

Computational Complexity of Itemset Frequency Satisfiability

Toon Calders ¹

toon.calders@ua.ac.be

Tel: +32 3 265 38 61 Fax: +32 3 265 37 77

University of Antwerp, Belgium

Middelheimlaan 1, BE-2020 Antwerpen

ABSTRACT

Computing frequent itemsets is one of the most prominent problems in data mining. We introduce a new, related problem, called **FREQSAT**: given some itemset-interval pairs, does there exist a database such that for every pair the frequency of the itemset falls in the interval? It is shown in this paper that **FREQSAT** is not finitely axiomatizable and that it is **NP**-complete. We also study cases in which other characteristics of the database are given as well. These characteristics can complicate **FREQSAT** even more. For example, when the maximal number of duplicates of a transaction is known, **FREQSAT** becomes **PP**-hard. We describe applications of **FREQSAT** in frequent itemset mining algorithms and privacy in data mining.

1. INTRODUCTION

The *frequent itemset mining problem* [1] is one of the core problems in data mining. We are given a database \mathcal{D} of sets, called *transactions*, and a threshold *minfreq*. The *frequency* of a set I in \mathcal{D} is the number of transactions in \mathcal{D} that contain all items of I divided by the total number of transactions in \mathcal{D} . The frequent itemset problem is to compute all sets I such that the frequency of I in \mathcal{D} is at least *minfreq*.

During the last decade, many algorithms to solve this problem were developed. For an overview, see [10]. All these frequent itemset mining algorithms rely heavily on the monotonicity of frequency: if $I \subseteq J$, then the frequency of J is bounded from above by the frequency of I . In general, this property of frequency allows for pruning substantial parts of the search space. Besides monotonicity, also other relationships between the frequencies can be identified. For example, in the MAXMINER algorithm [4], relations of the

¹Postdoctoral Fellow of the Fund for Scientific Research - Flanders (Belgium)(F.W.O. - Vlaanderen).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

PODS 2004 June 14-16, 2004, Paris, France.

Copyright 2004 ACM 1-58113-858-X/04/06 ... \$5.00.

following form are exploited:

$$\text{freq}(\{a, b, c\}) \geq \text{freq}(\{a, b\}) + \text{freq}(\{a, c\}) - \text{freq}(\{a\}) .$$

There are many more relations between the frequencies of itemsets. See [5] for extensions based on the inclusion-exclusion principle.

The relationships between the frequencies of itemsets can be seen as consistency constraints; only configurations of frequencies that satisfy these relationships, represent valid outcomes of frequent itemset mining. In this paper we are interested in the *complexity* of checking satisfiability of a given set of frequencies. In this context, we introduce the problem **FREQSAT**: given a collection of expressions $\text{freq}(I) \in [a, b]$, does there exist a transaction database that satisfies them? For example, $\{\text{freq}(\{a\}) \in [0, 0.5], \text{freq}(\{a, b\}) \in [0.6, 1]\}$ is not satisfiable, because of the monotonicity of frequency.

We prove that **FREQSAT** is not finitely axiomatizable. Hence, there are infinitely many non-redundant relations between frequencies. We show that **FREQSAT** is equivalent to *probabilistic satisfiability* [14], and hence **NP**-complete [9]. We also show that in **FREQSAT** we are able to express conditional frequencies. This ability allows us to express the confidence of association rules, and links **FREQSAT** to *probabilistic logic programming with conditional constraints* as studied by Lukasiewicz in [11].

We furthermore study cases in which, besides the frequency of itemsets, other characteristics of the database are given as well. Consider the following set \mathcal{C} of constraints:

$$\{\text{freq}(\{a\}) = \frac{1}{2}, \text{freq}(\{b\}) = \frac{1}{2}, \\ \text{freq}(\{c\}) = \frac{1}{2}, \text{freq}(\{a, b, c\}) = 0\}$$

\mathcal{C} is satisfiable by the database $\{a, b\}, \{a, c\}, \{b, c\}, \{\}$. If we, however, require that the number of transactions is 2, or that every transaction contains at most 1 item, \mathcal{C} is no longer satisfiable. This simple example already shows that a seemingly small adaptation of the original problem can have a large influence. Another important difference is in the entailment. $\text{ENT}_I(\mathcal{C})$ will denote the set of all possible frequency values for I given that \mathcal{C} holds. For **FREQSAT**, $\text{ENT}_I(\mathcal{C})$ is always an *interval* of the rational numbers. If we, however, fix the number of transactions, the set $\text{ENT}_I(\mathcal{C})$ can be *any finite subset* of rational numbers between 0 and 1.

The characteristics we consider are: the *maximal transaction size*, the *number of transactions*, and the *maximal number of duplicates* of a transaction. The complexity of the problem depends on the additional characteristics. We show that adding the transaction length does not affect the complexity of **FREQSAT**. Adding the number of transactions changes the properties of **FREQSAT** drastically, but it is left open whether it increases the complexity. Adding the maximal number of duplicates makes the problem provably more complex (assuming $\mathbf{NP} \neq \mathbf{PP}$).

Besides being of theoretical interest, **FREQSAT** and its variants have many practical applications too. These applications include: improving pruning in frequent set mining algorithms, constructing concise summaries of the frequent itemsets, and checking whether publishing the frequent itemsets provides a threat to the privacy of the original dataset.

The organization of the paper is as follows. In Section 2 we formally introduce important notions such as frequency constraints and entailment. Section 3 states the different variants of **FREQSAT** studied in the paper. In Section 4 **FREQSAT** without the extensions is studied. Then, in Sections 5, 6, and 7, **FREQSAT** is gradually extended with bounds on the transaction length, the number of transactions, and the number of duplicates. Section 8 gives some applications of **FREQSAT** and its variants, and Section 9 summarizes the most important results and concludes the paper.

2. PRELIMINARIES

2.1 Itemsets

Let \mathcal{I} be a finite set of items. A *transaction* over \mathcal{I} is a pair (tid, J) , with tid an identifier, and J a subset of \mathcal{I} . A *transaction database* over \mathcal{I} is a finite set of such transactions where every transaction has a unique identifier. Let I be some set of items. We say that the transaction (tid, J) *contains* I , denoted $I \subseteq (tid, J)$, if $I \subseteq J$. The *support* of I in \mathcal{D} , denoted $\text{supp}(I, \mathcal{D})$, is the absolute number of transactions in \mathcal{D} that contain I . The *frequency* of I in \mathcal{D} , denoted $\text{freq}(I, \mathcal{D})$, is $\text{supp}(I, \mathcal{D})$ divided by the number of transactions in \mathcal{D} . In all what follows, \mathcal{D} is a transaction database over \mathcal{I} .

2.2 Frequency Constraints

A *Frequency Constraint* is an expression $\text{freq}(I) \in [l, u]$, with I an itemset, and $0 \leq l, u \leq 1$ rational numbers. We say that \mathcal{D} *satisfies* this expression, denoted $\mathcal{D} \models \text{freq}(I) \in [l, u]$, if the frequency of I in \mathcal{D} is in the interval $[l, u]$. \mathcal{D} satisfies a set of frequency constraints, if it satisfies all of them. A set of frequency constraints \mathcal{C} *entails* a constraint $\text{freq}(I) \in [l, u]$, denoted $\mathcal{C} \models \text{freq}(I) \in [l, u]$, if every database \mathcal{D} that satisfies \mathcal{C} , satisfies $\text{freq}(I) \in [l, u]$ as well. The entailment is said to be *tight*, denoted $\mathcal{C} \models_{\text{tight}} \text{freq}(I) \in [l, u]$, if for every smaller interval $[l', u'] \subset [l, u]$, \mathcal{C} does not entail $\text{freq}(I) \in [l', u']$. That is, if $[l, u]$ is the best interval that can be derived for I , based on \mathcal{C} .

For notational convenience, we use the shorthand $\text{freq}(I) = f$ for $\text{freq}(I) \in [f, f]$.

2.3 Database Characteristics

In realistic situations, often, more characteristics of a transaction database are known than only the frequencies of some sets. We now describe what extra information we will consider in this paper.

Number of Transactions The size of the database $|\mathcal{D}|$ is often known to the user. Knowing the number of transactions seriously affects the properties of **FREQSAT**.

Transaction Length The number of items is of course always an upper bound for the maximal number of items in a transaction. Often, however, the maximal size of the transactions is given. Moreover, it is a common practice in frequent itemset mining to start from a relational table $R(A_1, \dots, A_n)$, and to encode this table as a transaction database before mining. This transformation is carried out as follows: for every attribute-value pair A, v of R , an item $I_{(A,v)}$ is introduced. A tuple (v_1, \dots, v_n) is represented by the transaction $\{I_{(A_1,v_1)}, \dots, I_{(A_n,v_n)}\}$. Hence, if the original schema is known, also the maximal transaction size is known.

Number of Duplicates In our definition of frequent set mining we did not require that the set of items in a transaction is unique; due to the identifier, two different transactions can have the same set of items. In many practical situations, however, duplicates cannot occur, or a maximal number of duplicates is known. For example, in case the transaction database was created from a relational table, no duplicate transactions can be present. Even if some attributes of the original table are filtered away, the maximal possible number of duplicates might be known. Suppose that the table $R(A_1, \dots, A_n)$ is transformed as described above, but some, binary valued attributes A_1, \dots, A_k are filtered away. In that case, the number of duplicates is at most 2^k .

2.4 Complexity Classes

We give a brief overview of the complexity classes used throughout this paper. For a more comprehensive overview, we refer to [15].

DP is the complexity class that contains all languages L such that there exist languages $L_1 \in \mathbf{NP}$ and $L_2 \in \mathbf{co-NP}$ with $L = L_1 \cap L_2$. **NP** is included in **DP**, and unless $\mathbf{NP} = \mathbf{co-NP}$, this inclusion is strict. **DP** on its turn is included in $\mathbf{P}^{\mathbf{NP}}$, the class of languages decidable in polynomial time with an **NP**-oracle. The prototypical **DP**-complete problem is **SAT-UNSAT**. **SAT-UNSAT** contains all pairs of Boolean formulas ϕ, ψ , such that ϕ is satisfiable, and ψ unsatisfiable.

We say that a language L is in **PP** if there exists a non-deterministic polynomially bounded Turing machine N such that, for all inputs x , $x \in L$ if and only if *more than half of the computations of N on input x end up accepting*. We say that N decides L “by majority”. It is known that **NP** is included in **PP**. It is also widely believed that this inclusion is strict, for a number of reasons. First, **PP** is closed under complement, whereas **NP** is believed to be not. Second, *Toda’s theorem* states that the polynomial hierarchy **PH** is a subset of $\mathbf{P}^{\mathbf{PP}}$. Hence, $\mathbf{PP} = \mathbf{NP}$ would cause the

polynomial hierarchy **PH** to collapse to **P^{NP}**. **P^{PP}** is included in **PSPACE**. The **MAJSAT**-problem, asking if more than half of the truth assignments for a given formula ϕ are accepting, is **PP**-complete.

Let Q be a polynomially balanced, polynomial-time decidable binary relation. The *counting problem* associated with Q is the following: Given x , how many y are there such that $(x, y) \in Q$? The output required is an integer in binary. **#P** is the class of all counting problems associated with polynomially balanced, polynomial-time decidable relations. **#SAT**, the problem asking for the number of accepting truth assignments of a given Boolean formula, is **#P**-complete. The relation between **PP** and **#P** is best summarized by the following equality, due to *Angluin*: **P^{#P}** = **P^{PP}**. Hence, their complexity is comparable.

Unless explicitly mentioned otherwise, we use logspace reductions throughout the paper. $L_1 \leq L_2$ denotes that L_1 is logspace reducible to L_2 , and $L_1 \equiv L_2$ denotes $L_1 \leq L_2$ and $L_2 \leq L_1$.

3. PROBLEM STATEMENT

We are now ready to state the main problems in this paper: the **FREQSAT**-problem and its variants.

Problem FREQSAT:

Input: A set of frequency constraints

$$\mathcal{C} = \{\text{freq}(I_j) \in [l_j, u_j], j = 1 \dots m\}$$

Accept: if and only if there exists a database \mathcal{D} over $\bigcup_{j=1}^m I_j$ that satisfies \mathcal{C} . \square

As we discussed before, we also study cases wherein more characteristics of the database \mathcal{D} are known. Consider the following characteristics of a transaction database:

$$\begin{aligned} \text{ltrans}(\mathcal{D}) &=_{\text{def}} \max\{|J| \mid (tid, J) \in \mathcal{D}\} \\ \text{ntrans}(\mathcal{D}) &=_{\text{def}} |\mathcal{D}| \\ \text{ndup}(\mathcal{D}) &=_{\text{def}} \max_{J \subseteq \mathcal{I}} |\{tid \mid (tid, J) \in \mathcal{D}\}| \end{aligned}$$

FREQSAT $\{c_1, \dots, c_k\}$ is the variant of the **FREQSAT**-problem where upper bounds on the characteristics c_1, \dots, c_k are part of the input as well. Hence, **FREQSAT** $\{\text{ltrans}, \text{ndup}\}$ denotes the variant in which besides frequency constraints, a maximal transaction length and a maximal number of duplicates have been given as well.

Problem FREQSAT $\{c_1, \dots, c_k\}$:

Input: A tuple $(\mathcal{C}, v_1, \dots, v_k)$, with \mathcal{C} a set of frequency constraints

$$\{\text{freq}(I_j) \in [l_j, u_j], j = 1 \dots m\},$$

and v_1, \dots, v_k numbers.

Accept: if and only if there exists a database \mathcal{D} over $\bigcup_{j=1}^m I_j$ that satisfies \mathcal{C} and for all $l = 1 \dots k$, $c_l(\mathcal{D}) \leq v_l$. \square

The main question in this paper is: what are the computational complexities of the different **FREQSAT**-variants, and what are the relations and differences between them?

4. FREQSAT

In this section we study the base case where no additional characteristics of the database are given. We show that **FREQSAT** is **NP**-complete, by proving that it is equivalent to *probabilistic satisfiability* (**pSAT**) [14]. This equivalence also gives important properties of the **FREQSAT**-problem, such as that the possible frequencies of an itemset I given a set of constraints is always an interval $[l, u]$.

We also show that confidence of association rules can be expressed with **FREQSAT**. Because of this, we can use complexity results of Lukasiewicz in the context of probabilistic logic programming with conditional constraints [11]. These results concern the complexities of deciding whether \mathcal{C} entails $\text{freq}(I) \in [l, u]$ (tight), and the function problem asking for the tight interval for I given \mathcal{C} .

4.1 Probabilistic Satisfiability

The probabilistic satisfiability problem (**pSAT**) [14] is defined as follows: Consider m logical sentences over the variables x_1, \dots, x_n with the usual Boolean operators \neg, \vee, \wedge . Assume probabilities π_1, \dots, π_m for these sentences to be true are given. Does there exist a probability distribution over the truth assignments over x_1, \dots, x_n that make these probabilities true; i.e., are the probabilities consistent? In [9], it is proven that **pSAT** is **NP**-complete. The extension from exact probabilities π_1, \dots, π_m to intervals does not change the computational properties of **pSAT**.

We can relate **FREQSAT** to **pSAT** as follows: Consider the items as variables, and the transactions as truth assignments. A transaction (tid, J) corresponds to the assignment that makes all items $i \in J$ true, and the others false. A transaction database then represents a probability distribution over the different truth assignments. The probability of a certain assignment is the fraction of transactions representing this valuation. The frequency of an itemset $\{i_1, \dots, i_k\}$ in the database \mathcal{D} , is now equal to the probability of the conjunction $\bigwedge_{j=1}^k i_j$. Hence, we can easily reduce **FREQSAT** to **pSAT**. Therefore, **FREQSAT** is in **NP**. The following lemma says that given a set of frequency constraints \mathcal{C} , the set of possible frequencies for I is always an interval that can be described succinctly. It also gives an upper bound on the number of transactions of a minimal satisfying database for \mathcal{C} .

Lemma 1. For every set of frequency constraints \mathcal{C} , and itemset I , the set

$$\text{ENT}_I(\mathcal{C}) =_{\text{def}} \{\text{freq}(I, \mathcal{D}) \mid \mathcal{D} \models \mathcal{C}\}$$

is always an interval $[l, u]$ of the rational numbers, with l and u having polynomial length in the length of the description of \mathcal{C} .

Furthermore, every satisfiable \mathcal{C} has a satisfying database \mathcal{D} with at most 2^c transactions, with c polynomial in the length of the description of \mathcal{C} .

On the other hand, we can extend **FREQSAT** to include frequency constraints over arbitrary Boolean formulas. This shows in fact that we can also reduce **pSAT** to **FREQSAT**.

Reduction from pSAT to FREQSAT: An extended frequency

constraint is an expression $\text{freq}(\varphi) \in [l, u]$, with φ a Boolean formula over the set of items \mathcal{I} . We say that a transaction (tid, J) *satisfies* φ , if the truth assignment V that assigns 1 to an item i if and only if $i \in J$, makes φ true. Frequency of a Boolean formula, and the satisfaction and entailment of an extended frequency constraint are now defined in the same way as for itemsets and regular frequency constraints. It is easy to see that the extension of FREQSAT to arbitrary Boolean formulas gives **psAT**. We thus show that in FREQSAT we can simulate extended frequency constraints.

Let $\mathcal{P} = \{\text{freq}(\varphi_1) \in [l_1, u_1], \dots, \text{freq}(\varphi_m) \in [l_m, u_m]\}$ be a set of m *extended* frequency constraints with $\varphi_1, \dots, \varphi_m$ Boolean formulas over the set of items $\{i_1, \dots, i_n\}$. For every subexpression σ of the formulas $\varphi_1, \dots, \varphi_m$ (also for the items), we introduce two new items, t_σ and f_σ . t_σ stands for σ is true, and f_σ for σ is false. A transaction $T = (tid, J)$ will represent the truth assignment V_T that assigns true to all items such that t_i is in J , and false to the others. We will now add constraints such that t_σ is in a transaction T if and only if the truth assignment V_T makes σ true.

The main crux in the reasoning is that only half of the transactions will represent valid truth assignments. These transactions will contain the item d , the others contain item \bar{d} (hence, \bar{d} is in fact *not* d):

$$\begin{aligned} \text{freq}(\{d\}) &= 0.5, & \text{freq}(\{\bar{d}\}) &= 0.5, \\ \text{freq}(\{d, \bar{d}\}) &= 0. \end{aligned}$$

For every subexpression σ , we add the following constraints:

$$\begin{aligned} \text{freq}(\{t_\sigma\}) &= 0.5, & \text{freq}(\{f_\sigma\}) &= 0.5, \\ \text{freq}(\{t_\sigma, f_\sigma\}) &= 0. \end{aligned}$$

In this way, we make sure that every transaction contains either t_σ , or f_σ , but not both. We use the transactions containing \bar{d} to compensate the fact that we do not know how many trues and falses we need for σ . For example, for $a \vee \neg a$, half of the transactions will contain $\{d, t_{a \vee \neg a}\}$, and the other half contains $\{\bar{d}, f_{a \vee \neg a}\}$. Hence, even though only *half* of the transactions contain $t_{a \vee \neg a}$, *all* transactions representing *valid* truth assignments contain $t_{a \vee \neg a}$.

We still have to make sure that within the d -part of a satisfying database, the trues and falses are consistent with each other. For example, a transaction representing a truth assignment cannot contain $t_{a \vee b}$, f_a , and f_b at the same time. The consistency is enforced as follows: for every subexpression σ that is not an atom, depending on its form, we introduce the following constraints:

$$\begin{aligned} \sigma \equiv \sigma_1 \vee \sigma_2 : & & \text{freq}(\{d, t_\sigma, f_{\sigma_1}, f_{\sigma_2}\}) &= 0, \\ & & \text{freq}(\{d, f_\sigma, t_{\sigma_1}\}) &= 0, \\ & & \text{freq}(\{d, f_\sigma, t_{\sigma_2}\}) &= 0. \\ \sigma \equiv \sigma_1 \wedge \sigma_2 : & & \text{freq}(\{d, f_\sigma, t_{\sigma_1}, t_{\sigma_2}\}) &= 0, \\ & & \text{freq}(\{d, t_\sigma, f_{\sigma_1}\}) &= 0, \\ & & \text{freq}(\{d, t_\sigma, f_{\sigma_2}\}) &= 0. \\ \sigma \equiv \neg \sigma_1 : & & \text{freq}(\{d, t_\sigma, f_{\sigma_1}\}) &= 0, \\ & & \text{freq}(\{d, f_\sigma, t_{\sigma_1}\}) &= 0. \end{aligned}$$

Hence, for every subexpression σ , every transaction contains either t_σ or f_σ , but not both. Every transaction T that contains d , contains t_σ if and only if $V_T(\sigma)$ is true.

TID	Items
1	$d, t_a, t_b, t_c, f_{\neg a}, t_{(\neg a) \vee b}, t_{b \wedge c}$
2	$d, f_a, f_b, t_c, t_{\neg a}, t_{(\neg a) \vee b}, f_{b \wedge c}$
3	$d, f_a, f_b, t_c, t_{\neg a}, t_{(\neg a) \vee b}, f_{b \wedge c}$
4	$d, t_a, f_b, f_c, f_{\neg a}, f_{(\neg a) \vee b}, f_{b \wedge c}$
5	$d, t_a, f_b, f_c, f_{\neg a}, f_{(\neg a) \vee b}, f_{b \wedge c}$
6	$\bar{d}, f_a, f_b, f_c, t_{\neg a}, f_{(\neg a) \vee b}, f_{b \wedge c}$
7	$\bar{d}, t_a, t_b, f_c, f_{\neg a}, f_{(\neg a) \vee b}, t_{b \wedge c}$
8	$\bar{d}, t_a, t_b, f_c, f_{\neg a}, f_{(\neg a) \vee b}, t_{b \wedge c}$
9	$\bar{d}, f_a, t_b, t_c, t_{\neg a}, t_{(\neg a) \vee b}, t_{b \wedge c}$
10	$\bar{d}, f_a, t_b, t_c, t_{\neg a}, t_{(\neg a) \vee b}, t_{b \wedge c}$

$$\mathcal{P} = \left\{ \begin{array}{l} \text{freq}(a) \in [0.4, 0.7], \quad \text{freq}((\neg a) \vee b) = 0.6, \\ \text{freq}(b \wedge c) \in [0.2, 0.4], \quad \text{freq}(c) = 0.6 \end{array} \right\}.$$

Figure 1: A database satisfying \mathcal{P} and a corresponding database for $\mathcal{F}(\mathcal{P})$.

Furthermore, for all $j = 1 \dots m$, we introduce the constraint

$$\{\text{freq}(\{d, t_{\varphi_j}\}) \in [l/2, u/2]\}.$$

That is, we only measure the frequency of the formulas φ within the fraction of the database with d , that is, the valid truth assignments. Since exactly half of the transactions contain d , the bounds of the intervals have to be divided by 2.

We denote the resulting FREQSAT-instance by $\mathcal{F}(\mathcal{P})$. It is now true that $\mathcal{F}(\mathcal{P})$ is satisfiable if and only if \mathcal{P} is. Henceforth, we can reduce **psAT** to FREQSAT. In the rest of the paper we will often identify the itemset I with the conjunction $\bigwedge_{i \in I} i$.

Lemma 2. \mathcal{P} is satisfiable if and only if $\mathcal{F}(\mathcal{P})$ is satisfiable. Furthermore, $\text{ENT}_{\varphi}(\mathcal{P}) = [l, u]$ if and only if $\text{ENT}_{\{d, t_{\varphi}\}}(\mathcal{F}(\mathcal{P})) = [l/2, u/2]$.

Example 1. Consider the following set of extended frequency constraints \mathcal{P} :

$$\mathcal{P} = \left\{ \begin{array}{l} \text{freq}(a) \in [0.4, 0.7], \quad \text{freq}((\neg a) \vee b) = 0.6, \\ \text{freq}(b \wedge c) \in [0.2, 0.4], \quad \text{freq}(c) = 0.6 \end{array} \right\}.$$

$\mathcal{F}(\mathcal{P})$ is a set of frequency constraints over the items

$$\{t_a, f_a, t_b, f_b, t_c, f_c, t_{\neg a}, f_{\neg a}, t_{(\neg a) \vee b}, f_{(\neg a) \vee b}, t_{b \wedge c}, f_{b \wedge c}, d, \bar{d}\}.$$

The first type of constraints in $\mathcal{F}(\mathcal{P})$ make sure that t_σ and f_σ are complements of each other:

$$\begin{aligned} \text{freq}(\{t_a, f_a\}) &= 0, & \text{freq}(\{t_a\}) &= 0.5, & \text{freq}(\{f_a\}) &= 0.5 \\ \text{freq}(\{t_b, f_b\}) &= 0, & \text{freq}(\{t_b\}) &= 0.5, & \text{freq}(\{f_b\}) &= 0.5 \end{aligned}$$

$$\dots$$

$$\text{freq}(\{t_{b \wedge c}, f_{b \wedge c}\}) = 0, \text{freq}(\{t_{b \wedge c}\}) = 0.5, \text{freq}(\{f_{b \wedge c}\}) = 0.5$$

The item d is in half of the transactions, and \bar{d} is its complement:

$$\text{freq}(\{d, \bar{d}\}) = 0, \text{freq}(\{d\}) = 0.5, \text{freq}(\{\bar{d}\}) = 0.5.$$

The second type of constraints makes sure that within the transactions that contain d of a satisfying database, the trues and falses are consistent:

$$\begin{aligned} \text{freq}(\{d, t_a, t_{\neg a}\}) &= 0, & \text{freq}(\{d, f_a, f_{\neg a}\}) &= 0 \\ \text{freq}(\{d, t_{\neg a}, f_{(\neg a) \vee b}\}) &= 0, & \text{freq}(\{d, t_b, f_{(\neg a) \vee b}\}) &= 0, \\ \text{freq}(\{d, f_{\neg a}, f_b, t_{(\neg a) \vee b}\}) &= 0 \\ \text{freq}(\{d, f_b, t_{b \wedge c}\}) &= 0, & \text{freq}(\{d, f_c, t_{b \wedge c}\}) &= 0, \\ \text{freq}(\{d, t_a, t_b, f_{b \wedge c}\}) &= 0 \end{aligned}$$

Finally, the third type of constraints translates the extended frequency constraints:

$$\begin{aligned} \text{freq}(\{d, t_a\}) &\in [0.2, 0.35], & \text{freq}(\{d, t_{(\neg a) \vee b}\}) &= 0.3, \\ \text{freq}(d, t_{b \wedge c}) &\in [0.1, 0.2], & \text{freq}(\{d, t_c\}) &= 0.3 \end{aligned}$$

In Figure 1, two databases satisfying respectively \mathcal{P} and $\mathcal{F}(\mathcal{P})$ have been given.

The following theorem follows immediately from Lemma 2.

Theorem 1. $\text{FREQSAT} \equiv \text{pSAT}$ and hence, FREQSAT is **NP-complete**.

4.2 Association Rules

We can also simulate association rules in FREQSAT . An *association constraint* is an expression $\text{conf}(I \rightarrow J) \in [l, u]$, with I, J itemsets. A database \mathcal{D} satisfies this association constraint if and only if

$$l \cdot \text{freq}(I, \mathcal{D}) \leq \text{freq}(I \cup J, \mathcal{D}) \leq u \cdot \text{freq}(I, \mathcal{D}) .$$

4.2.1 Multiplication Lemma

We first introduce the following multiplication-lemma. This lemma shows how we can construct a new item m that has exactly d times the frequency of a given itemset I .

Lemma 3. Let \mathcal{C} be a set of frequency constraints, I an itemset, m an item not used in \mathcal{C} , and d a positive integer.

There exists a set of constraints $\mathcal{M}_d(I, m)$, polynomial in the length of d , such that:

(1) $\mathcal{C} \cup \{\text{freq}(I) \in [0, 1/d]\}$ is satisfiable if and only if $\mathcal{C} \cup \mathcal{M}_d(I, m)$ is, and

(2) If \mathcal{D} satisfies $\mathcal{C} \cup \mathcal{M}_d(I, m)$, then $\text{freq}(\{m\}, \mathcal{D}) = d \cdot \text{freq}(I, \mathcal{D})$.

Sketch We will give extended frequency constraints with Boolean formulas that describe $\mathcal{M}_d(I, J)$. Later on, using Lemma 2, these extended frequency constraints can be translated into regular frequency constraints.

Introduce a new item s_I^m . Let $\mathcal{M}_1(I, m) = \{\text{freq}(s_I^m \vee I) = 1/d, \text{freq}(s_I^m \vee m) = 1/d, \text{freq}(s_I^m \wedge I) = 0, \text{freq}(s_I^m \wedge m) = 0\}$. Hence, s_I^m and I never occur together, and the sum of their frequencies is $1/d$. Thus, $\text{freq}(I) = 1/d - \text{freq}(s_I^m)$. Similarly, $\text{freq}(m) = 1/d - \text{freq}(s_I^m)$, and hence, $\text{freq}(I) = \text{freq}(m)$.

We now express \mathcal{M}_2 by applying \mathcal{M}_1 twice, and setting the result item m equal to the union of the results of the two \mathcal{M}_1 . We can express that m is in exactly those transactions that contain either m_1 or m_2 as follows:

$$\text{freq}((m \wedge \neg(m_1 \vee m_2)) \vee ((m_1 \vee m_2) \wedge \neg m)) = 0$$

We denote this expression $E(m, m_1 \vee m_2)$ (E of “equal”). Let $\mathcal{M}_2(I, m)$ be the following expression:

$$\begin{aligned} &\mathcal{M}_1(I, m_1) \cup \mathcal{M}_1(I, m_2) \\ &\cup \{\text{freq}(m_1 \wedge m_2) = 0, E(m, m_1 \vee m_2)\} . \end{aligned}$$

Hence, we can multiply frequencies by 2. By iteratively multiplying by two, we can multiply by 2^k . Using the addition technique with $E(\cdot, \cdot)$, we can add frequencies. Therefore, we can multiply with an arbitrary number d . \square

Example 2. Let I be an itemset. The following expression $\mathcal{M}_1(\{a\}, x)$ forces $\text{freq}(\{x\})$ to be equal to $\text{freq}(\{a\})$, assuming that $\text{freq}(\{a\})$ is at most $1/d$:

$$\begin{aligned} \mathcal{M}_1(I, x) &= \{\text{freq}(s_I^x \vee I) = 1/d, \text{freq}(s_I^x \vee x) = 1/d, \\ &\text{freq}(s_I^x \wedge I) = 0, \text{freq}(s_I^x \wedge x) = 0\} \end{aligned}$$

In the rest of the construction we will often use \mathcal{M}_1 as a copying mechanism. The parameter d , however, can be different for the different expressions. Therefore we denote the value of d as a superscript to the notation; that is, we will use $\mathcal{M}_1^d(I, x)$. $\mathcal{M}_2, \mathcal{M}_4$, and \mathcal{M}_5 are now constructed as follows:

$$\begin{aligned} \mathcal{M}_2(I, x_2) &= \mathcal{M}_1^{1/5}(I, x_1) \cup \mathcal{M}_1^{1/5}(I, x_1^2), \\ &\cup \{\text{freq}(x_1 \wedge x_2) = 0, E(x_2, x_1 \vee x_1^2)\} \\ \mathcal{M}_4(I, x_4) &= \mathcal{M}_2(I, x_2) \cup \mathcal{M}_1^{2/5}(\{x_2\}, x_2) \cup \mathcal{M}_1^{2/5}(\{x_2\}, x_2^2) \\ &\cup \{\text{freq}(x_2 \wedge x_2^2) = 0, E(x_4, x_2 \vee x_2^2)\} \\ \mathcal{M}_5(I, x_5) &= \mathcal{M}_1^{1/5}(I, x_1) \cup \mathcal{M}_4(I, x_4) \\ &\cup \{\text{freq}(x_1 \wedge x_4) = 0, E(x_5, x_1 \vee x_4)\} \end{aligned}$$

The following database satisfies

$$\{\text{freq}(\{a\}) = 0.5, \text{freq}(\{b\}) = 0.5\} \cup \mathcal{M}_5(\{a, b\}, m) .$$

TID	Items					
1	a, b	x_1	x_1^1, x_2			m
2	a	$s_{\{a,b\}}^{x_1}$	$s_{\{a,b\}}^{x_1^1}, s_{\{a,b\}}^{x_1^2}$	$s_{\{x_2\}}^{x_2^1}, s_{\{x_2\}}^{x_2^2}$		
3	a		x_2^2, x_2	x_2^1, x_4		m
4	a			$s_{\{x_2\}}^{x_2^1}, s_{\{x_2\}}^{x_2^2}$		
5	a			x_2^1, x_4		m
6	b			x_2^2, x_4		m
7	b			x_2^2, x_4		m
8	b					
9	b					
10						

4.2.2 Reduction

Assume that besides frequency constraints \mathcal{C} , also a set of association constraints \mathcal{A} has been given. We show that there exists a FREQSAT -instance $\mathcal{F}(\mathcal{C} \cup \mathcal{A})$ that is equivalent to $\mathcal{C} \cup \mathcal{A}$.

Reduction from $\mathcal{C} \cup \mathcal{A}$ to $\mathcal{F}(\mathcal{C} \cup \mathcal{A})$ An association constraint $\text{conf}(I \rightarrow J) \in [l, u]$ holds if and only if

$$l \cdot \text{freq}(I, \mathcal{D}) \leq \text{freq}(I \cup J, \mathcal{D}) \leq u \cdot \text{freq}(I, \mathcal{D}) .$$

We concentrate on $l \cdot \text{freq}(I, \mathcal{D}) \leq \text{freq}(I \cup J, \mathcal{D})$; the simulation of the upper bound u is similar. Let $l = \frac{\alpha}{\beta}$. Then, the lower bound holds if $\beta \cdot \text{freq}(I \cup J) \geq \alpha \cdot \text{freq}(I)$. We will construct $\mathcal{F}(\mathcal{C} \cup \mathcal{A})$ in such a way that when it is satisfiable by a database \mathcal{D} , only a small part of it is the encoding of a database that satisfies $\mathcal{C} \cup \mathcal{A}$. We mark the transactions that form the database for $\mathcal{C} \cup \mathcal{A}$, by adding a new item d to it. This gives the following constraints (we will specify the exact value of N later): $\text{freq}(\{d\}) = 1/N$, and for all $\text{freq}(I_j) \in [l_j, u_j]$, the constraint $\text{freq}(I_j \cup \{d\}) \in [l_j/N, u_j/N]$. The transactions that do not contain d can be considered as extra “workspace.”

We are now ready to encode $\beta \cdot \text{freq}(I \cup J) \geq \alpha \cdot \text{freq}(I)$. We have two items $m_{I \cup J}^\beta$ and m_I^α . Using the multiplication Lemma 3, we set the frequency of $m_{I \cup J}^\beta$ to β times the frequency of $I \cup J$: $\mathcal{M}_\beta(I \cup J, m_{I \cup J}^\beta)$. Similarly, we set the frequency of m_I^α to α times the frequency of I . We can now express that $\beta \cdot \text{freq}(I \cup J) \geq \alpha \cdot \text{freq}(I)$ by requiring that every m_I^α occurs together with $m_{I \cup J}^\beta$: $\text{freq}(\neg m_{I \cup J}^\beta \wedge m_I^\alpha) = 0$. In this way we can translate the association constraints one by one.

We have to choose N large enough, to make sure that there is enough workspace to do the multiplications. Hence, N should be at least α , where α is the greatest denominator of a bound of an association constraint. We denote this number N_A . This gives us the set of extended frequency constraints $\mathcal{P}(\mathcal{C} \cup \mathcal{A})$. Since we used arbitrary Boolean formulas in the translation, we still need to apply the translation from extended frequency constraints to regular frequency constraints. We call the resulting set of constraints $\mathcal{F}(\mathcal{C} \cup \mathcal{A})$.

Theorem 2. $\mathcal{C} \cup \mathcal{A}$ is satisfiable if and only if $\mathcal{F}(\mathcal{C} \cup \mathcal{A})$ is satisfiable. Furthermore, $\text{ENT}_I(\mathcal{C} \cup \mathcal{A}) = [l, u]$ if and only if $\text{ENT}_I(\mathcal{F}(\mathcal{C} \cup \mathcal{A})) = [l/(2 \cdot N_A), u/(2 \cdot N_A)]$.

Example 3. Let $\mathcal{C} \cup \mathcal{A}$ be the following set:

$$\{\text{freq}(a) = 3/4, \text{freq}(\{a, b\}) = 1/2\} \cup \{\text{conf}(a \rightarrow b) \in [1/2, 1]\} .$$

N_A equals 2. The following databases satisfy respectively $\mathcal{C} \cup \mathcal{A}$ and $\mathcal{P}(\mathcal{C} \cup \mathcal{A})$:

TID	Items
1	a
2	a, b
3	a, b
4	b

 \rightarrow

TID	Items
1	$d, a \quad m_a^1, \quad m_a^2, \quad m_{\{a,b\}}^2$
2	$d, a, b \quad m_a^1, \quad m_a^2, \quad m_{\{a,b\}}^2$
3	$d, a, b \quad m_a^1, \quad m_a^2, \quad m_{\{a,b\}}^2$
4	$d, b \quad m_a^2, \quad m_{\{a,b\}}^2$
5	m_a^2
6	m_a^2
7	
8	

Entailment problems. Consider the following three entailment problems associated with FREQSAT:

- (1) FREQUENT(\mathcal{C} , $\text{freq}(I) \in [l, u]$)
Decide whether $\mathcal{C} \models \text{freq}(I) \in [l, u]$.
- (2) T-FREQUENT(\mathcal{C} , $\text{freq}(I) \in [l, u]$)
Decide whether $\mathcal{C} \models_{\text{tight}} \text{freq}(I) \in [l, u]$.
- (3) Func T-FREQUENT(\mathcal{C} , I)
Give $[l, u]$ such that $\mathcal{C} \models_{\text{tight}} \text{freq}(I) \in [l, u]$.

The complexity of these three problems is related to the complexity of FREQSAT. Since we can use association rules, the entailment problems are equivalent with the entailment problems in the context of conditional events studied by Lukasiewicz in [11]. Hence, we obtain the following theorem:

Theorem 3.

- FREQUENT is **co** – **NP**-complete.
- T-FREQUENT is **DP**-complete.
- Func T-FREQUENT is **FP**^{NP}-complete.

4.3 Axiomatization

We show that FREQSAT is not finitely axiomatizable.

We first give a lemma and theorem that provide a set of axioms that are sound and complete in the special case of sets of frequency constraints that contain an expression $\text{freq}(I) = f_I$ for all sets $I \subseteq \mathcal{I}$. The number of axioms depends on the set \mathcal{I} , and the axioms are only complete in this very special case. In Theorem 4, we then show that in general, no finite axiomatization of FREQSAT exists.

Lemma 4. Let f and g be two functions defined on the subsets of a finite set \mathcal{I} . The following two expressions are equivalent:

- (1) For all $I \subseteq \mathcal{I}$, $f(I) = \sum_{I \subseteq K} g(K)$;
- (2) For all $J \subseteq \mathcal{I}$, $g(I) = \sum_{I \subseteq K} (-1)^{|K-I|} f(K)$.

Theorem 4. Let for all $I \subseteq \mathcal{I}$, f_I be a rational number. There exists a transaction database \mathcal{D} such that for all $I \subseteq \mathcal{I}$, $\text{freq}(I, \mathcal{D}) = f_I$ if and only if, for all $I \subseteq \mathcal{I}$, the following rule holds:

$$\mathcal{R}_{\mathcal{I}}(I) \quad \sigma_{\mathcal{I}}(I) = \sum_{I \subseteq K \subseteq \mathcal{I}} (-1)^{|K-I|} f_K \geq 0$$

PROOF. If Let d be the least common multiplier of the denominators of the sums $\sigma_{\mathcal{I}}(I)$. The database \mathcal{D} consists of the following transactions: for every set $I \subseteq \mathcal{I}$, there are $d \cdot \sigma_{\mathcal{I}}(I)$ transactions (tid, I) . Notice that all $d \cdot \sigma_{\mathcal{I}}(I)$ are positive integers. Thus, for all $I \subseteq \mathcal{I}$, $\text{freq}(I, \mathcal{D}) = \sum_{I \subseteq J} \sigma_{\mathcal{I}}(J)$. Using Lemma 4, we hence get $\text{freq}(I, \mathcal{D}) = f_I$. Only If Let \mathcal{D} be a database that fulfills the given frequencies. Let ϕ_I denote the fraction of transactions (tid, J) , with $J = I$. Hence, $\text{freq}(I, \mathcal{D}) = \sum_{I \subseteq K} \phi_K$. Via Lemma 4, we get that $\sigma_{\mathcal{I}}(I)$ equals ϕ_I . Hence, since all ϕ_I must be positive, all $\sigma_{\mathcal{I}}(I)$ must be as well. \square

Theorem 5. *Every axiomatization for FREQSAT that does not include an axiom that involves the frequency of all non-empty itemsets is incomplete. Therefore, FREQSAT is not finitely axiomatizable.*

Sketch Let n be an arbitrary number. We can construct a FREQSAT problem \mathcal{C} over the set $\mathcal{I} = \{i_1, \dots, i_n\}$, such that (a) \mathcal{C} is not satisfiable, but, (b) every strict subset of \mathcal{C} is satisfiable. Furthermore, \mathcal{C} contains one expression $\text{freq}(I) = f_I$ for every $I \subseteq \mathcal{I}$.

From (a) and (b) it follows then that an axiomatization for FREQSAT must contain at least one axiom that involves every frequency constraint in the input. Indeed; suppose that the axioms A_1, A_2, \dots, A_m are sound and complete for FREQSAT, but none of the axioms A_i involves all frequencies. Because \mathcal{C} is not satisfiable, there must be at least one axiom A that is not satisfied by \mathcal{C} . This is so because \mathcal{C} contains an expression $\text{freq}(I) = f_I$, for every subset I of \mathcal{I} . Hence, every expression $\text{freq}(I) \in [l, u]$ entailed by \mathcal{C} is either in contradiction with $\text{freq}(I) = f_i$, or is less expressive. Therefore, if it can be derived by the axioms that \mathcal{C} is not satisfiable, then this can be derived in one step. Suppose that this unsatisfied axiom A does not involve itemset I , and c is the constraint in \mathcal{C} involving I . Then we have the following contradiction: $\mathcal{C} \setminus \{c\}$ is satisfiable, but violates A .

We assume that n is even. (a similar system can be found for n odd) Let \mathcal{C} be

$$\left\{ \text{freq}(I) = \frac{2^{(n-|I|)}}{(2^n)-1} \mid \emptyset \subset I \subset \mathcal{I} \right\} \cup \{ \text{freq}(\mathcal{I}) = 0 \}$$

This set \mathcal{C} fulfils the conditions (a) and (b). For all $I \neq \emptyset$, we have:

$$\begin{aligned} \sigma_{\mathcal{I}}(I) &= \sum_{I \subseteq K \subset \mathcal{I}} (-1)^{|K-I|} \frac{2^{(n-|K|)}}{(2^n)-1} \\ &= \frac{1}{(2^n)-1} (1 - (-1)^{n-|I|}) \end{aligned}$$

Hence, for all $I \neq \emptyset$, $\sigma_{\mathcal{I}}I$ equals 0 if $|I|$ is even, and 2 if $|I|$ is odd. For $I = \emptyset$, we get:

$$\sigma_{\mathcal{I}}(\emptyset) = \sum_{K \subset \mathcal{I}} (-1)^{|K|} \frac{2^{(n-|K|)}}{(2^n)-1} = -\frac{1}{(2^n)-1}$$

Thus, \mathcal{C} is not satisfiable. However, for every nonempty set I , if we remove the expression with I from \mathcal{C} , the resulting system \mathcal{C}' is satisfiable. Let I be odd: $\mathcal{C}' \cup \{ \text{freq}(I) = \frac{2^{(n-|I|)}-1}{(2^n)} \}$ is satisfiable, if $I \neq \mathcal{I}$ is even, $\mathcal{C}' \cup \{ \text{freq}(I) = \frac{2^{(n-|I|)}+1}{(2^n)} \}$ is satisfiable, and for $I = \mathcal{I}$, $\mathcal{C}' \cup \{ \text{freq}(\mathcal{I}) = \frac{1}{(2^n)} \}$ is satisfiable. These claims can easily be proved by checking the changes in the sums $\sigma_{\mathcal{I}}(I)$ given above. \square

5. FREQSAT{LTRANS}

In this section we show that knowing an upper bound on the length of the transactions does not affect the complexity of the FREQSAT-problem. Moreover, for any subset C of $\{ntrans, ndup\}$, $\text{FREQSAT}(\{ltrans\} \cup C)$ is equivalent to $\text{FREQSAT}(C)$.

Lemma 5. *Let \mathcal{J} be a finite set of items, $n = |\mathcal{J}|$, k is an integer with $1 \leq k \leq n$.*

Let \mathcal{D} be a transaction database that satisfies the following collection $\mathcal{C}_k[\mathcal{J}]$ of frequency expressions:

$$\begin{aligned} \forall j \in \mathcal{J} : \text{freq}(\{j\}) &= \binom{n-1}{k-1} / \binom{n}{k} \\ \forall j_1 \neq j_2 \in \mathcal{J} : \text{freq}(\{j_1, j_2\}) &= \binom{n-2}{k-2} / \binom{n}{k} \end{aligned}$$

Then, for all transactions (tid, J) in \mathcal{D} , the number of items in $J \cap \mathcal{J}$ equals k .

Sketch Let for all $i = 0 \dots n$,

$$\delta_i =_{def} |\{(tid, I) \in \mathcal{D} \mid |J \cap I| = i\}| .$$

It is clear that $|\mathcal{D}| = \sum_{i=1}^n \delta_i$. Let

$$S_i =_{def} \sum_{I \subseteq \mathcal{J}, |I|=i} \text{supp}(I, \mathcal{D}) .$$

Hence, since \mathcal{D} satisfies $\mathcal{C}_k[\mathcal{J}]$,

$$\begin{aligned} S_0 &= |\mathcal{D}|, & S_1 &= |\mathcal{D}| \cdot n \cdot \binom{n-1}{k-1} / \binom{n}{k} \\ S_2 &= |\mathcal{D}| \binom{n}{2} \binom{n-2}{k-2} / \binom{n}{k} \end{aligned}$$

Thus, $k(k-1)S_0 = (k-1)S_1 = 2S_2$.

Every transaction of length i has i subsets of length 1, and $i(i-1)/2$ subsets of length 2. So, it is also true that

$$\begin{aligned} S_0 &= \sum_{i=1}^n \delta_i, & S_1 &= \sum_{i=1}^n i\delta_i \\ S_2 &= \sum_{i=1}^n i(i-1)\delta_i/2 \end{aligned}$$

These last equalities in combination with

$$k(k-1)S_0 = (k-1)S_1 = 2S_2 ,$$

lead to

$$\begin{aligned} 0 &= kS_0 - S_1 = \sum_{i=1}^n (k-i)\delta_i \\ 0 &= (k-1)S_1 - 2S_2 = \sum_{i=1}^n i(k-i)\delta_i \end{aligned}$$

From this it can be shown that for all $i \neq k$, $\delta_i = 0$. \square

For every δ , the set of constraints $\mathcal{C}_k[\mathcal{J}]$ is satisfied by the database $\mathcal{D}_{k,\delta}$ that consists of δ transactions (tid, J) , for all $J \subseteq \mathcal{J}$ of length k . Hence, every database \mathcal{D} with all transactions having length exactly k , can be embedded in the database $\mathcal{D}_{k,|\mathcal{D}|}$. The next definition and theorem are based on this observation. Again an item d , not in \mathcal{J} is used to mark the transactions that identify the embedding of \mathcal{D} .

Definition 1. Let \mathcal{C} be the following set of frequency constraints:

$$\mathcal{C} = \{\text{freq}(I_1) \in [l_1, u_1], \dots, \text{freq}(I_m) \in [l_m, u_m]\} .$$

Let $\mathcal{J} = \bigcup_{j=1}^m I_j$, $n = |\mathcal{J}|$, $1 \leq k \leq n$, and let d be an item not in \mathcal{J} .

$$\lambda_k(\mathcal{C}) =_{def} \left\{ \text{freq}(\{d\}) = 1 / \binom{n}{k} \right\} \cup \mathcal{C}_k[\mathcal{J}] \cup \bigcup_{j=1}^m \left\{ \text{freq}(\{d\} \cup I_j) \in \left[l_j / \binom{n}{k}, u_j / \binom{n}{k} \right] \right\}$$

Example 4. The following databases satisfy respectively \mathcal{C} and $\lambda_2(\mathcal{C})$ for

$$\mathcal{C} = \{\text{freq}(\{a, b\}) = 0.5, \text{freq}(\{b, c\}) \in [0.4, 0.6]\} .$$

TID	Items
1	a, b
2	b, c

 \longrightarrow

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

Theorem 6. A set \mathcal{C} of frequency constraints is satisfiable by a database with all transactions of constant length k if and only if $\lambda_k(\mathcal{C})$ is in **FREQSAT**.

PROOF. Let \mathcal{C} be

$$\{\text{freq}(I_1) \in [l_1, u_1], \dots, \text{freq}(I_m) \in [l_m, u_m]\} ,$$

and let \mathcal{J} be the set of items $\bigcup_{j=1}^m I_j$.

If: Suppose that \mathcal{D} satisfies $\lambda_k(\mathcal{C})$. Let \mathcal{D}^d be the following database:

$$\mathcal{D}^d =_{def} \{(tid, J) \mid (tid, J \cup \{d\}) \in \mathcal{D}\}$$

The number of transactions in \mathcal{D}^d is $|\mathcal{D}| / \binom{n}{k}$, and the number of transactions in \mathcal{D}^d that contain I_j lays between $|\mathcal{D}| \cdot l_j / \binom{n}{k}$ and $|\mathcal{D}| \cdot u_j / \binom{n}{k}$. Hence, its frequency in \mathcal{D}^d is between l_j and u_j . Therefore, \mathcal{D}^d satisfies \mathcal{C} . Furthermore, because \mathcal{D} satisfies $\mathcal{C}_k[\mathcal{J}]$, Lemma 5 states that every transaction in \mathcal{D} contains exactly k items from \mathcal{J} . Henceforth, all transactions in \mathcal{D}^d have length k (d is not in \mathcal{J}).

Only If: Suppose that \mathcal{D} is a database with all transactions of length k that satisfies \mathcal{C} . We construct a database \mathcal{D}' with $|\mathcal{D}| \binom{n}{k}$ transactions that satisfy $\lambda_k(\mathcal{C})$. For every subset I of size k of \mathcal{J} , there will be $\text{supp}(I, \mathcal{D})$ transactions $(tid, I \cup \{d\})$, and $|\mathcal{D}| - \text{supp}(I, \mathcal{D})$ transactions (tid, I) in \mathcal{D}' . Thus, the absolute number of transactions containing $I \cup \{d\}$ in \mathcal{D}' is the same as the absolute number of transactions containing I in \mathcal{D} , but $|\mathcal{D}'|$ is $\binom{n}{k}$ times larger than $|\mathcal{D}|$. Hence, the frequency of $I_j \cup \{d\}$ in \mathcal{D}' equals $\text{freq}(I_j, \mathcal{D}) / \binom{n}{k}$. The frequency of d is $|\mathcal{D}| / (|\mathcal{D}| \cdot \binom{n}{k}) = 1 / \binom{n}{k}$. It also holds that \mathcal{D}' satisfies $\mathcal{C}_k[\mathcal{J}]$; the projection of \mathcal{D}' on \mathcal{J} consists of $|\mathcal{D}|$ copies of I , for every subset I of \mathcal{J} of size k . \square

Corollary 1. For all $\mathcal{C} \subseteq \{ntrans, ndup\}$, $\text{FREQSAT}(\{ltrans\} \cup \mathcal{C}) \equiv \text{FREQSAT}(\mathcal{C})$.

Sketch Let $\mathcal{C} = \{\text{freq}(I_j) \in [l_j, u_j] \mid j = 1 \dots m\}$ be a set of frequency constraints, and let $\mathcal{I} = \bigcup_{j=1}^m I_j$, $|\mathcal{I}| = n$.

$\text{FREQSAT}(\{ltrans\} \cup \mathcal{C}) \geq \text{FREQSAT}(\mathcal{C})$: the number of items is an upper bound for transaction length, and hence, $(\mathcal{C}, v_1, \dots, v_k)$ is in $\text{FREQSAT}\{c_1, \dots, c_k\}$ if and only if $(\mathcal{C}, n, v_1, \dots, v_k)$ is in $\text{FREQSAT}\{ltrans, c_1, \dots, c_k\}$.

$\text{FREQSAT}(\{ltrans\} \cup \mathcal{C}) \leq \text{FREQSAT}(\mathcal{C})$: the proof of this direction is based on Lemma 5. Let \mathcal{C} be a set of frequency constraints. Assume that \mathcal{C} is satisfiable by a transaction database \mathcal{D} , and \mathcal{D} has a maximal transaction size of n . Since Lemma 5 only holds for transaction databases of length *exactly* lt , we need to add extra items to compensate for transactions that are too short. When \mathcal{C} includes $ndup$, some care is required to avoid that the new items change the number of duplicates. \square

6. FREQSAT{NTRANS}

In the last section we saw that knowing a maximal transaction length does not add expressive power to **FREQSAT**. For the number of transactions $ntrans$, the question whether it adds to the *complexity* is open. We show that **FREQSAT** reduces to **FREQSAT**{ $ntrans$ }, and that **FREQSAT**{ $ntrans$ } is equivalent to *Intersection Pattern Problem* (**IP**) [8] w.r.t. computational complexity. **IP** is the following problem: *given an $n \times n$ matrix C with integer entries, do there exist sets S_1, \dots, S_n such that $|S_i \cap S_j| = C[i, j]$?* If such sets exist, C is called an *intersection pattern*. In [8], it is claimed that **IP** is **NP**-complete. However, the inclusion in **NP** has only been proven for the case the entries in the matrix C are bounded by a fixed constant [7]. For the general problem, the inclusion of **IP** in **NP** is still open.

For the entailment, we show that, unlike for **FREQSAT**, the set $\text{ENT}_I^n(\mathcal{C}) = \{\text{freq}(I, \mathcal{D}) \mid \mathcal{D} \models \mathcal{C}, |\mathcal{D}| \leq n\}$, is no longer an interval of the rational numbers. This is of course hardly surprising, since the frequencies in a database with at most n transactions can only be of the form $\frac{p}{q}$, with $0 \leq p \leq n$, and $1 \leq q \leq n$. Therefore, it would be more fair to ask the following question: if $\frac{p_1}{q}, \frac{p_2}{q} \in \text{ENT}_I^n(\mathcal{C})$, is it true that for every p with $p_1 \leq p \leq p_2$, also $\frac{p}{q} \in \text{ENT}_I^n(\mathcal{C})$? We will answer this question negatively. Moreover, given an arbitrary set $R = \{r_1, \dots, r_k\}$ of rational numbers, we will show that there exists a set of constraints \mathcal{C} , an itemset I , and a positive integer n , all having description size polynomial in the size of R , such that $\text{ENT}_I^n(\mathcal{C}) = R$. This shows that the properties of the **FREQSAT**-problem change fundamentally if we restrict the number of transactions.

6.1 Relation with FREQSAT

Theorem 7. **FREQSAT** \leq **FREQSAT**{ $ntrans$ }

PROOF. Given a **FREQSAT**-problem \mathcal{C} , by Lemma 1, there exists an upper bound n_C (with size polynomial in \mathcal{C}), such that if \mathcal{C} is satisfiable, then \mathcal{C} is satisfiable by a database of size maximally n_C . Hence, \mathcal{C} is in **FREQSAT** if and only if (\mathcal{C}, n_C) is in **FREQSAT**{ $ntrans$ }. \square

6.2 INTERSECTION PATTERN

We show that IP is equivalent to $\text{FREQSAT}\{ntrans\}$.

6.2.1 Reduction From IP to $\text{FREQSAT}\{ntrans\}$

It is clear that IP can be reduced to $\text{FREQSAT}\{ntrans\}$; if an $n \times n$ matrix C is an intersection pattern, then there exists a realization S_1, \dots, S_n of C such that $|\bigcup_{i=1}^n S_i|$ is at most $N(C) = \sum_{1 \leq i \leq n} C[i, i]$. Every instance of IP can now be reduced to the an instance of $\text{FREQSAT}\{ntrans\}$ as follows. The upper bound on the number of transactions is set to $N(C)$. We make sure that the number of transactions in a satisfying database is exactly $N(C)$ by adding the following constraint (e is a new item): $\text{freq}(\{e\}) = 1/N(C)$. We furthermore have items s_1, \dots, s_n . For every $1 \leq i, j \leq n$, the constraint $\text{freq}(\{s_i, s_j\}) = C[i, j]/N(C)$ is added. If \mathcal{D} is a satisfying database, then the sets $S_i = \{tid \mid (tid, J) \in \mathcal{D}, s_i \in J\}$, for $i = 1 \dots n$ form a realization of C , and vice versa.

Example 5. Let $C = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$. The following sets form a realization of C : $S_1 = \{1, 2\}, S_2 = \{1\}$. The corresponding $\text{FREQSAT}\{ntrans\}$ -problem is $(C, 3)$ with

$$C = \left\{ \begin{array}{l} \text{freq}(\{e\}) = 1/3, \quad \text{freq}(\{s_1\}) = 2/3, \\ \text{freq}(\{s_2\}) = 1/3, \quad \text{freq}(\{s_1, s_2\}) = 1/3 \end{array} \right\}.$$

The satisfying database of C that corresponds with the realization $S_1 = \{1, 2\}, S_2 = \{1\}$ is:

TID	Items
1	s_1, s_2
2	s_1
3	

6.2.2 Reduction From $\text{FREQSAT}\{ntrans\}$ to IP

We give a non-deterministic polynomial many-one reduction from $\text{FREQSAT}\{ntrans\}$ to IP. Such a reduction shows that if IP is in NP, then so is $\text{FREQSAT}\{ntrans\}$. We only illustrate the reduction with an example.

Let (C, nt) be an instance of the $\text{FREQSAT}\{ntrans\}$ problem. The first step in the reduction is to reduce the cardinalities of the sets in the input to 2. For example; a constraint $\text{freq}(\{a, b, c, d\}) \in [0.1, 0.3]$ in C , must be replaced with a number of constraints that only involve itemsets of cardinality at most 2. This would be easy if we knew the frequencies of the prefixes of $\{a, b, c, d\}$. Indeed; suppose that we know that $\text{freq}(\{a\}) = 0.5$, $\text{freq}(\{a, b\}) = 0.3$, $\text{freq}(\{a, b, c\}) = 0.2$, and $\text{freq}(\{a, b, c, d\}) = 0.1$. Then we could introduce new items $i_{\{a,b\}}$, $i_{\{a,b,c\}}$, and $i_{\{a,b,c,d\}}$. These items replace respectively $\{a, b\}$, $\{a, b, c\}$, and $\{a, b, c, d\}$. We enforce these semantics as follows:

$$\begin{aligned} \text{freq}(\{i_{\{a,b\}}\}) &= 0.3, \quad \text{freq}(\{i_{\{a,b\}}, a\}) = 0.3, \\ \text{freq}(\{i_{\{a,b\}}, b\}) &= 0.3, \quad \text{freq}(\{a, b\}) = 0.3 \\ \\ \text{freq}(\{i_{\{a,b,c\}}\}) &= 0.2, \quad \text{freq}(\{i_{\{a,b,c\}}, i_{\{a,b\}}\}) = 0.2, \\ \text{freq}(\{i_{\{a,b,c\}}, c\}) &= 0.2, \quad \text{freq}(\{i_{\{a,b\}}, c\}) = 0.2 \\ \\ \text{freq}(\{i_{\{a,b,c,d\}}\}) &= 0.1, \quad \text{freq}(\{i_{\{a,b,c,d\}}, i_{\{a,b,c\}}\}) = 0.1, \\ \text{freq}(\{i_{\{a,b,c,d\}}, d\}) &= 0.1, \quad \text{freq}(\{i_{\{a,b,c\}}, d\}) = 0.1 \end{aligned}$$

In this way, we can replace itemsets of high cardinality by a chain of sets of cardinality at most 2. Of course, in general, we do not know the exact frequencies of the prefixes of the

sets that are too long. Therefore, in the non-deterministic polynomial many-one reduction, we start by guessing them. If C has a solution, then there exists a correct guess.

In the second step, we have to encode the $\text{FREQSAT}\{ntrans\}$ -problem as a matrix C . We can at this point assume that C only contains itemsets of cardinality at most 2, and that the frequencies are given exactly (that is, no intervals). We guess the total number of transactions n , under the constraint $0 \leq n \leq nt$. In the matrix C , every row and column corresponds to one item. The entry $C[i, j]$ that corresponds to the item i and the item j is filled as follows: if there is an expression $\text{freq}(\{i, j\}) = f$ in C , then $C[i, j] = f \cdot n$. Else, the entry $C[i, j]$ is filled randomly by a number between 0 and n . If in the end, one of the entries in C is not an integer, we reject, since one of the guesses was wrong. In the other case, an instance for IP has been constructed. There exists a series of guesses that leads to an intersection pattern C if and only if the original problem (C, nt) is in $\text{FREQSAT}\{ntrans\}$.

6.3 Entailment

Lemma 2 can be extended to $\text{FREQSAT}\{ntrans\}$; that is, we can extend $\text{FREQSAT}\{ntrans\}$ to arbitrary Boolean formulas. We assume that the maximal number of transactions is set to nt . Consider the following set of expressions over the items a, b, c :

$$\begin{aligned} \text{freq}(\{i\}) &= 1/nt & \text{freq}(\{a, b\}) &= 0 \\ \text{freq}(\{a, c\}) &= 0 & \text{freq}(\{b, c\}) &= 0 \\ \text{freq}(a \vee c) &= k/nt & \text{freq}(b \vee c) &= k/nt \end{aligned}$$

The first constraint makes sure that there are exactly nt transactions. The next three constraints enforce that the transactions with a , the ones with b , and the ones with c are disjoint. Let A be the set of transactions with a , B the ones with b , and C the ones with c . The last two constraints express that $|A \cup C| = |B \cup C| = k/nt$. Let's now consider the set $\text{ENT}_{a \vee b}^{nt}(C)$. Suppose that C contains l items, $0 \leq l \leq k$. Then, both A and B contain $k - l$ transactions, and hence, $|A \cup B| = 2(k - l)$. Therefore, $\text{ENT}_{a \vee b}^{nt}(C) = \{\frac{2-l}{nt} \mid l = 0 \dots k\}$. Thus, $0/nt, 2/nt \in \text{ENT}_{a \vee b}^{nt}(C)$, but $1/nt$ is not in $\text{ENT}_{a \vee b}^{nt}(C)$.

We now show that we can express every arbitrary set. Let $R = \{r_1, \dots, r_k\}$ be a set of positive rational numbers between 0 and 1. First, we equalize the denominators, that is, let $R = \{\frac{p_1}{q}, \dots, \frac{p_k}{q}\}$. We set the number of transactions to q . We make sure that the number of transactions is exactly q , by adding the constraint $\text{freq}(\{i\}) = 1/q$. c_i and n_i are new items that are introduced for the construction. We construct a set of constraints such that $\text{freq}(\{n_i\})$ is either 0 or p_i/q . The expression will be such that $\text{freq}(\{n_i\}) = p_i/q$, if and only if $\text{freq}(\{c_i\}) = 0$. Hence, c_i acts as some sort of switch; if c_i is "on", $\text{freq}(n_i)$ will be 0. We will make sure in the construction that only one of the switches is "off." Let d_i be a new item. We first make a construction such that $\text{freq}(d_i) = 1/q$ if $\text{freq}(c_i) = 0$ and vice versa:

$$\text{freq}(c_i \vee d_i) = 1/q, \text{freq}(c_i \wedge d_i) = 0$$

Then we use the multiplication lemma to express that the frequency of n_i is p_i times the frequency of d_i :

$$\mathcal{M}_{p_i}(d_i, n_i).$$

We set a target item t to $n_1 \vee \dots \vee n_k$: $E(t, n_1 \vee \dots \vee n_k)$. We still have to make sure that only one of the n_i 's is non-zero. Remember that n_i was non-zero if and only if c_i was zero. Hence, we add

$$\text{freq}(c_1 \vee \dots \vee c_k) = (k - 1)/q$$

Thus, exactly $k - 1$ of the frequencies of c_1, \dots, c_k are $1/q$, and therefore, for exactly one i , $\text{freq}(c_i) = 0$. Let \mathcal{E} be the set of constraints we just constructed. It is now true that $\text{ENT}_t^q(\mathcal{E}) = R$.

Example 6. Consider the set $R = \{1/2, 1/3, 1/4\}$. First we equalize the denominators:

$$R = \{6/12, 4/12, 3/12\} .$$

We set the upper bound on the number of transactions to 12. We make sure that there are exactly 12 transactions by adding the constraint

$$\text{freq}(\{i\}) = 1/12 .$$

New items c_1, c_2, c_3 , and d_1, d_2, d_3 are introduced. We add the following constraints to ensure that $\text{freq}(d_j) = 1/12$ if and only if $\text{freq}(c_j) = 0$. Otherwise $\text{freq}(d_j) = 0$.

$$\begin{aligned} \text{freq}(c_1 \vee d_1) &= 1/12, \text{freq}(c_1 \wedge d_1) = 0 \\ \text{freq}(c_2 \vee d_2) &= 1/12, \text{freq}(c_2 \wedge d_2) = 0 \\ \text{freq}(c_3 \vee d_3) &= 1/12, \text{freq}(c_3 \wedge d_3) = 0 \end{aligned}$$

Next, the items n_1, n_2, n_3 are introduced that have a frequency of respectively $3 \cdot \text{freq}(d_1)$, $4 \cdot \text{freq}(d_2)$, and $6 \cdot \text{freq}(d_3)$:

$$\mathcal{M}_3(d_1, n_1), \mathcal{M}_4(d_2, n_2), \mathcal{M}_6(d_3, n_3)$$

We make sure that exactly 2 of c_1, c_2, c_3 are non-zero:

$$\text{freq}(c_1 \vee c_2 \vee c_3) = 2/12 .$$

Hence, exactly one of $\text{freq}(c_j)$ is 0, and thus, exactly one of $\text{freq}(d_j)$ is $1/12$, the other are 0. Therefore, either $\text{freq}(n_1) = 3/12$, $\text{freq}(n_2) = 0$, $\text{freq}(n_3) = 0$, or $\text{freq}(n_1) = 0$, $\text{freq}(n_2) = 4/12$, $\text{freq}(n_3) = 0$, or $\text{freq}(n_1) = 0$, $\text{freq}(n_2) = 0$, $\text{freq}(n_3) = 6/12$.

Finally, the item t is set to equal $n_1 \vee n_2 \vee n_3$:

$$E(t, n_1 \vee n_2 \vee n_3)$$

The set of frequencies for t entailed by this set of constraints equals $\{3/12, 4/12, 6/12\}$.

7. FREQSAT{NDUP}

In this section we study $\text{FREQSAT}\{ndup\}$. First we show that we can always reduce a $\text{FREQSAT}\{ndup\}$ -instance (\mathcal{C}, nd) to an instance $(\mathcal{C}', 1)$. Hence, we show that the following problem: given \mathcal{C} , decide whether $(\mathcal{C}, 1)$ is in $\text{FREQSAT}\{ndup\}$, is equivalent to $\text{FREQSAT}\{ndup\}$. We denote this problem $\text{FREQSAT}\{ndup = 1\}$.

We show that $\text{FREQSAT}\{ntrans\}$ reduces to $\text{FREQSAT}\{ndup\}$. We also show that $\text{FREQSAT}\{ndup = 1\}$ is **PP**-hard. Hence, knowing the number of duplicates does add complexity to the FREQSAT -problem.

7.1 FREQSAT{ndup=1}

Let \mathcal{C} be a set of constraints, and nd be a positive integer. Let the binary representation of nd be $B_l \dots B_0$. We introduce $l + 1$ new items, b_0, \dots, b_l . We use the b_j 's to eliminate duplicates. That is, $nd + 1$ transactions with set of items I , will be replaced by transactions with set of items: $I, I \cup \{b_0\}, I \cup \{b_1\}, I \cup \{b_0, b_1\}, \dots, I \cup \{b_j \mid B_j = 1\}$. Let $I \cup B$ be an itemset, $I \cap B = \emptyset$, and $b_j \in (B \cup I) \rightarrow b_j \in B$. $\nu(I \cup B)$ is the number associated with I ; that is:

$$\nu(I \cup B) = \sum_{b_j \in B} 2^j .$$

We have to make sure that the numbers of the transactions are never higher than nd . This can be done as follows: for all ℓ such that $B_\ell = 0$, add the constraint $\text{freq}(\{b_j \mid B_j = 1, j > \ell\} \cup \{b_\ell\}) = 0$. Let \mathcal{B}_{nd} be the set of these constraints.

$$\Delta_{nd}(\mathcal{C}) =_{def} \mathcal{C} \cup \mathcal{B}_{nd-1} .$$

It is now true that (\mathcal{C}, nt, nd) is in $\text{FREQSAT}\{ntrans, ndup\}$, if and only if $(\Delta_{nd}(\mathcal{C}), nt)$ is in $\text{FREQSAT}\{ntrans, ndup = 1\}$.

Example 7. The binary representation of 10 is 1010. Hence, \mathcal{B}_{10} is the following set of constraints:

$$\{\text{freq}(\{b_3, b_2\}) = 0, \text{freq}(\{b_3, b_1, b_0\}) = 0\} .$$

Every database that satisfies these constraints can have transactions (tid, J) with $J \cap \{b_0, b_1, b_2, b_3\}$ equal to:

$$\begin{aligned} \{ \}, \{b_0\}, \{b_1\}, \{b_0, b_1\}, \{b_2\}, \{b_0, b_2\}, \{b_1, b_2\} \\ \{b_0, b_1, b_2\}, \{b_3\}, \{b_0, b_3\}, \{b_1, b_3\} \end{aligned}$$

These transactions have respectively as associated numbers 0, ..., 10. The constraints in \mathcal{B}_{10} disallow transactions that contain

$$\{b_3, b_1, b_0\}, \{b_3, b_2\}, \{b_3, b_2, b_0\}, \{b_3, b_2, b_1\}, \{b_3, b_2, b_1, b_0\} .$$

These transactions have respectively as associated numbers 11, ..., 15.

Therefore, adding the items b_0, \dots, b_3 , and \mathcal{B}_{10} makes it possible to reduce the number of duplicates with a factor 11.

Theorem 8.

$$\begin{aligned} \text{FREQSAT}\{ndup\} &\equiv \text{FREQSAT}\{ntrans, ndup\} \\ \text{FREQSAT}\{ntrans\} &\leq \text{FREQSAT}\{ndup\} \end{aligned}$$

PROOF. $\text{FREQSAT}\{ndup\} \leq \text{FREQSAT}\{ntrans, ndup\}$: With n items and nd duplicates, one can have maximally $nd \cdot 2^n$ transactions. Hence, $(\mathcal{C}, nd) \in \text{FREQSAT}\{ndup\}$ if and only if $(\mathcal{C}, nd \cdot 2^n, nd) \in \text{FREQSAT}\{ntrans, ndup\}$.

$\text{FREQSAT}\{ndup = 1\} \geq \text{FREQSAT}\{ntrans, ndup = 1\}$: Let $\mathcal{C} = \{\text{freq}(I_j) \in [l_j, u_j], j = 1 \dots m\}$, $\mathcal{I} = \bigcup_{j=1}^m I_j$. Let $b_l \dots b_0$ be the binary representation of nt . $(\mathcal{C}, nt) \in \text{FREQSAT}\{ntrans, ndup = 1\}$ if and only if

$$\begin{aligned} \{\text{freq}(\{d\} \cup I_j) \in [l_j/2, u_j/2], j = 1 \dots m\} \\ \cup \{\text{freq}(\{d\}) = 0.5, \text{freq}(\bar{d}) = 0.5, \text{freq}(d, \bar{d}) = 0\} \\ \cup \mathcal{B}_{nt-1} \cup \{\text{freq}(\{b_j, d\}) = 0, j = 1 \dots l\} \\ \cup \{\text{freq}(\{i, \bar{d}\}) = 0 \mid i \in \mathcal{I}\} \end{aligned}$$

is in $\text{FREQSAT}\{ndup = 1\}$. In this reduction, the simulating database is split into two equally sized parts. The actual

database consists of the transactions containing d . In the other part, every transaction contains \bar{d} and some items of $\{b_0, \dots, b_l\}$. Since \mathcal{B}_{nt-1} holds, and the number of duplicates is 1, the \bar{d} -part has maximally nt transactions. Because both parts have equal size, the actual database, that is embedded as the d -part, contains maximally nt transactions as well.

$\text{FREQSAT}\{ndup\} \geq \text{FREQSAT}\{ntrans\}$:
 (\mathcal{C}, nt) is in $\text{FREQSAT}\{ntrans\}$ if and only if (\mathcal{C}, nt, nt) is in $\text{FREQSAT}\{ntrans, ndup\}$. \square

Theorem 9. $\text{FREQSAT}\{ndup = 1\}$ is in **PSPACE**

PROOF. Let $\mathcal{C} = \{\text{freq}(I_j) \in [l_j, u_j], j = 1 \dots m\}$, and let $\mathcal{I} = \bigcup_{j=1}^m I_j$. Every database \mathcal{D} that satisfies \mathcal{C} , and with $ndup(\mathcal{D}) \leq 1$, has at most $2^{|\mathcal{I}|}$ transactions.

We show a non-deterministic procedure to decide the satisfiability of \mathcal{C} that uses at most polynomial space in the length of \mathcal{C} . In this way we show that $\text{FREQSAT}\{ndup = 1\}$ is in **NPSpace**, and thus by Savitch's Theorem [15, p. 149-150], also in **PSPACE**.

We “guess” a database \mathcal{D} , transaction by transaction. We avoid generating the same transaction twice, by requiring that every new transaction comes lexicographically strictly after the previous one. During database generation, we maintain m counters for I_1, \dots, I_m , and 1 counter for $|\mathcal{D}|$. For every new transaction (tid, J) , we increment the counter $|\mathcal{D}|$, and we do the checks $I_j \subseteq J$. For all j such that $I_j \subseteq J$, the counter for I_j is incremented. After at most $2^{|\mathcal{I}|}$ guesses, we stop the database generation. We then check whether $Counter(I_j)/Counter(|\mathcal{D}|)$ is within the interval $[l_j, u_j]$. If this is the case for all $j = 1 \dots m$, we accept, otherwise, we reject. \square

7.2 $\text{FREQSAT}\{ndup\}$ is PP-hard

Theorem 10. $\text{FREQSAT}\{ndup\}$ is **PP-hard**.

Sketch We reduce MAJSAT to $\text{FREQSAT}\{ntrans, ndup = 1\}$. Let φ be the given formula with variables x_1, \dots, x_n . We construct a set of constraints \mathcal{C} , such that $(\mathcal{C}, 2^{n+1})$ is in $\text{FREQSAT}\{ntrans, ndup = 1\}$ if and only if more than half of the truth assignments to φ are accepting.

This reduction is very similar to the reduction of **pSAT** to **FREQSAT**. In $\text{FREQSAT}\{ntrans, ndup = 1\}$ we can furthermore make sure that every transaction represents a different truth assignment, and that the total number of truth assignments is 2^n . Hence, the frequency of a certain itemset $\{d, t_\sigma\}$ corresponds directly to the number of satisfying truth assignments. The requirement that φ is true in more than half of the truth assignments can thus be stated as follows:

$$\text{freq}(\{t_\varphi, d\}) \in [(2^{n-1} + 1)/2^{n+1}, 1],$$

since 2^n transactions represent valid truth assignments (the ones that contain d), and hence “more than half” is the same as “at least $(2^{n-1} + 1)$.” \square

7.3 Entailment

Notice that the construction in the proof of the **PP**-hardness of $\text{FREQSAT}\{ndup = 1\}$, has direct repercussions for the study of the complexity of the following function problem, named $\text{FuncFREQUENT}\{ndup = 1\}$.

Problem $\text{FuncFREQUENT}\{ndup = 1\}$:

Input: A pair (\mathcal{C}, I) , with \mathcal{C} a finite set of expressions

$$\mathcal{C} = \{I_j \in [l_j, u_j] \mid j = 1 \dots m\}$$

Output: Let $S = \{\text{freq}(I, \mathcal{D}) \mid \mathcal{D} \models \mathcal{C}, ndup(\mathcal{D}) = 1\}$. If \mathcal{C} is satisfiable, the interval $[\min(S), \max(S)]$.

Else, if \mathcal{C} is not satisfiable, no . \square

Theorem 11. $\text{FuncFREQUENT}\{ndup\}$ is **#P-hard**.

PROOF. Let φ be an arbitrary Boolean formula. Construct \mathcal{C} as in the proof of Theorem 10. Let now \mathcal{C}' be the constructed \mathcal{C} minus the constraint $\text{freq}(\{t_\varphi, d\}) \in [(2^{n-1} + 1)/2^{n+1}, 1]$. Let $[l, u]$ be output of the $\text{FuncFREQUENT}\{ndup = 1\}$ -problem $(\mathcal{C}, \{d, t_\varphi\})$. Via a similar reasoning as in the proof of Theorem 10, we get that l and u both equal the number of satisfying assignments of φ , divided by 2^{n+1} . Hence, $2^{n+1} \cdot l$ is the solution of the **#SAT**-problem with input φ . \square

8. APPLICATIONS

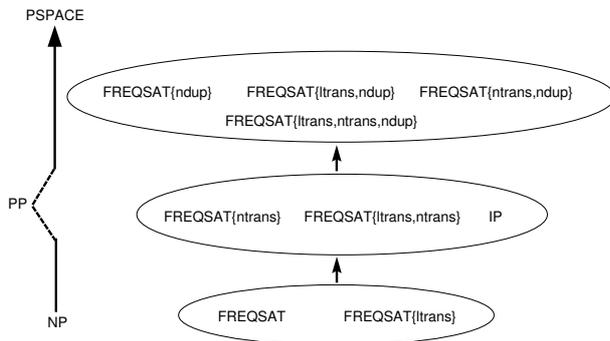
Privacy Data Mining can be a serious threat to the privacy. Therefore, methods are developed to adapt databases in such a way that still meaningful data mining results can be produced from it, but the privacy of the individual data are not compromised [2]. It is, however, conceivable that the mining is done by a trusted party. In that case, there is no risk of disclosure based on the original data. Even though, the results of the mining themselves can disclose more of the original data than is desirable. The process of trying to reconstruct parts of the original database from data mining results is called *inverse data mining* [13]. The **FREQSAT**-problem, its various variants and the entailment problems can be situated in this context. The results of a frequent set mining operation can be represented as an instance of **FREQSAT**. Inverse data mining would then amount to deriving the frequencies of other itemsets, not in the result set. In this context, the high complexities of the problems studied in this paper are bad news: suppose that we want to publish some itemsets with their frequencies, but first we want to assess how much these frequencies disclose of the original dataset. This problem can be stated as one of the variants of **FREQSAT**. The high complexity of the **FREQSAT**-problems in this paper, however, show that there is little hope that it is effectively possible to assess the degree of disclosure. On the bright side, the high complexity means also that it is potentially very hard to break the privacy. However, the situation is different from that of, for example, public key encryption. In inverse mining, partial information can be derived with incomplete methods, whereas, in general, in public key encryption, the code cannot be *partially* broken. Hence, in inverse mining, the more computing power one has, the more one can be derived. Therefore, unless one has superior computing power over potentially malicious parties, the results of mining cannot be guaranteed to be safe.

Condensed Representations Another application is making condensed representations [12] of frequent itemsets. In such condensed representations typically only non-redundant information is stored. Entailment of frequencies as in the **FREQSAT**-problem allows for derivation of frequencies. The stronger the deduction mechanism, the more redundancy in the set of frequencies can be found. The complexity results in this paper indicate that complete deduction in the most general context is infeasible, and hence, incomplete, yet tractable methods are more appropriate.

Frequent Itemset Mining Algorithms A third application is improving the pruning of frequent itemset mining algorithms. All frequent set mining algorithms use the monotonicity rule to prune substantial parts of the search space. This monotonicity rule can be seen as a very simple example of deduction. Based on partial frequency information of some itemsets, bounds on the frequencies of yet to be counted sets are derived. If these bounds establish that a certain set must be certainly frequent or certainly infrequent, the counting of it can be omitted in some cases. In the context of **FREQSAT**, frequency constraints can be used to model the frequency information gathered in previous scans over the database. The deduction can then be used to identify sets that are certainly frequent/infrequent. In [3, 4, 6], in some form, deduction rules are used in order to improve pruning and speed up frequent set mining algorithms.

9. SUMMARY AND CONCLUSION

The complexity of the **FREQSAT**-problem and its variants was studied. The following hierarchy illustrates the relations:



FREQSAT was shown to be **NP**-complete. It was also shown that the extensions to arbitrary formulas and to association rules do not add extra expressive power to **FREQSAT**.

The complexity of **FREQSAT{ntrans}** is still an open question. We proved that **FREQSAT{ntrans}** is **NP**-complete if **IP** is in **NP**. We also illustrated that **FREQSAT{ntrans}** has different properties than **FREQSAT** by showing that the set $\text{ENT}_I^{nt}(\mathcal{C})$ can be any set of rational numbers, whereas in **FREQSAT**, this set is always an interval of the rational numbers. This can be an indication that the complexity of **FREQSAT{ntrans}** is higher than the complexity of **FREQSAT**, since the linear programming techniques that can be used for **FREQSAT** and **pSAT** cannot be used for **FREQSAT{ntrans}**.

FREQSAT{ndup} is the most complex of the different variants of **FREQSAT**. Its complexity is between **PP** and **PSPACE**.

The exact complexity is unknown. Assuming that **NP** \neq **PP**, **FREQSAT{ndup}** is provably harder than **FREQSAT**. This very high complexity is bad news from a privacy point of view; it states that it is almost impossible to assess how much some itemsets with their frequencies reveal from the underlying database.

The main question left open in this paper is what the exact complexities of **FREQSAT{ntrans}** and **FREQSAT{ndup}** are.

10. 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] R. Agrawal, and R. Srikant. Privacy-preserving data mining. In *Proc. ACM SIGMOD*, pages 439–450, 2000.
- [3] Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, and L. Lakhal. Mining frequent patterns with counting inference. *SIGKDD Explorations*, 2(2):66–75, 2000.
- [4] R. J. Bayardo. Efficiently mining long patterns from databases. In *Proc. ACM SIGMOD*, pages 85–93, 1998.
- [5] T. Calders. Deducing bounds on the support of itemsets. In *Database Support for Data Mining Applications*, LNCS 2682, Springer, to appear 2004.
- [6] T. Calders and B. Goethals. Mining all non-derivable frequent itemsets. In *Proc. PKDD Int. Conf. Principles of Data Mining and Knowledge Discovery*, pages 74–85. Springer, 2002.
- [7] V. Chvátal. Recognizing intersection patterns. *Annals of Discrete Mathematics - Combinatorics 79*, 8(1):249–251, 1980.
- [8] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, New York, 1979.
- [9] G. Georgakopoulos, D. Kavvadias, and C. H. Papadimitriou. Probabilistic satisfiability. *Journal of Complexity*, 4:1–11, 1988.
- [10] J. Hipp, U. Güntzer, and G. Nakhaeizadeh. Algorithms for association rule mining - a general survey and comparison. *SIGKDD Explorations*, 2(1):58–64, 2000.
- [11] T. Lukasiewicz. Probabilistic logic programming with conditional constraints. *ACM Transactions on Computational Logic*, 2(3):289–339, 2001.
- [12] H. Mannila and H. Toivonen. Multiple uses of frequent sets and condensed representations. In *Proc. KDD Int. Conf. Knowledge Discovery in Databases*, 1996.
- [13] T. Mielikäinen. On inverse frequent set mining. In *Workshop on Privacy Preserving Data Mining*, 2003.
- [14] N. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28:71–87, 1986.
- [15] C.H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.