exists correlation between the items.

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, 1993.
2. J.-F. Boulicaut, A. Bykowski, and C. Rigotti. Approximation of frequency queries by means of free-sets. In *Proc. PKDD*, pages 75–85, 2000.
3. A. Bykowski and C. Rigotti. A condensed representation to find frequent patterns. In *Proc. PODS*, 2001.
4. T. Calders. Deducing bounds on the frequency of itemsets. In *EDBT Workshop DTDM*, 2002.
5. T. Calders. *Axiomatization and Deduction Rules for the Frequency of Itemsets.* PhD thesis, University of Antwerp, Belgium, 2003.
6. T. Calders and B. Goethals. Minimal k-free representations of frequent sets. In *Proc. PKDD*, pages 71–82, 2002.
7. T. Calders and B. Goethals. Mining all non-derivable frequent itemsets. In *Proc. PKDD*, pages 74–85, 2002.
8. J. Van den Bussche F. Geerts, B. Goethals. A tight upper bound on the number of candidate patterns. In *Proc. ICDM*, pages 155–162, 2001.
9. M. Kryszkiewicz. Concise representation of frequent patterns based on disjunction-free generators. In *Proc. ICDM*, pages 305–312, 2001.
10. M. Kryszkiewicz. Upper bound on the length of generalized disjunction free patterns. In *SSDBM*, 2004.
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.