A Survey on Condensed Representations for Frequent Sets

Toon Calders[1], Christophe Rigotti[2], and Jean-François Boulicaut[2]

[1] University of Antwerp, Belgium
toon.calders@ua.ac.be
[2] INSA Lyon, LIRIS CNRS UMR 5205, France
{crigotti, jfboulicaut}@liris.cnrs.fr

Abstract. Solving inductive queries which have to return complete collections of patterns satisfying a given predicate has been studied extensively the last few years. The specific problem of frequent set mining from potentially huge boolean matrices has given rise to tens of efficient solvers. Frequent sets are indeed useful for many data mining tasks, including the popular association rule mining task but also feature construction, association-based classification, clustering, etc. The research in this area has been boosted by the fascinating concept of condensed representations w.r.t. frequency queries. Such representations can be used to support the discovery of every frequent set and its support without looking back at the data. Interestingly, the size of condensed representations can be several orders of magnitude smaller than the size of frequent set collections. Most of the proposals concern exact representations while it is also possible to consider approximated ones, i.e., to trade computational complexity with a bounded approximation on the computed support values. This paper surveys the core concepts used in the recent works on condensed representation for frequent sets.

1 Introduction

Knowledge Discovery in Databases (KDD) is a complex interactive and iterative process which involves many steps. Within the inductive database (IDB) framework, the needed data mining tasks are formalized as inductive queries which can be used to generate (mine), manipulate, and apply patterns [33,24]. The IDB framework is appealing because it employs *declarative* queries instead of ad-hoc *procedural constructs*. Since its introduction in [1], one of the most studied problems has been frequent itemset mining (FIM) and the popular post-processing of found itemsets into collections of association rules. Originally, this task has been dedicated to basket data analysis. Given a database of purchases or transactions, the association rule mining problem is to find associations between sets of products. In this context, the frequent itemsets correspond to sets of products that are often purchased together. Since then, the scope of association rule mining applications has been broadened towards many data analysis problems which are based on boolean or O/1 data (e.g., documents, WWW sessions, or microarray experiments can be considered as transactions whose items

J.-F. Boulicaut et al. (Eds.): Constraint-Based Mining, LNAI 3848, pp. 64–80, 2005.
© Springer-Verlag Berlin Heidelberg 2005

are respectively descriptors, uploaded resources or gene expression properties). Finding the frequent itemsets given a user-defined support threshold is not only the computationally most intensive step of association rule mining, but also it can be used for many other mining tasks, e.g., feature construction for classification or clustering methods. As such, the efficient extraction of frequent itemsets directly leads to significant performance improvements for many interactive KDD processes. It is indeed widely recognized that mining frequent itemsets should be one of the main operations supported by an inductive database management system.

The FIM problem has been studied as an inductive querying problem (see, e.g., [10]) and it is a prototypical task for which the general idea of condensed representations introduced in [39] has been proved extremely useful. The simple model introduced in [40] enables to abstract the semantics of inductive queries. Given a language $\mathcal{L}$ of patterns, the *theory* of a database $\mathcal{D}$ w.r.t. $\mathcal{L}$ and a selection predicate $\mathcal{C}$ is the collection $Th(\mathcal{D}, \mathcal{L}, \mathcal{C}) = \{\phi \in \mathcal{L} \mid \mathcal{C}(\phi, \mathcal{D}) = true\}$. The predicate selection or *constraint* $\mathcal{C}$ indicates whether a pattern ϕ is interesting or not. We say that computing $Th(\mathcal{D}, \mathcal{L}, \mathcal{C})$ is the evaluation for the inductive query $\mathcal{C}$ where $\mathcal{C}$ is defined as a boolean expression over some primitive constraints. The FIM problem concerns inductive queries where the data is a set of transactions (i.e., a potentially huge boolean matrix), the patterns are itemsets (i.e., sets of columns of the boolean matrix), and the constraint $\mathcal{C}$ is reduced to a minimal support constraint. In the first years, most of the research on the FIM problem has concentrated on extracting *all* frequent sets as efficiently as possible. Levelwise and depth-first search methods based on the anti-monotonicity of minimal support, and efficient data structures have been studied. Since the first algorithm AIS [1], there have been important historical gains on performance such as: improving pruning (Apriori [2]) and counting (e.g., Partition [48], Sampling [49]), reducing the number of database scans (e.g., DIC [15]), and avoiding explicit candidate generation (e.g., FP-Growth [32]). This list is not exhaustive, and it should also be noticed that these approaches are often based on a mix of several improvements. Often, however, the number of frequent itemsets is so huge that their storage and support counting require unrealistic resources. This blow-up happens, for example, when we set the support threshold too low, or when the data is heavily correlated. Indeed, in the worst case, the number of frequent itemsets can be exponential in the number of items. Even though typical basket data is sparse and weakly correlated, many new applications of FIM have turned to be computationally too hard.

One solution to this problem relies on the *condensed representation* principle. The idea is to compute $\mathcal{CR} \subseteq \mathcal{L}$ which might be as concise as possible such that deriving $Th(\mathcal{D}, \mathcal{L}, \mathcal{C})$ from $\mathcal{CR}$ can be performed efficiently. In the context of huge database mining, efficiently means without any further access to $\mathcal{D}$. Using border sets [40], e.g., the maximal frequent itemsets for FIM [5], might be considered as a good solution: all the subsets of the maximal frequent itemsets are frequent itemsets (i.e., this condensed representation is a proper subset of the theory) and can be derived without looking at the data. In most of the applications of

FIM, however, the user wants not only the collection of the frequent patterns but also their supports (e.g., to compute association rule interestingness measures like the confidence values). Now, a condensed representation $\mathcal{CR}$ must enable to regenerate not only the patterns, but also the values of an evaluation function like the support without accessing the data. If the regenerated values are only approximated, the condensed representation is called *approximate*. Otherwise, it is called an *exact* condensed representation. For a condensed representation, different characteristics determine its usefulness, depending on the application area. It is clear that good characteristics are: the size of the representation (theoretically and in practice), the efficiency, and the completeness of the algorithms which compute these representations, the fast and complete generation of useful information from the representation (e.g., all the frequent itemsets and their supports, relevant association rules).

Starting from the formalization of ϵ-adequate representations [39] and its first concrete application to FIM in [11], many useful condensed representations have been designed over the last 5 years. The main objective of this survey is to present, in a synthetic way, the core concepts used in the recent works on condensed representation for frequent itemsets, including: *Closed Sets* [55,43,44,11] *δ-Free Sets* [12,13], *Disjunction-Free Sets* [17,18], *Generalized Disjunction-Free Sets* [37], *Non-Derivable Itemsets* [20], and the unified framework presented in [21].

The organization of the paper is as follows. In the next section, we recall some preliminary definitions. Then, we present several condensed representations in Sections 3 to 6. Section 7 concerns a recent framework which provides a unified view of most of these representations. Section 8 provides pointers to representative algorithms for computing condensed representations. Section 9 gives complementary bibliographic information concerning applications. Finally, Section 10 is a short conclusion.

2 Preliminary Definitions

The FIM problem is by now well known [1]. We are given a set of items $\mathcal{I}$ and a database $\mathcal{D}$ of subsets of $\mathcal{I}$ (to allow duplicates, $\mathcal{D}$ can be defined as a multi-set). The elements of $\mathcal{D}$ are called *transactions*. An *itemset* $I \subseteq \mathcal{I}$ is a set of items; its *support* in $\mathcal{D}$, denoted $supp(I, \mathcal{D})$, is defined as the number of transactions in $\mathcal{D}$ that contain all items of I. An itemset is called *σ-frequent* in $\mathcal{D}$ if its support in $\mathcal{D}$ exceeds σ. The goal is now, given a minimal support threshold and a database, to compute the collection $\mathcal{F}(\mathcal{D}, \sigma)$ of all frequent itemsets and their supports. We denote itemsets by strings, e.g., *abcd* denotes the set $\{a, b, c, d\}$.

The presentation of most of the condensed representations needs for the concept of *negative border* introduced in [40]. The negative border of a collection of itemsets $\mathcal{J}$, denoted $\mathcal{B}d^-(\mathcal{J})$ is the collection $\{X | X \subseteq \mathcal{I} \wedge X \notin \mathcal{J} \wedge (\forall Y \subset X, Y \in \mathcal{J})\}$. Intuitively, $\mathcal{B}d^-(\mathcal{J})$ contains the smallest itemsets not in $\mathcal{J}$. For instance, $\mathcal{B}d^-(\mathcal{F}(\mathcal{D}, \sigma))$ denotes the collection of the smallest (w.r.t. set inclusion) infrequent itemsets.

The last notion that we recall in this section, is *anti-monotonicity*. It is a commonly used property leading to safe pruning criteria and efficient pattern mining (e.g., [40]). A property ρ is anti-monotonic if and only if for all itemsets X and Y, $\rho(X)$ and $Y \subseteq X$ implies $\rho(Y)$. Clearly, the minimal support property is anti-monotonic.

3 Closed Sets

This representation is based on the notion of *closed set* used in *formal concept analysis* [51,28], a branch of lattice theory dedicated to the study of the lattice structure induced by a binary relation (structure called *Galois lattice* or *concept lattice*).

The application of this theory to frequent itemset mining has been proposed independently by Pasquier et al. in [43,44] and by Zaki and Ogihara in [55].

In this context, an itemset I is said to be *closed* in $\mathcal{D}$ if and only if no proper superset of I has the same support than I in $\mathcal{D}$. The *closure* of an itemset I in $\mathcal{D}$, denoted $cl(I)$, is the unique maximal superset of I having the same support than I and a closed itemset is equal to its closure. One elegant alternative definition is to consider the equivalence classes of the itemsets appearing in the same sets of transactions, i.e., the equivalence classes of the relation "has the same closure": closed itemsets are the unique maximal elements of each equivalence class [4].

For a given support threshold, it is thus sufficient to know the collection of all frequent closed itemsets (denoted *FreqClosed*) and their supports, to be able to generate all the frequent itemsets and their supports, i.e., $\mathcal{F}$. For example, consider an itemset X, if X has no superset in *FreqClosed*, this means that $cl(X)$ is not frequent, and thus X can not be frequent. If X has at least one superset in *FreqClosed*, then $supp(X) = supp(Y)$ where $Y = cl(X)$ is the smallest superset of X in *FreqClosed*.

Let us consider the database containing the following transactions: two transactions $\{a, b\}$, two transactions $\{a, b, c, d\}$ two transactions $\{a, b, c, d, e\}$ and one transaction $\{a, b, c, d, e, f\}$ (see Table 1).

In such a database, for example, the itemset *abc* is not closed, since it has the same support (i.e., 5 transactions) than *abcd*, one of its proper supersets.

Table 1. A toy database

	Items					
Trans.	a	b	c	d	e	f
t_1	1	1	0	0	0	0
t_2	1	1	0	0	0	0
t_3	1	1	1	1	0	0
t_4	1	1	1	1	0	0
t_5	1	1	1	1	1	0
t_6	1	1	1	1	1	0
t_7	1	1	1	1	1	1

The itemset *abcd* is the maximal superset of *abc* having the same support, and thus is the closure of *abc*. If we choose a support threshold of 2 transactions, then the frequent closed sets are *ab*, *abcd*, *abcde* and their respective supports are 7, 5 and 3. Having only at hand these frequent closed sets, to generate the support of *abc* we consider the smallest frequent closed set that is a superset of *abc*. This frequent closed set is *abcd* and its support (i.e., 5 transactions) gives us the support of *abc*.

4 Free Sets

The *free sets* (also termed δ-*free sets*) have been introduced in [12,13] and are based on the notion of δ-*strong rule*[1]. Informally, a δ-strong rule is an association rule of the form $X \Rightarrow^{\delta} a$, where $X \subseteq \mathcal{I}$, $a \in \mathcal{I} \setminus X$, and δ is a natural number. This rule is valid in a database if $supp(X) - supp(X \cup \{a\}) \leq \delta$, i.e., the rule is violated in no more than δ transactions. Since δ is supposed to be small w.r.t. $|\mathcal{D}|$, δ-strong rules have a high confidence (in particular confidence 1 when $\delta = 0$).

An itemset $Y \subseteq \mathcal{I}$ is a δ-free set if and only if there is no valid δ-strong rule $X \Rightarrow^{\delta} a$ such that $X \subset Y$, $a \in Y$ and where by definition $a \notin X$.

The set of all frequent δ-free sets, denoted $FreqFree^{\delta}$, and their supports enables to approximate the support of the frequent non-δ-free sets. Let us consider Y a frequent non-δ-free set. Then, there exists a valid δ-strong rule $X \Rightarrow^{\delta} a$ such that $X \subset Y$ and $a \in Y$. Moreover, $Y \setminus \{a\} \Rightarrow^{\delta} a$ is also valid. Thus the support of Y can be approximated by the support of the frequent set $Y \setminus \{a\}$ (more precisely, this support is an upper bound of $supp(Y)$). If $Y \setminus \{a\}$ is a free-set then we have its support, if not, it can be in turn approximated by the support of a smaller itemset. This recursive process gives an approximation of the support of Y. Using this principle, the best approximation is the lowest upper bound. Thus, in practice, the support of Y is approximated by the minimal support value of the frequent δ-free sets that are subsets of Y. The error made has been formalized using the framework of an ϵ-adequate representation [39], and is small on common real datasets [13].

When $\delta = 0$, the support of all frequent non-δ-free sets can be determined exactly. In fact, the 0-free sets corresponds to the *key patterns* (also called *generators*) developed independently in [4], and also used in other works, such as [36]. The following property mentioned by several authors (e.g., [4]) establishes a direct link between 0-free sets and closed sets: any frequent closed sets is the closure of at least one frequent 0-free sets. As a result, when considering each (frequent) 0-free set X, $cl(X)$ is a (frequent) closed set but also $X \Rightarrow cl(X) \setminus X$ is an association rule with confidence 1. In fact, 0-free sets are the minimal elements of the already mentioned equivalence classes. Since several minimal elements are possible, collections of 0-free sets are generally larger than collections of closed sets. In our toy example from Table 1, the 2-frequent 0-free sets are $\emptyset$, c, d and e.

Even though the frequent δ-free sets are sufficient to approximate the support of all frequent non-δ-free sets (or to determine this support exactly when $\delta = 0$),

[1] Stemming from the notion of *strong rule* of [46].

they are not sufficient to decide whether an itemset is frequent or not. For this purpose, the collection of frequent δ-free sets is completed by the collection of minimal infrequent δ-free itemsets, that can be defined as $\mathcal{B}d^-(\mathit{FreqFree}^\delta) \cap \mathit{Free}^\delta$, where Free^δ is the collection of δ-free sets. Now, given any itemset Y, if there exists $Z \subseteq Y$, such that Z is a minimal infrequent δ-free itemsets, then we know that Y is not frequent. In the other case, the support of Y can be approximated as described above.

5 Disjunction-Free Sets

5.1 Simple Disjunction-Free Sets

This representation has been proposed in [17,18] as a generalization of 0-free sets. It is based on *disjunctive rules* of the form $X \Rightarrow a \vee b$, where $X \subseteq \mathcal{I}$ and $a, b \in \mathcal{I} \setminus X$. Such a rule is said to be valid if any transaction containing X contains also a or b (maybe both).

Thus the support of X is equal to the sum of $\mathit{supp}(X \cup \{a\})$ and $\mathit{supp}(X \cup \{b\})$ minus $\mathit{supp}(X \cup \{a, b\})$ since the transactions containing $X \cup \{a, b\}$ have been counted both in $\mathit{supp}(X \cup \{a\})$ and $\mathit{supp}(X \cup \{b\})$. So, we have the relation $\mathit{supp}(X \cup \{a, b\}) = \mathit{supp}(X \cup \{a\}) + \mathit{supp}(X \cup \{b\}) - \mathit{supp}(X)$ and the satisfaction of this relation is equivalent to the validity of the rule $X \Rightarrow a \vee b$.

Similarly to δ-free sets, an itemset $Y \subseteq \mathcal{I}$ is a *disjunction-free set* if and only if there is no valid disjunctive rule $X \Rightarrow a \vee b$, such that $X \subset Y$, $a, b \in Y$ and where by definition $a \notin X$ and $b \notin X$. In the following, the collection of all frequent disjunction-free sets is denoted *FreqDFree*.

Knowing all elements in *FreqDFree* and their supports is not sufficient to determine the support of all frequent itemsets. For that purpose the representation can be completed in different ways. The representation based on disjunction-free sets proposed in [17] has been revisited in [36] and [18], leading to reduce the size of this border[2].

Intuitively, *FreqDFree* must be completed with the collection of all the valid rules of the form $X \Rightarrow a \vee b$, where $X \in \mathit{FreqDFree}$ and $X \cup \{a, b\}$ is frequent. This can be illustrated inductively as follows. Suppose that using *FreqDFree* (and the supports of its elements) and the collection of rules defined above, we are able to compute the support of any itemset having a size lesser or equal to k. Let us consider a frequent itemset Y such that $|Y| = k + 1$. If Y is disjunction-free then $Y \in \mathit{FreqDFree}$ and we know its support. If Y is not disjunction-free, then there exists a valid rule $X \Rightarrow a \vee b$ such that $X \subset Y$ and $a, b \in Y$. By definition of a valid rule, $Y \setminus \{a, b\} \Rightarrow a \vee b$ is also valid. Hence the relation $\mathit{supp}(Y) = \mathit{supp}(Y \setminus \{b\}) + \mathit{supp}(Y \setminus \{a\}) - \mathit{supp}(Y \setminus \{a, b\})$ holds. Since Y is frequent, the itemsets $Y \setminus \{b\}$, $Y \setminus \{a\}$ and $Y \setminus \{a, b\}$ are also frequent. Moreover, these three sets have a size strictly lesser than $k + 1$. Thus, by hypothesis, we can determine their supports, and then compute $\mathit{supp}(Y)$.

[2] The core part of the representation, i.e. the frequent disjunction-free sets (called frequent disjunction-free generators in [36]), remains the same.

5.2 Generalized Disjunction-Free Sets

The generalization of disjunction-free sets towards rules of the form $X \Rightarrow a_1 \vee \ldots \vee a_i \vee \ldots \vee a_n$, has been suggested in [17,18], and explored in [37]. In this context, an itemset X is a generalized disjunction-free set if and only if for any value of $n > 0$, there is no valid rule $X \setminus \{a_1, \ldots, a_i, \ldots, a_n\} \Rightarrow a_1 \vee \ldots \vee a_i \vee \ldots \vee a_n$, where $\{a_1, \ldots, a_i, \ldots, a_n\} \subseteq X$.

6 Non-derivable Itemsets

In [20], the *non-derivable itemsets* (NDIs) were introduced as a new condensed representation. The NDIs rely on a complete set of deduction rules that derive bounds on the support of an itemset. In this section, we first discuss the deduction rules, and then introduce the representation based on these rules.

6.1 Deduction Rules

In [20], formulas to bound the support of an itemset I, based on the supports of its subsets were introduced. For all $X \subseteq I$, the following rule holds:

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

This rule is denoted $\mathcal{R}_X(I)$. The rules are based on the inclusion-exclusion principle [27]. For a proof of the rules, see [19]. Depending on the subset X of 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 $X \subseteq I$.

Notice that these rules reflect the monotonicity principle. Let $i \in I$, then $\mathcal{R}_{I \setminus \{i\}}(I)$ is the following rule:

$$supp(I) \leq supp(I \setminus \{i\}) \ .$$

In Figure 1, all rules $\mathcal{R}_X(I)$, for $I = abcd$ and $X \subseteq I$ are given.

We denote the greatest lower bound on I by $LB(I)$ and the least upper bound by $UB(I)$. In practice it occurs often that $LB(I) = UB(I)$. Such a set I is called a *derivable itemset (DI)*, since we know without counting its support in the database, that $supp(I) = LB(I) = UB(I)$. In [20] it was shown that derivability is monotonic. Hence, if a set I is derivable, then are all its supersets.

Another interesting property proven in [20] is that for a non-derivable itemset, the interval width, that is, $w(I) := UB(I) - LB(I)$, decreases exponentially in $|I|$. Thus, $w(I \cup \{j\}) \leq w(I)/2$, for every itemset I and item j not in I. This property guarantees that non-derivable itemsets cannot be very large, because the intervals can only be halved a logarithmic number of times.

$$
\begin{array}{lll}
supp(abcd) \geq supp(abc) + supp(abd) + supp(acd) + supp(bcd) & & \\
\quad -supp(ab) - supp(ac) - supp(ad) - supp(bc) & & \mathcal{R}_{\emptyset}(abcd) \\
\quad -supp(bd) - supp(cd) + supp(a) + supp(b) & & \\
\quad +supp(c) + supp(d) - supp(\emptyset) & & \\
supp(abcd) \leq supp(a) - supp(ab) - supp(ac) - supp(ad) & & \mathcal{R}_{a}(abcd) \\
\quad +supp(abc) + supp(abd) + supp(acd) & & \\
supp(abcd) \leq supp(b) - supp(ab) - supp(bc) - supp(bd) & & \mathcal{R}_{b}(abcd) \\
\quad +supp(abc) + supp(abd) + supp(bcd) & & \\
supp(abcd) \leq supp(c) - supp(ac) - supp(bc) - supp(cd) & & \mathcal{R}_{c}(abcd) \\
\quad +supp(abc) + supp(acd) + supp(bcd) & & \\
supp(abcd) \leq supp(d) - supp(ad) - supp(bd) - supp(cd) & & \mathcal{R}_{d}(abcd) \\
\quad +supp(abd) + supp(acd) + supp(bcd) & & \\
supp(abcd) \geq supp(abc) + supp(abd) - supp(ab) & & \mathcal{R}_{ab}(abcd) \\
supp(abcd) \geq supp(abc) + supp(acd) - supp(ac) & & \mathcal{R}_{ac}(abcd) \\
supp(abcd) \geq supp(abd) + supp(acd) - supp(ad) & & \mathcal{R}_{ad}(abcd) \\
supp(abcd) \geq supp(abc) + supp(bcd) - supp(bc) & & \mathcal{R}_{bc}(abcd) \\
supp(abcd) \geq supp(abd) + supp(bcd) - supp(bd) & & \mathcal{R}_{bd}(abcd) \\
supp(abcd) \geq supp(acd) + supp(bcd) - supp(cd) & & \mathcal{R}_{cd}(abcd) \\
supp(abcd) \leq supp(abc) & & \mathcal{R}_{abc}(abcd) \\
supp(abcd) \leq supp(abd) & & \mathcal{R}_{abd}(abcd) \\
supp(abcd) \leq supp(acd) & & \mathcal{R}_{acd}(abcd) \\
supp(abcd) \leq supp(bcd) & & \mathcal{R}_{bcd}(abcd) \\
supp(abcd) \geq 0 & & \mathcal{R}_{abcd}(abcd)
\end{array}
$$

Fig. 1. Tight bounds on $supp(abcd)$

The size 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 may require a lot of resources, in practise only rules of limited depth are used. The greatest lower and least upper bounds on the support of I resulting from evaluation of rules up to depth k are denoted $LB_k(I)$ and $UB_k(I)$. Hence, the interval $[LB_k(I), UB_k(I)]$ is formed by the bounds calculated by the rules $\{\mathcal{R}_X(I) \mid X \subseteq I, |I \setminus X| \leq k\}$.

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

$$
\begin{array}{lll}
supp(abc) & \geq & 0 \\
& \leq & supp(ab) = 2 \\
& \leq & supp(ac) = 2 \\
& \leq & supp(bc) = 2 \\
& \geq & supp(ab) + supp(ac) - supp(a) = 0 \\
& \geq & supp(ab) + supp(bc) - supp(b) = 0 \\
& \geq & supp(ac) + supp(bc) - supp(c) = 0 \\
& \leq & supp(ab) + supp(ac) + supp(bc) \\
& & \quad -supp(a) - supp(b) - supp(c) + supp(\emptyset) = 1
\end{array}
$$

These rules are $\mathcal{R}_{abc}(X)$ when X is respectively abc, ab, ac, bc, a, b, c, and $\emptyset$. 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, $LB_0(abc) = 0$, $LB_2(abc) = 0$, $UB_1(abc) = 2$, $UB_3(abc) = 1$. The interval width for abc is $UB(abc) - LB(abc) = 1$.

For ab, we have the following rules:

$$supp(ab) \geq 0 \qquad supp(ab) \leq supp(a) = 4$$
$$supp(ab) \geq supp(a) + supp(b) - supp(\emptyset) = 1 \qquad supp(ab) \leq supp(b) = 4$$

Therefore, $LB(ab) = 1$, and $UB(ab) = 4$. The interval width for ab is 3. Notice that the interval width for abc is indeed less than half of the interval width for ab.

6.2 Representation Based on Deduction Rules

In [20], the NDI representation was introduced, based on the deduction rules. The NDI-representation is defined as follows:

$$NDIRep(\mathcal{D}, \sigma) := \{(I, supp(I, \mathcal{D})) \mid supp(I, \mathcal{D}) \geq \sigma, LB(I) \neq UB(I)\}$$

From *NDIRep*, for every set I it can be decided whether or not it is frequent, and if it is frequent, its support can be derived. This can be seen as follows: every itemset I that is not in *NDIRep* is either infrequent, or derivable (or both). We calculate and compare the bounds $LB(I)$ and $UB(I)$. If they are not equal, I must be infrequent (otherwise I would have been in *NDIRep*). If they are equal, then we know $supp(I) = LB(I) = UB(I)$. In order to calculate the bounds on the support of I, however, we need to know the support of all subsets of I. This can be done in an iterative way; first we calculate the bounds on the subsets of I that are in the border of *NDIRep*. For these subsets, the bounds can be calculated. If one of them is infrequent, I must be infrequent as well. Otherwise, we know the supports of all subsets of I in the border of *NDIRep*. Subsequently, we can calculate bounds on the subsets of I that are just above the border, and so on, until either the supports of all subsets of I are known and we can calculate the bounds for I, or one of the subsets turned out to be infrequent.

7 Unified View

In [21], a unified view of 0-freeness, disjunction-freeness and non-derivability was given. In this framework, the notion of a k-free[3] set is central, as it captures different properties in several previously studied exact condensed representations. It was shown that the different representations can be described as a main component, that is based on frequent k-free, and a border. We now describe the main ideas of this unified view.

7.1 *k*-Free Sets

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

[3] Notice that the k-free sets are different from the δ-free sets of Section 5.

Definition 1.
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 property states, these definitions cover freeness, disjunction-freeness, and generalized disjunction-freeness.

Property 1. [21] Let I be a frequent itemset.

- I is free (δ-free with $\delta = 0$) 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.

The next property forms the basis of the representations based on k-free sets.

Property 2. k-freeness is anti-monotonic; if a set I is k-free, then all its subsets are k-free as well. Moreover, if $supp(J) = LB_k(J)$ (resp. $supp(J) = UB_k(J)$), then also $supp(I) = LB_k(I)$ (resp. $supp(I) = UB_k(I)$), for all $J \subseteq I$.

The frequent k-free sets together with the border, that is, the collection

$$\{(I, supp(I)) \mid I \in FFree_k\} \cup \{(J, supp(J)) \mid \forall j \in J : J \setminus \{j\} \in FFree_k\} ,$$

forms a condensed representation. It can be shown by induction that for every itemset I, we can derive whether or not it is frequent, and if it is frequent, we can find its support. For the sets that are frequent and k-free or that are in the border, the support is known because they are in the representation. Next, let I be a set such that all its subsets are in the representation. Then the support of all subsets of I is known, as they are all in the representation. Also, I has at least one subset J in the border of the k-free sets (otherwise I would have been in the border itself, and thus in the representation). If J is infrequent, then I is as well. Otherwise, $supp(J)$ is either $LB_k(J)$ or $UB_k(J)$. Suppose that $supp(J) = LB_k(J)$. Then we know from Property 2 that also $supp(I) = LB_k(I)$. Since the support of all subsets of I are known, we can calculate $LB_k(I)$, and thus we can derive the support of I. Hence, for all itemsets that contain only one more item than the sets in the representation, we can find the support. We can now iteratively repeat this procedure to find the sets that contain two more items, three more items, and so on, until we have found all frequent itemsets.

7.2 Groups in the Border

Let us recall from Section 5 that frequent free sets alone do not form a condensed representation. In order to have a condensed representation, part of the border need to be stored as well. For disjunction-free and generalized disjunction-free sets, parts of the border are needed as well. The reason that some of the sets of the border are needed is because otherwise it is impossible to tell why the sets are not in the representation. For example, for the disjunction-free sets, were

they left out because the were infrequent, or because they were not disjunction-free? And if they are not disjunction-free, what rule should be used to derive the support? Because of the anti-monotonicity of both frequency and disjunction-freeness, it suffices to store only the sets on the border; if we know them, we know the rest as well; either the set on the border is infrequent, and then are all its supersets as well, or it is not disjunction-free with a certain rule, and in that case, its supersets are not disjunction-free as well, because of the same rule.

In general, as we illustrated in the previous subsection, this situation applies for k-free sets as well. Again, some elements of the border are needed to have a condensed representation.

In [21], a systematic study of which parts of the border are really needed was made. The border of the frequent k-free sets can be divided into different parts, based on the deduction rules. For example: the group of infrequent sets in the border, the group of sets I with $supp(I) \neq LB_1(I)$, or the group of frequent sets with $supp(I) = LB_\infty(I)$. In this way the existing representations could be improved by storing a smaller part of the border.

7.3 Relations Between the Different Representations

From the unified view of the different representations, many relations between the representations can be derived. In fact, the k-free based representations form an interesting hierarchy. The higher k is, the more complex the representation becomes, but at the same time, the more concise. For example, the disjunction-free sets are based on the 2-free sets, while the non-derivable itemsets are based on the ∞-free sets. Henceforth, on the one hand, the NDI-representation is more concise than the disjunction-free representation, but on the other hand, it can be far more costly to compute it and to derive the support of the sets which are not in the collection [21].

8 Algorithms

Many algorithms and variants have been proposed to extract condensed representations for frequent itemsets. The main principles are similar to the ones that have been proposed for the extraction of frequent itemsets. This includes two main aspects, firstly the strategy used to explore the pattern space and secondly the representation of the database used to count the support of the patterns.

Nearly all algorithms start the exploration from the empty itemset and go towards larger ones. This is performed either in a levelwise way (i.e., considering all patterns of size n and then all patterns of size $n + 1$) or using a depth-first approach. For the counting steps, three main representations have been adopted. The first one called *horizontal database layout* is a very natural one, in which the database is handle as a list of transactions. The second is based on a *vertical database layout* representation, so that for each pattern the algorithms store the identifiers of the transactions in which this pattern occur. Such a list, called *occurrence list* or *tid-list*, are used to count the support of the pattern and also to generate the occurrence lists of longer patterns. And finally, the third approach

that relies on *projected databases*, which contain in a compact way, subsets of the data needed to explore sub-spaces of the whole pattern space.

The main representative algorithms are a combination of these exploration strategies and database representations. The levelwise strategy is used together with an horizontal database layout to extract the closed sets by the algorithms Close [44] and Pascal [4], and also to mine the δ-free sets [12,13], the disjunction-free sets [17,18] (algorithm HLinHex) and the NDIs [20]. The depth-first strategy and a projected database approach are combined in the Closet [45] and VLinHex [17,18] algorithms to find respectively closed itemsets and disjunction-free sets. The vertical database layout has been used conjointly to a depth first exploration in the Charm [54] and the dfNDI [22] algorithms.

Beyond the usual pruning based on support, the various algorithms used pruning conditions stemming from properties of the different condensed representations (e.g., anti-monotonicity of freeness) to reduce the search space. It should be noticed that a major effort has been made to obtain efficient implementations (see, e.g., the first and second Workshop on Frequent Itemset Mining Implementations [31,6]).

9 Applications

Our goal is not to provide an exhaustive list of applications of condensed representations of frequent sets. Instead, we want to point out some typical examples of such works.

It is obvious that condensed representations of frequent sets can be used for any application of frequent sets: frequent sets and their supports are just computed faster from dense and/or correlated data. It is however important to notice that, when condensed representations enable a high condensation, the regeneration process might fail due to the size of the complete collection of the frequent sets. Therefore, it makes sense either to use condensed representations as cache mechanisms and/or to derive relevant patterns directly from the condensed representations. For instance, it is possible to provide summaries or even covers of large collections of association rules [53,3,30]. One typical application has been considered in [7] where 0-free sets and their closures are computed from a boolean gene expression data set. One can also point out the generation of a synthetic view of rule confidence variations from disjunction-free sets [16]. The recent Ph.D thesis [41] studies summarization techniques for large collections of patterns and thus many applications of condensed representations. Association-based classification (see, e.g., [38]) can also benefit from condensed representations. For instance, using δ-strong association rules built on δ-free itemsets and their closures has been proved useful in this context [23]. It is also possible to exploit condensed representations as patterns for themselves, e.g., closed sets in boolean gene expression data sets correspond to putative synexpression groups or transcription modules [8].

Condensed representations can be used for optimizing not only one inductive query on sets but also sequences of queries on set patterns [34,29]. One condensed

representation can also be used as an intermediate representation to mine efficiently another one (see, e.g., the generation of closed sets from disjunction-free sets [18]). Related to inductive querying on sets, one interesting issue concerns condensed representation mining when the minimal support constraint is not the only constraint. This has been considered, e.g., for free sets in [14] and for closed sets in [9].

Finally, we have removed maximal frequent itemsets from consideration while it can be useful for some applications where the support of every frequent itemset is not needed, e.g., feature construction. Indeed, border sets have many applications. For instance, border sets have been studied extensively in the context of conjunctions of minimal support and maximal support constraints (see, e.g., [25]).

10 Conclusion and Perspectives

This paper has surveyed the core concepts used in the recent works on condensed representations for frequent sets. These concepts have been proved extremely useful not only for an algorithmic breakthrough concerning the many applications of frequent set mining but also for deriving more useful patterns, e.g., covers of association rules. An important direction of work, is the detailed comparison of practical pros and cons of the different representations. This includes fair experiments on representative real data sets, to compare (1) the representation sizes (in number of patterns, and also their true sizes in bytes) and (2) their related time costs, (not only for their extractions, but also for the generation of patterns like frequent itemsets, rule covers, and for the derivation of other condensed representations). All the condensed representations mentioned in this paper are based on equality or inequality relations on itemset supports. Similar relation on support have been used by other authors in different contexts, e.g., for the approximation of the support of itemsets with negation in [39]. It might be interesting to consider whether the state-of-the-art in condensed representations enables or not to consider new data mining tasks based on, e.g., association rule with negated items.

The condensed representation principle can be applied for many other pattern domains and more sophisticated types of inductive queries. For instance, a similar concept of freeness has been studied for functional dependency discovery [42] and various condensed representations have been studied recently for frequent sequences, trees or graphs (see, e.g., [50,47,52]). It can be also studied w.r.t. quite general forms of inductive queries which are arbitrary boolean combinations of some primitive constraints. The results on using collections of version spaces as condensed representations for queries that involve arbitrary combinations of monotonic and anti-monotonic constraints provides an interesting starting point [26]. Also, the relationship between condensed representations and witnesses [35] might be explored.

As a conclusion, starting from efficient solutions to the Frequent Itemset Mining problem, the notion of condensed representation has been identified as a core

concept for inductive query optimization and its interest goes far beyond simple KDD processes based on itemsets, say standard association rule mining. We are pretty confident that this will become one major topic for research in the next few years, either for innovative applications of frequent pattern mining or for new pattern domains.

References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. ACM Int. Conf. on Management of Data SIGMOD'93*, pages 207–216, Washington, D.C., USA, May 1993. ACM Press.
2. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proc. Int. Conf. on Very Large Data Bases VLDB'94*, pages 487–499, Santiago de Chile, Chile, Sept. 1994. Morgan Kaufmann.
3. Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, and L. Lakhal. Mining minimal non-redundant association rules using frequent closed itemsets. In *Proc. Int. Conf. on Deductive and Object-Oriented Databases DOOD'00*, volume 1861 of *LNCS*, pages 972–986, London, UK, July 2000. Springer-Verlag.
4. Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, and L. Lakhal. Mining frequent patterns with counting inference. *SIGKDD Explorations*, 2(2):66–75, Dec. 2000.
5. R. J. Bayardo. Efficiently mining long patterns from databases. In *Proc. ACM Int. Conf. on Management of Data SIGMOD'98*, pages 85–93, Seattle, USA, June 1998. ACM Press.
6. R. J. Bayardo, B. Goethals, and M. J. Zaki, editors. *Proc. Int. Workshop on Frequent Itemset Mining Implementations FIMI'04*, Brighton, UK, Nov. 2004.
7. C. Becquet, S. Blachon, B. Jeudy, J.-F. Boulicaut, and O. Gandrillon. Strong association rule mining for large gene expression data analysis: a case study on human SAGE data. *Genome Biology*, 12, 2002.
8. J. Besson, C. Robardet, J.-F. Boulicaut, and S. Rome. Constraint-based bi-set mining for biologically relevant pattern discovery in microarray data. *Intelligent Data Analysis*, 9(1):59–82, 2005.
9. F. Bonchi and C. Lucchese. On closed constrained frequent pattern mining. In *Proc. IEEE Int. Conf. on Data Mining ICDM'04*, pages 35–42, Brighton, UK, Nov. 2004. IEEE Computer Press.
10. J.-F. Boulicaut. Inductive databases and multiple uses of frequent itemsets: the cInQ approach. In *Database Technologies for Data Mining - Discovering Knowledge with Inductive Queries*, volume 2682 of *LNCS*, pages 1–23. Springer-Verlag, 2004.
11. J.-F. Boulicaut and A. Bykowski. Frequent closures as a concise representation for binary data mining. In *Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining PAKDD'00*, volume 1805 of *LNAI*, pages 62–73, Kyoto, JP, Apr. 2000. Springer-Verlag.
12. J.-F. Boulicaut, A. Bykowski, and C. Rigotti. Approximation of frequency queries by mean of free-sets. In *Proc. Principles and Practice of Knowledge Discovery in Databases PKDD'00*, volume 1910 of *LNAI*, pages 75–85, Lyon, F, Sept. 2000. Springer-Verlag.
13. J.-F. Boulicaut, A. Bykowski, and C. Rigotti. Free-sets: a condensed representation of boolean data for the approximation of frequency queries. *Data Mining and Knowledge Discovery journal*, 7(1):5–22, 2003.

14. J.-F. Boulicaut and B. Jeudy. Mining free itemsets under constraints. In *Proc. Int. Database Engineering and Application Symposium IDEAS'01*, pages 322–329, Grenoble, F, July 2001. IEEE Computer Press.
15. S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proc. ACM Int. Conf. on Management of Data SIGMOD'97*, pages 255–264, Tucson, USA, May 1997. ACM Press.
16. A. Bykowski, T. Daurel, N. Méger, and C. Rigotti. Integrity constraints over association rules. In *Database Technologies for Data Mining - Discovering Knowledge with Inductive Queries*, volume 2682 of *LNCS*, pages 311–330. Springer-Verlag, 2004.
17. A. Bykowski and C. Rigotti. A condensed representation to find frequent patterns. In *Proc. ACM Symposium on Principles of Database Systems PODS'01*, pages 267–273, Santa Barbara, CA, USA, May 2001. ACM Press.
18. A. Bykowski and C. Rigotti. DBC: A condensed representation of frequent patterns for efficient mining. *Information Systems*, 28(8):949–977, 2003.
19. T. Calders. Deducing bounds on the support of itemsets. In *Database Technologies for Data Mining - Discovering Knowledge with Inductive Queries*, volume 2682 of *LNCS*, pages 214–233. Springer-Verlag, 2004.
20. T. Calders and B. Goethals. Mining all non derivable frequent itemsets. In *Proc. Principles and Practice of Knowledge Discovery in Databases PKDD'02*, volume 2431 of *LNAI*, pages 74–85, Helsinki, FIN, Aug. 2002. Springer-Verlag.
21. T. Calders and B. Goethals. Minimal k-free representations of frequent sets. In *Proc. Principles and Practice of Knowledge Discovery in Databases PKDD'03*, volume 2838 of *LNAI*, pages 71–82, Cavtat-Dubrovnik, HR, Sept. 2003. Springer-Verlag.
22. T. Calders and B. Goethals. Depth-first non derivable itemset mining. In *Proc. SIAM Int. Conf. on Data Mining SDM'05*, Newport Beach, USA, Apr. 2005.
23. B. Crémilleux and J.-F. Boulicaut. Simplest rules characterizing classes generated by delta-free sets. In *Proc. BCS Int. Conf. on Knowledge Based Systems and Applied Artificial Intelligence ES'02*, pages 33–46, Cambridge, UK, Dec. 2002. Springer-Verlag.
24. L. De Raedt. A perspective on inductive databases. *SIGKDD Explorations*, 4(2):69–77, 2003.
25. L. De Raedt. Towards query evaluation in inductive databases using version spaces. In *Database Technologies for Data Mining - Discovering Knowledge with Inductive Queries*, volume 2682 of *LNCS*, pages 117–134. Springer-Verlag, 2004.
26. L. De Raedt, M. Jaeger, S. D. Lee, and H. Mannila. A theory of inductive query answering. In *Proc. IEEE Int. Conf. on Data Mining ICDM'02*, pages 123–130, Maebashi City, JP, Dec. 2002. IEEE Computer Press.
27. J. Galambos and I. Simonelli. *Bonferroni-type Inequalities with Applications.* Springer, 1996.
28. B. Ganter and R. Wille. *Formal Concept Analysis, Mathematical Foundations.* Springer-Verlag, 1999.
29. A. Giacometti, D. Laurent, and C. T. Diop. Condensed representations for sets of mining queries. In *Database Technologies for Data Mining - Discovering Knowledge with Inductive Queries*, volume 2682 of *LNCS*, pages 250–269. Springer-Verlag, 2004.
30. B. Goethals, J. Muhonen, and H. Toivonen. Mining non derivable association rules. In *Proc. SIAM Int. Conf. on Data Mining SDM'05*, Newport Beach, USA, Apr. 2005.

31. B. Goethals and M. J. Zaki, editors. *Proc. Int. Workshop on Frequent Itemset Mining Implementations FIMI'03*, Melbourne, Florida, USA, Nov. 2003.
32. J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proc. ACM Int. Conf. on Management of Data SIGMOD'00*, pages 1 – 12, Dallas, Texas, USA, May 2000. ACM Press.
33. T. Imielinski and H. Mannila. A database perspective on knowledge discovery. *Communications of the ACM*, 39(11):58–64, 1996.
34. B. Jeudy and J.-F. Boulicaut. Using condensed representations for interactive association rule mining. In *Proc. Principles and Practice of Knowledge Discovery in Databases PKDD'02*, volume 2431 of *LNAI*, pages 225–236, Helsinki, FIN, Aug. 2002. Springer-Verlag.
35. D. Kifer, J. Gehrke, C. Bucila, and W. M. White. How to quickly find a witness. In *Proc. ACM Symposium on Principles of Database Systems PODS'03*, pages 272–283, San Diego, USA, June 2003. ACM Press.
36. M. Kryszkiewicz. Concise representation of frequent patterns based on disjunction-free generators. In *Proc. IEEE Int. Conf. on Data Mining ICDM'01*, pages 305–312, San Jose, USA, Nov. 2001. IEEE Computer Press.
37. M. Kryszkiewicz and M. Gajek. Concise representation of frequent patterns based on generalized disjunction-free generators. In *Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining PAKDD'02*, volume 2336 of *LNCS*, pages 159–171, Taipei, Taiwan, 2002. Springer-Verlag.
38. B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rules mining. In *Proc. Int. Conf. on Knowledge Discovery and Data Mining KDD'98*, pages 80–86, New York, USA, 1998. AAAI Press.
39. H. Mannila and H. Toivonen. Multiple uses of frequent sets and condensed representations. In *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining KDD'96*, pages 189–194, Portland, USA, 1996. AAAI Press.
40. H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.
41. T. Mielikäinen. *Summarization Techniques for Pattern Collections in Data Mining.* PhD thesis, University of Helsinki, Department of Computer Science, Ph.D. thesis Report A-2005-1, 2005.
42. N. Novelli and R. Cicchetti. Mining functional and embedded dependencies using free sets. In *Actes Bases de Données Avancées BDA'00*, pages 201–220, 2000.
43. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Pruning closed itemset lattices for association rules. In *Actes Bases de Données Avancées BDA'98*, Hammamet, Tunisie, Oct. 1998.
44. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1):25–46, Jan. 1999.
45. J. Pei, J. Han, and R. Mao. CLOSET an efficient algorithm for mining frequent closed itemsets. In *Proc. SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery DMKD'00*, Dallas, USA, May 2000.
46. G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, pages 229–248. AAAI Press, 1991.
47. U. Rückert and S. Kramer. Generalized version space trees. In *Proc. Int. Workshop on Inductive Databases KDID'03*, pages 119–129, Cavtat-Dubrovnik, HR, 2003. Rudjer Boskovic Institute, Zagreb, HR.
48. A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. In *Proc. Int. Conf. on Very Large Data Bases VLDB'95*, pages 432 – 444, Zürich, CH, Sept. 1995. Morgan Kaufmann.

49. H. Toivonen. Sampling large databases for association rules. In *Proc. Int. Conf. on Very Large Data Bases VLDB'96*, pages 134–145, Mumbay, India, Sept. 1996. Morgan Kaufmann.
50. J. Wang and J. Han. BIDE: Efficient mining of frequent closed sequences. In *Proc. IEEE Int. Conf. on Data Engineering ICDE'04*, pages 79–90, Boston, USA, Apr. 2004. IEEE Computer Press.
51. R. Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In I. Rival, editor, *Ordered Sets*, pages 445–470. Reidel, Dordrecht-Boston, 1982.
52. A. Xu and H. Lei. LCGMiner: Levelwise closed graph pattern mining from large databases. In *Proc. Int. Conf. on Scientific and Statistical Database Management SSDBM'04*, pages 421–422, Santorini Island, EL, June 2004. IEEE Computer Press.
53. M. J. Zaki. Generating non-redundant association rules. In *Proc. ACM Int. Conf. on Knowledge Discovery and Data Mining SIGKDD'00*, pages 34–43, Boston, USA, Aug. 2000. ACM Press.
54. M. J. Zaki and C.-J. Hsiao. CHARM: An efficient algorithm for closed itemset mining. In *Proc. SIAM Int. Conf. on Data Mining SDM'02*, Arlington, USA, Apr. 2002.
55. M. J. Zaki and M. Ogihara. Theoretical foundations of association rules. In *Proc. SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery DMKD'98*, pages 1–8, June 1998.