Acta Informatica manuscript No.
(will be inserted by the editor)

Nikolaj Tatti

Safe Projections of Binary Data Sets

January, 2006

Abstract Selectivity estimation of a boolean query based on frequent itemsets can be solved by describing the problem by a linear program. However, the number of variables in the equations is exponential, rendering the approach tractable only for small-dimensional cases. One natural approach would be to project the data to the variables occurring in the query. This can, however, change the outcome of the linear program.

We introduce the concept of safe sets: projecting the data to a safe set does not change the outcome of the linear program. We characterise safe sets using graph theoretic concepts and give an algorithm for finding minimal safe sets containing given attributes. We describe a heuristic algorithm for finding almost-safe sets given a size restriction, and show empirically that these sets outperform the trivial projection.

We also show a connection between safe sets and Markov Random Fields and use it to further reduce the number of variables in the linear program, given some regularity assumptions on the frequent itemsets.

Keywords Itemsets · Boolean Query Estimation · Linear Programming

Mathematics Subject Classification (2000) 68R10 · 90C05

CR Subject Classification G.3

1 Introduction

Consider the following problem: given a large, sparse matrix that holds boolean values, and a boolean formula on the columns of the matrix, approximate the probability that the formula is true for a random row of the matrix. A straightforward exact solution is to evaluate the formula on each

HIIT Basic Research Unit, Laboratory of Computer and Information Science, Helsinki University of Technology, Finland. E-mail: ntatti@cc.hut.fi

row. Now consider the same problem using instead of the original matrix a family of frequent itemsets, i.e., sets of columns where true values co-occur in a large fraction of all rows [1,2]. An optimal solution is obtained by applying linear programming in the space of probability distributions [11,19,3], but since a distribution has exponentially many components, the number of variables in the linear program is also large and this makes the approach infeasible. However, if the target formula refers to a small subset of the columns, it may be possible to remove most of the other columns without degrading the solution; somewhat surprisingly, it is not safe to remove all columns that do not appear in the formula. In this paper we investigate the question of which columns may be safely removed. Let us clarify this scenario with the following simple example.

Example 1 Assume that we have three attributes, say a, b, and c, and a data set D having five transactions

$$D = \left\{(1,0,1), (0,0,1), (0,1,1), (1,1,0), (1,0,0)\right\}.$$

Let us consider five itemsets, namely a, b, c, ab, and ac. The frequency of an itemset is the fraction of transactions in which all the attributes appearing in the itemset occur simultaneously. This gives us the frequencies $\theta_a = \frac{3}{5}$, $\theta_b = \frac{2}{5}$, $\theta_c = \frac{3}{5}$, $\theta_{ab} = \frac{1}{5}$, and $\theta_{ac} = \frac{1}{5}$. Let $\theta = \left[\theta_a, \theta_b, \theta_c, \theta_{ab}, \theta_{ac}\right]^T$. Let us now assume that we want to estimate the frequency of the formula $b \wedge c$. Consider now a distribution p defined on these three attributes. We assume that the distribution satisfies the frequencies, that is, $p(a = 1) = \theta_a$, $p(a = 1, b = 1) = \theta_{ab}$, etc. We want to find a distribution minimising/maximising $p(b \wedge c = 1)$. To convert this problem into a linear program we consider p as a real vector having $2^3 = 8$ elements. To guarantee that p is indeed a distribution we must require that p sum to 1 and that $p \geq 0$. The requirements that p must satisfy the frequencies can be expressed in a form $Ap = \theta$ for a certain A. In addition, $p(b \wedge c = 1)$ can be expressed as $c^T p$ for a certain c. Thus we have transform the original problem into a linear program

$$\min c^T p \quad \text{s.t.} \sum p = 1, p \geq 0, Ap = \theta.$$

Solving this program (and also the max-version of the program) gives us an interval $I = \left[\frac{1}{5}, \frac{2}{5}\right]$ for possible frequencies of $p(b \wedge c = 1)$. This interval has the following property: A rational frequency $\eta \in I$ if and only if there is a data set having the frequencies θ and having η as the fraction of the transactions satisfying the formula $b \wedge c$. If we, however, delete the attribute a from the data set and evaluate the boundaries using only the frequencies θ_b and θ_c, we obtain a different interval $I' = \left[0, \frac{2}{5}\right]$.

The problem is motivated by data mining, where fast methods for computing frequent itemsets are a recurring research theme [10]. A potential new application for the problem is privacy-preserving data mining, where the data is not made available except indirectly, through some statistics. The idea of using itemsets as a surrogate for data stems from [16], where inclusion-exclusion is used to approximate boolean queries. Another approach is to assume a model for the data, such as maximum entropy [21]. The linear programming approach requires no model assumptions.

The boolean query scenario can be seen as a special case for the following minimisation problem: Let K be the number of attributes. Given a family $\mathcal{F}$ of itemsets, frequencies θ for $\mathcal{F}$, and some function f that maps any distribution defined on a set $\{0,1\}^K$ to a real number find a distribution satisfying the frequencies θ and minimising f. To reduce the dimension K we assume that f depends only on a small subset, say B, of items, that is, if p is a distribution defined on $\{0,1\}^K$ and p_B is p marginalised to B, then we can write $f(p) = f(p_B)$. The projection is done by removing all the itemsets from $\mathcal{F}$ that have attributes outside B.

The question is, then, how the projection to B alters the solution of the minimisation problem. Clearly, the solution remains the same if we can always extend a distribution defined on B satisfying the projected family of itemsets to a distribution defined on all items and satisfying all itemsets in $\mathcal{F}$. We describe sufficient and necessary conditions for this extension property. This is done in terms of a certain graph extracted from the family $\mathcal{F}$. We call the set B safe if it satisfies the extension property.

If the set B is not safe, then we can find a safe set C containing B. We will describe an efficient polynomial-time algorithm for finding a safe set C containing B and having the minimal number of items. We will also show that this set is unique. We will also provide a heuristic algorithm for finding a restricted safe set C having at maximum M elements. This set is not necessarily a safe set and the solution to the minimisation problem may change. However, we believe that it is the best solution we can obtain using only M elements.

The rest of the paper is organised as follows: Some preliminaries are described in Section 2. The concept of a safe set is presented in Section 3 and the construction algorithm is given in Section 4. In Section 5 we explain in more details the boolean query scenario. In Section 6 we study the connection between safe sets and MRFs. Section 7 is devoted to restricted safe sets. We present empirical tests in Section 8 and conclude the paper with Section 9. Proofs for the theorems are given in Appendix.

2 Preliminaries

We begin by giving some basic definitions. A 0–1 *database* is a pair $\langle D, A \rangle$, where A is a set of items $\{a_1, \ldots, a_K\}$ and D is a *data set*, that is, a multiset of subsets of A.

A subset $U \subseteq A$ of items is called an *itemset*. We define an *itemset indicator function* $S_U : \{0,1\}^K \to \{0,1\}$ such that

$$S_U(z) = \begin{cases} 1, & z_i = 1 \text{ for all } a_i \in U \\ 0, & \text{otherwise} \end{cases} .$$

Throughout the paper we will use the following notation: We denote a random binary vector of length K by $X = X_A$. Given an itemset U we define X_U

to be the binary vector of length $|U|$ obtained from X by taking only the elements corresponding to U.

The *frequency* of the itemset U taken with respect of D, denoted by $U(D)$, is the mean of S_U taken with respect D, that is, $U(D) = \frac{1}{|D|}\sum_{z \in D} S_U(z)$. For more information on itemsets, see e.g. [1].

An *antimonotonic family* $\mathcal{F}$ of itemsets is a collection of itemsets such that for each $U \in \mathcal{F}$ each subset of U also belongs to $\mathcal{F}$. We define straightforwardly the itemset indicator function $S_{\mathcal{F}} = \{S_U \mid U \in \mathcal{F}\}$ and the frequency $\mathcal{F}(D) = \{U(D) \mid U \in \mathcal{F}\}$ for families of itemsets.

If we assume that $\mathcal{F}$ is an ordered family, then we can treat $S_{\mathcal{F}}$ as an ordinary function $S_{\mathcal{F}} : \{0,1\}^K \to \{0,1\}^L$, where L is the number of elements in $\mathcal{F}$. Also it makes sense to consider the frequencies $\mathcal{F}(D)$ as a vector (rather than a set). We will often use θ to denote this vector. We say that a distribution p defined on $\{0,1\}^K$ *satisfies* the frequencies θ, if $\mathrm{E}_p[S_{\mathcal{F}}] = \theta$.

Given a set of items C, we define a *projection* operator in the following way: A data set D_C is obtained from D by deleting the attributes outside C. A projected family of itemsets $\mathcal{F}_C = \{U \in \mathcal{F} \mid U \subseteq C\}$ is obtained from $\mathcal{F}$ by deleting the itemsets that have attributes outside C. The projected frequency vector θ_C is defined similarly. In addition, if we are given a distribution p defined on $\{0,1\}^K$, we define a distribution p_C to be the marginalisation of p to C. Given a distribution q over C we say that p is an *extension* of q if $p_C = q$.

3 Safe Projection

In this section we define a safe set and describe how such sets can be characterised using certain graphs.

We assume that we are given a set of items $A = \{a_1, \ldots, a_K\}$ and an antimonotonic family $\mathcal{F}$ of itemsets and a frequency vector θ for $\mathcal{F}$. We define $\mathbb{P}$ to be the set of all probability distributions defined on the set $\{0,1\}^K$. We assume that we are given a function $f : \mathbb{P} \to \mathbb{R}$ mapping a distribution to a real number. Let us consider the following problem:

$$\begin{array}{ll} \textsc{Problem P}: & \\ \text{Minimise} & f(p) \\ \text{subject to} & p \in \mathbb{P} \\ & \mathrm{E}_p[S_{\mathcal{F}}] = \theta. \end{array} \tag{1}$$

That is, we are looking for the minimum value of f among the distributions satisfying the frequencies θ. Generally speaking, this is a very difficult problem. Each distribution in $\mathbb{P}$ has 2^K entries and for large K even the evaluation of $f(p)$ may become infeasible. This forces us to make some assumptions on f. We assume that there is a relatively small set C such that f does not depend on the attributes outside C. In other words, we can define f by a function f_C such that $f_C(p_C) = f(p)$ for all p. Similarly, we define $\mathbb{P}_C$ to be the set of all distributions defined on the set $\{0,1\}^{|C|}$. We will now consider the following projected problem:

$$\begin{array}{ll} \textsc{Problem P}_C\text{:} & \\ \text{Minimise} & f_C(q) \\ \text{subject to} & q \in \mathbb{P}_C \\ & \mathrm{E}_q\left[S_{\mathcal{F}_C}\right] = \theta_C. \end{array}$$

Let us denote the minimising distribution of Problem P by $\hat{p}$ and the minimising distribution of Problem P_C by $\hat{q}$. It is easy to see that $f(\hat{p}) \geq f_C(\hat{q})$. In order to guarantee that $f(\hat{p}) = f_C(\hat{q})$, we need to show that C is safe as defined below.

Definition 1 Given an antimonotonic family $\mathcal{F}$ and frequencies θ for $\mathcal{F}$, a set C is *θ-safe* if for any distribution $q \in \mathbb{P}_C$ satisfying the frequencies θ_C, there exists an extension $p \in \mathbb{P}$ satisfying the frequencies θ. If C is safe for all θ, we say that it is *safe*.

Example 2 Let us continue Example 1. We saw that the outcome of the linear program changes if we delete the attribute a. Let us now show that the set $C = \{b, c\}$ is not a safe set. Let q be a distribution defined on the set C such that $q(b = 0, c = 0) = 0$, $q(b = 1, c = 0) = \frac{2}{5}$, $q(b = 0, c = 1) = \frac{3}{5}$, and $q(b = 1, c = 1) = 0$. Obviously, this distribution satisfies the frequencies θ_b and θ_c. However, we cannot extend this distribution to a such that all the frequencies are to be satisfied. Thus, C is not a safe set.

We will now describe a sufficient condition for safeness. We define a *dependency graph* G such that the vertices of G are the items $V(G) = A$ and the edges correspond to the itemsets in $\mathcal{F}$ having two items $E(G) = \{\{a_i, a_j\} \mid a_ia_j \in \mathcal{F}\}$. The edges are undirected. Assume that we are given a subset C of items and select $x \notin C$. A path $P = (a_{i_1}, \ldots, a_{i_L})$ from x to C is a graph path such that $x = a_{i_1}$ and only $a_{i_L} \in C$. We define a *frontier* of x with respect of C to be the set of the last items of all paths from x to C

$$\mathrm{front}\,(x, C) = \left\{a_{i_L} \mid P = (a_{i_1}, \ldots, a_{i_L}) \text{ is a path from } x \text{ to } C\right\}.$$

Note that $\mathrm{front}\,(x, C) = \mathrm{front}\,(y, C)$, if x and y are connected by a path not going through C. The following theorem gives a sufficient condition for safeness.

Theorem 1 *Let $\mathcal{F}$ be an antimonotonic family of itemsets. Let C be a set of items $C \subseteq A$ such that for each $x \notin C$ the frontier of x is in $\mathcal{F}$, that is,* $\mathrm{front}\,(x, C) \in \mathcal{F}$*. It follows that C is a safe set.*

The vague intuition behind Theorem 1 is the following: x has influence on C only through $\mathrm{front}\,(x, C)$. If $\mathrm{front}\,(x, C) \in \mathcal{F}$, then the distributions marginalised to $\mathrm{front}\,(x, C)$ are *fixed* by the frequencies. This means that x has no influence on C and hence it can be removed.

We saw in Examples 1 and 2 that the projection changes the outcome if the projection set is not safe. This holds also in the general case:

Theorem 2 *Let $\mathcal{F}$ be an antimonotonic family of itemsets. Let C be a set of items $C \subseteq A$ such that there exists $x \notin C$ whose frontier is not in $\mathcal{F}$, that is,* $\mathrm{front}\,(x, C) \notin \mathcal{F}$*. Then there are frequencies θ for $\mathcal{F}$ such that C is not θ-safe.*

Safeness implies that we can extend every satisfying distribution q in Problem P_C to a satisfying distribution p in Problem P. This implies that the optimal values of the problems are equal:

Theorem 3 *Let $\mathcal{F}$ be an antimonotonic family of itemsets. If C is a safe set, then the minimum value of Problem P is equal to the minimum value of Problem P_C for any query function and for any frequencies θ for $\mathcal{F}$.*

If the condition of being safe does not hold, that is, there is a distribution q that cannot be extended, then we can define a query f resulting 0 if the input distribution is q, and 1 otherwise. This construction proves the following theorem:

Theorem 4 *Let $\mathcal{F}$ be an antimonotonic family of itemsets. If C is not a safe set, then there is a function f and frequencies θ for $\mathcal{F}$ such that the minimum value of Problem P is strictly larger than the minimum value of Problem P_C.*

Example 3 Assume that we have 6 attributes, namely, $\{a, b, c, d, e, f\}$, and an antimonotonic family $\mathcal{F}$ whose maximal itemsets are ab, bc, cd, ad, de, ce, and af. The dependency graph is given in Fig. 1.

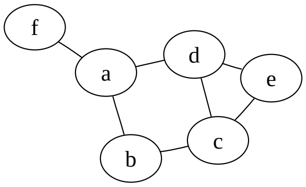

Fig. 1 An example of dependency graph.

Let $C_1 = \{a, b, c\}$. This set is not a safe set since front $(d, C_1) = ac \notin \mathcal{F}$. On the other hand the set $C_2 = \{a, b, c, d\}$ is safe since front $(f, C_2) = a \in \mathcal{F}$ and front $(e, C_2) = cd \in \mathcal{F}$.

The proof of Theorem 1 reveals also an interesting fact:

Theorem 5 *Let $\mathcal{F}$ be an antimonotonic family of itemsets and let θ be frequencies for $\mathcal{F}$. Let C be a safe set. Let p^{ME} be the maximum entropy distribution defined on A and satisfying θ. Let q^{ME} be the maximum entropy distribution defined on C and satisfying the projected frequencies θ_C. Then q^{ME} is p^{ME} marginalised to C.*

The theorem tells us that if we want to obtain the maximum entropy distribution marginalised to C and if the set C is safe, then we can remove the items outside C. This is useful since finding maximum entropy using Iterative Fitting Procedure requires exponential amount of time [7,12]. Using maximum entropy for estimating the frequencies of itemsets has been shown to be an effective method in practice [21]. In addition, if we estimate the frequencies of several boolean formulae using maximum entropy distribution marginalised to safe sets, then the frequencies are consistent. By this we mean that the frequencies are all evaluated from the same distribution, namely p^{ME}.

4 Constructing a Safe Set

Assume that we are given a function f that depends only on a set B, not necessarily safe. In this section we consider a problem of finding a safe set C such that $B \subseteq C$ for a given B. Since there are usually several safe sets that include B, for example, the set of all attributes A is always a safe set, we want to find a safe set having the minimal number of attributes. In this section we will describe an algorithm for finding such a safe set. We will also show that this particular safe set is unique.

The idea behind the algorithm is to augment B until the safeness condition is satisfied. However, the order in which we add the items into B matters. Thus we need to order the items. To do this we need to define a few concepts: A *neighbourhood* $N(x \mid r)$ of an item x of radius r is the set of the items reachable from x by a graph path of length at most r, that is,

$$N(x \mid r) = \{y \mid \exists P : x \rightarrow y, |P| \leq r\}. \tag{2}$$

In addition, we define a *restricted neighbourhood* $N_C(x \mid y)$ which is similar to $N(x \mid r)$ except that now we require that only the last element of the path P in Eq. 2 can belong to C. Note that $N_C(x \mid r) \cap C \subseteq \operatorname{front}(x, C)$ and that the equality holds for sufficiently large r.

The *rank* of an item x with respect of C, denoted by $\operatorname{rank}(x \mid C)$, is a vector v of length $|A| - 1$ such that v_i is the number of elements in C to whom the shortest path from x has the length i, that is,

$$v_i = |C \cap (N_C(x \mid i) - N_C(x \mid i - 1))|.$$

We can compare ranks using the bibliographic order. In other words, if we let $v = \operatorname{rank}(x \mid C)$ and $w = \operatorname{rank}(y \mid C)$, then $\operatorname{rank}(x \mid C) < \operatorname{rank}(y \mid C)$ if and only if there is an integer M such that $v_M < w_M$ and $v_i = w_i$ for all $i = 1, \ldots, M - 1$.

We are now ready to describe our search algorithm. The idea is to search the items that violate the assumption in Theorem 1. If there are several candidates, then items having the maximal rank are selected. Due to efficiency reasons, we do not look for violations by calculating $\operatorname{front}(x, C)$. Instead, we check whether $N_C(x \mid r) \cap C \in \mathcal{F}$. This is sufficient because

$$N_C(x \mid r) \cap C \notin \mathcal{F} \implies \operatorname{front}(x, C) \notin \mathcal{F}.$$

This is true because $N_C(x \mid r) \cap C \subseteq \operatorname{front}(x, C)$ and $\mathcal{F}$ is antimonotonic. The process is described in full detail in Algorithm 1.

We will refer to the safe set Algorithm 1 produces as $\operatorname{safe}(B \mid \mathcal{F})$. We will now show that $\operatorname{safe}(B \mid \mathcal{F})$ is the smallest possible, that is,

$$|\operatorname{safe}(B \mid \mathcal{F})| = \min\{|Y| \mid B \subseteq Y, Y \text{ is a safe set}\}.$$

The following theorem shows that in Algorithm 1 we add only necessary items into C during each iteration.

Theorem 6 *Let C be a set of items during some iteration of Algorithm 1 and let $Z = \{x \in W \mid \operatorname{rank}(x \mid C) = v\}$ be the set of items as it is defined in Algorithm 1. Let Y be any safe set containing C. Then it follows that $Z \subseteq Y$.*

Algorithm 1 The algorithm for finding a safe set C. The required input is B, the set that should be contained in C, and an antimonotonic family $\mathcal{F}$ of itemsets. The graph G is the dependency graph evaluated from $\mathcal{F}$.

```
$C \Leftarrow B$.
repeat
  $r \Leftarrow 1$.
  $V \Leftarrow \{x \mid \exists y \in C, xy \in E(G)\} - C$ {$V$ contains the neighbours of $C$.}
  repeat
    For each $x \in V$, $U_x \Leftarrow N_C(x \mid r) \cap C$.
    if there exists $U_x$ such that $U_x \notin \mathcal{F}$ then
      Break {A violation is found.}
    end if
    $r \Leftarrow r + 1$.
  until no $U_x$ changed
  if there is a violation then
    $W \Leftarrow \{x \in V \mid U_x \notin \mathcal{F}\}$ {$W$ contains the violating items.}
    $v \Leftarrow \max\{\mathrm{rank}(x \mid C) \mid x \in W\}$.
    $Z \Leftarrow \{x \in W \mid \mathrm{rank}(x \mid C) = v\}$
    $C \Leftarrow C \cup Z$ {Augment $C$ with the violating items having the largest rank.}
  end if
until there are no violations.
```

Corollary 1 *A safe set containing B containing the minimal number of items is unique. Also, this set is contained in each safe set containing B.*

Corollary 2 *Algorithm 1 produces the optimal safe set.*

Example 4 Let us continue Example 3. Assume that our initial set B is $\{a, b, c\}$. We note that $\mathrm{front}(d, B) = \mathrm{front}(e, B) = ac \notin \mathcal{F}$. Therefore, B is not a safe set. The ranks are $\mathrm{rank}(d \mid B) = 2$ and $\mathrm{rank}(e \mid B) = [1, 1]^T$ (the trailing zeros are removed). It follows that the rank of d is larger than the rank of e and therefore d is added into B during Algorithm 1. The resulting set $C = \{a, b, c, d\}$ is the minimal safe set containing B.

5 Frequencies of Boolean Formulae.

A boolean formula $f : \{0, 1\}^K \to \{0, 1\}$ maps a binary vector to a binary value. Given a family $\mathcal{F}$ of itemsets and frequencies θ for $\mathcal{F}$ we define a *frequency interval*, denoted by $\mathrm{fi}(f \mid \mathcal{F}, \theta)$, to be

$$\mathrm{fi}(f \mid \mathcal{F}, \theta) = \{\mathrm{E}_p[f] \mid \mathrm{E}_p[S_{\mathcal{F}}] = \theta\},$$

that is, a set of possible frequencies coming from the distribution satisfying given frequencies. For example, if the formula f is of form $a_1 \wedge \ldots \wedge a_M$, then we are approximating the frequency of a possibly unknown itemset.

Note that this set is truly an interval and its boundaries can be found using the optimisation problem given in Eq. 1. It has been shown that finding the boundaries can be reduced to a linear programming [11,19,3]. However, the problem is exponential in K and therefore it is crucial to reduce the dimension. Let us assume that the boolean formula depends only on the variables coming from some set, say B. We can now use Algorithm 1 to find a safe set C including B and thus to reduce the dimension.

Example 5 Let us continue Example 3. We assign the following frequencies to the itemsets: $\theta_x = 0.5$ where $x \in \{a, b, c, d, e, f\}$, $\theta_{bd} = 0.5$, $\theta_{cd} = 0.4$, and the frequencies of the rest itemsets in $\mathcal{F}$ are equal to 0.25. We consider the formula $f = b \wedge c$. In this case f depends only on $B = \{b, c\}$. If we project directly to B, then the frequency is equal to $\operatorname{fi}(f \mid \mathcal{F}_B, \theta_B) = [0, 0.5]$.

The minimal safe set containing B is $C = \{a, b, c, d\}$. Since $\theta_{bd} = 0.5$ it follows that b is equivalent to d. This implies that the frequency of f must be equal to $\operatorname{fi}(f \mid \mathcal{F}_C, \theta_C) = \theta_{cd} = 0.4$.

There exists many problems similar to ours: A well-studied problem is called PSAT in which we are given a CNF-formula and probabilities for each clause asking whether there is a distribution satisfying these probabilities. This problem is **NP**-complete [9]. A reduction technique for the minimisation problem where the constraints and the query are allowed to be conditional is given in [14]. However, this technique will not work in our case since we are working only with unconditional queries. A general problem where we are allowed to have first-order logic conditional sentences as the constraints/queries is studied in [15]. This problem is shown to be **NP**-complete. Though these problems are of more general form they can be emulated with itemsets [4]. However, we should note that in the general case this construction does not result an antimonotonic family.

There are many alternative ways of approximating boolean queries based on statistics: For example, the use of wavelets has been investigated in [17]. Query estimation using histograms was studied in [18] (though this approach does not work for binary data). We can also consider assigning some probability model to data such as Chow-Liu tree model or mixture model (see e.g. [22,21,6]). Finally, if B is an itemset and we know all the proper subsets of B and B is safe, then to estimate the frequency of B we can use inclusion-exclusion formulae given in [5].

6 Safe Sets and Junction Trees

Theorem 1 suggests that there is a connection between safe sets and Markov Random Fields (see e.g. [13] for more information on MRF). In this section we will describe how the minimal safe sets can be obtained from junction trees. We will demonstrate through a counter-example that this connection cannot be used directly. We will also show that we can use junction trees to reformulate the optimisation problem and possibly reduce the computational burden.

6.1 Safe Sets and Separators

Let us assume that the dependency graph G obtained from a family $\mathcal{F}$ of itemsets is *triangulated*, that is, the graph does not contain chordless circuits of size 4 or larger. In this case we say that $\mathcal{F}$ is triangulated. For simplicity, we assume that the dependency graph is connected. We need some concepts from Markov Random Field theory (see e.g. [13]): The *clique graph* is a graph having cliques of G as vertices and two vertices are connected if the corresponding cliques share a mutual item. Note that this graph is connected. A spanning tree of the clique graph is called a *junction tree* if it has a *running intersection* property. By this we mean that if two cliques contain the same item, then each clique along the path in the junction tree also contains the same item. An edge between two cliques is called a *separator*, and we associate with each separator the set of items mutual to both cliques.

We also make some further assumptions concerning the family $\mathcal{F}$: Let V be the set of items of some clique of the dependency graph. We assume that every proper subset of V is in $\mathcal{F}$. If $\mathcal{F}$ satisfies this property for each clique, then we say that $\mathcal{F}$ is *clique-safe*. We do not need to have $V \in \mathcal{F}$ because there is no node having an entire clique as a frontier.

Let us now investigate how safe sets and junction trees are connected. First, fix some junction tree, say T, obtained from G. Assume that we are given a set B of items, not necessarily safe. For each item $b \in B$ we select some clique $Q_b \in V(T)$ such that $b \in Q_b$ (same clique can be associated with several items). Let $b, c \in B$ and consider the path in T from Q_b to Q_c. We call the separators along such paths *inner separators*. The other separators are called *outer separators*. We always choose cliques Q_b such that the number of inner separators is the smallest possible. This does not necessarily make the choice of the cliques unique, but the set of inner separators is always unique. We also define an *inner clique* to be a clique incident to some inner separator. We refer to the other cliques as *outer cliques*.

Example 6 Let us assume that we have 5 items, namely $\{a, b, c, d, e\}$. The dependency graph, its clique graph, and the possible junction trees are given in Figure 2.

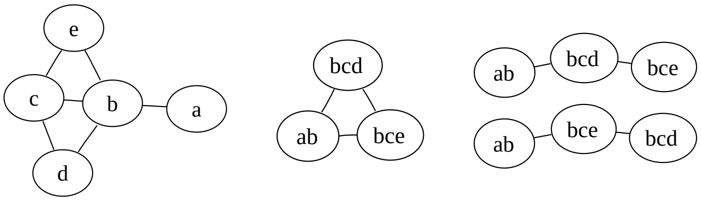

Fig. 2 An example of an dependency graph, a corresponding clique graph, and the possible junction trees.

Let $B = \{a, d\}$. Then the inner separator in the upper junction tree is the left edge. In the lower junction tree both edges are inner separators.

The following three theorems describe the relation between the safe sets containing B and the inner separators.

Theorem 7 *Let $\mathcal{F}$ be an antimonotonic, triangulated and clique-safe family of itemsets. Let T be a junction tree. Let C be a set containing B and all the items from the inner separators of B. Then C is a safe set.*

The following corollary follows from Corollary 1.

Corollary 3 *Let $\mathcal{F}$ be an antimonotonic, triangulated and clique-safe family of itemsets. Let T be a junction tree. The minimal safe set containing B may contain (in addition to the set B) only items from the inner separators of B.*

Theorem 8 *Let $\mathcal{F}$ be an antimonotonic, triangulated and clique-safe family of itemsets. There exists a junction tree such that the minimal safe set is precisely the set B and the items from the inner separators of B.*

Theorem 8 raises the following question: Is there a tree, *not* depending on B, such that the minimal safe set is precisely the set B and the items from the inner separators. Unfortunately, this is not the case as the following example shows.

Example 7 Let us continue Example 6. Let $B_1 = \{a, d\}$ and $B_2 = \{a, e\}$. The corresponding minimal safe sets are $C_1 = \{a, b, d\}$ and $C_2 = \{a, b, e\}$. The first case corresponds to the upper junction tree given in Figure 2, and the latter case corresponds the lower junction tree.

6.2 Reformulation of the Optimisation Problem Using Junction Trees

We have seen that a optimisation problem can be reduced to a problem having $2^{|C|}$ variables, where C is a safe set. However, it may be the case that C is very large. For example, imagine that the dependency graph is a single path $(a_{i_1}, \ldots, a_{i_L})$ and we are interested in finding the frequency for $a_{i_1} \wedge a_{i_L}$. Then the safe set contains the entire path. In this section we will try to reduce the computational burden even further.

The main benefit of MRF is that we are able to represent the distribution as a fraction of certain distributions. We can use this factorisation to encode the constraints. A small drawback is that we may not be able to express easily the distribution defined on B, the set of which the query depends. This happens when B is not contained in any clique. This can be remedied by adding edges to the dependency graph.

Let us make the previous discussion more rigorous. Let f be a query function and let B be the set of attributes of which f depends. Let $C = \operatorname{safe}(B \mid \mathcal{F})$ be the minimal safe set containing B. Project the items outside C and let G be the connectivity graph obtained from $\mathcal{F}_C$. We add some additional edges to G. First, we make the set B fully connected. Second, we triangulate the graph. Let T be a junction tree of the resulting graph.

Since B is fully connected, there is a clique Q_r such that $B \subseteq Q_r$. For each clique Q_i in T we define p_i to be a distribution defined on Q_i. Similarly, for each separator S_j we define q_j to be a distribution defined on S_j. Denote by $\mathbb{S}_i$ the collection of separators of a clique Q_i.

$$\begin{array}{ll}
\textsc{Problem LP}: & \\
\text{Minimise} & f(p_r) \\
\text{subject to} & \text{For each } Q_i \in V(T), \\
 & \quad p_i \text{ satisfies } \theta_{Q_i} \\
 & \quad p_i \text{ is an extension of } q_j \\
 & \quad \text{for each } S_j \in \mathbb{S}_i.
\end{array} \tag{3}$$

The following theorem states that the above formulation is correct:

Theorem 9 *The problem in Eq. 3 solves correctly the optimisation problem.*

Note that we can remove all q_j by combining the constraining equations. Thus we have replaced the original optimisation problem having $2^{|C|}$ variables with a problem having $\sum 2^{|Q_i|}$ variables. The number of cliques in T is bounded by $|C|$, the number of attributes in the safe set. To see this select any leaf clique Q_i. This clique must contain a variable that is not contained in any other clique because otherwise Q_i is contained in its parent clique. We remove Q_i and repeat this procedure. Since there are only $|C|$ attributes, there can be only $|C|$ cliques. Let M be the size of the maximal clique. Then the number of variables is bounded by $|C|2^M$. If M is small, then solving the problem is much easier than the original formulation.

Example 8 Assume that we have a family of itemsets whose dependency graph G is a path $(a_{i_1}, \ldots, a_{i_L})$ and that we want to evaluate the boundaries for a formula $a_{i_1} \wedge a_{i_L}$. We cannot neglect any variable inside the path, hence we have a linear program having 2^L variables.

By adding the edge $\{a_{i_1}, a_{i_L}\}$ to G we obtain a cycle. To triangulate the graph we add the edges $\left\{a_{i_1}, a_{i_j}\right\}$ for $3 \leq j \leq L-1$. The junction tree in consists of $L-2$ cliques of the form $a_{i_1} a_{i_j} a_{i_{j+1}}$, where $2 \leq j \leq L-1$. The reformulation of the linear program gives us a program containing only $(L-2)\, 2^3$ variables.

7 Restricted Safe Sets

Given a set B Algorithm 1 constructs the minimal safe set C. However, the set C may still be too large. In this section we will study a scenario where we require that the set C should have M items, at maximum. Even if such a safe set may not exist we will try to construct C such that the solution of the original minimisation problem described in Eq. 1 does not alter. As a solution we will describe a heuristic algorithm that uses the information available from the frequencies.

First, let us note that in the definition of a safe set we require that we can extend the distribution for any frequencies. In other words, we assume that the frequencies are the worst possible. This is also seen in Algorithm 1 since the algorithm does not use any information available from the frequencies.

Let us now consider how we can use the frequencies. Assume that we are given a family $\mathcal{F}$ of itemsets and frequencies θ for $\mathcal{F}$. Let C be some (not necessarily a safe) set. Let $x \notin C$ be some item violating the safeness condition. Assume that each path from x to C has an edge $e = (u, v)$ having the following property: Let θ_{uv}, θ_u, and θ_v be the frequencies of the itemsets uv, u, and v, respectively. We assume that $\theta_{uv} = \theta_u \theta_v$ and that the itemset uv is not contained in any larger itemset in $\mathcal{F}$. We denote the set of such edges by E.

Let W be the set of items reachable from x by paths not using the edges in E. Note that the set W has the same property than x. We argue that we can remove the set W. This is true since if we are given a distribution p defined on $A - W$, then we can extend this distribution, for example, by setting $p(X_A) = p^{ME}(X_W)p(X_{A-W})$, where $p^{ME}(X_W)$ is the maximum entropy distribution defined on W. Note that if we remove the edges E, then Algorithm 1 will not include W.

Let us now consider how we can use this situation in practice. Assume that we are given a function w which assign to each edge a non-negative weight. This weight represents the correlation of the edge and should be 0 if the independence assumption holds. Assume that we are given an item $x \notin C$ violating the safeness condition but we cannot afford adding x into C. Define H to be the subgraph containing x, the frontier $\mathrm{front}(x, C)$ and all the intermediate nodes along the paths from x to C. We consider finding a set of edges E that would cut x from its frontier and have the minimal cost $\sum_{e \in E} w(e)$. This is a well-known min-cut problem and it can be solved efficiently (see e.g. [20]). We can now use this in our algorithm in the following way: We build the minimal safe set containing the set B. For each added item we construct a cut with a minimal cost. If the safe set is larger than a constant M, we select from the cuts the one having the smallest weight. During this selection we neglect the items that were added before the constraint M was exceeded. We remove the edges and the corresponding itemsets and restart the construction. The algorithm is given in full detail in Algorithm 2.

Algorithm 2 The algorithm for finding a restricted safe set C. The required input is B, the set that should be contained in C, an antimonotonic family $\mathcal{F}$ of itemsets, a constant M which is an upper bound for $|C|$, and a weight function w for the edges. The graph G is the dependency graph evaluated from $\mathcal{F}$.

```
$C \Leftarrow B$.
repeat
  Find a violating item $x$ having the largest rank.
  if $|C| + 1 > M$ then
    Let $H$ be the graph containing $x$, $\mathrm{front}(x, C)$ and all the intermediate nodes.
    Let $E_x$ be the min-cut of $H$ cutting $x$ and $\mathrm{front}(x, C)$ from each other.
    Let $v_x$ be the cost of $E_x$.
  end if
  $C \Leftarrow C + x$.
until there are no violations.
if $|C| > M$ then
  Let $x$ be the item such that $v_x$ is the smallest possible.
  Remove the edges $E_x$ from the dependency graph.
  Remove the itemsets corresponding to the edges from $\mathcal{F}$.
  Remove also possible higher-order itemsets to preserve the antimonotonicity of $\mathcal{F}$.
  Restart the algorithm.
end if
```

Example 9 We continue Example 5. As a weight function for the edges we use the mutual information. This gives us $w_{bd} = 0.6931$ and $w_{cd} = 0.1927$. The rest of the weights are 0. Let $B = \{b, c\}$. We set the upper bound for the size of the safe set to be $M = 3$. The minimal safe set is $C = \{a, b, c, d\}$. The min cuts are $E_a = \{(a,b), (a,c)\}$ and $E_d = \{(d,b), (d,c)\}$. The corresponding weights are $v_a = 0$ and $v_d = w_{bd} + w_{cd} > 0$. Thus by cutting the edges E_a we obtain the set $C^r = \{b,c,d\}$. The frequency interval for the formula $b \wedge c$ is $\mathrm{fi}\left(f \mid \mathcal{F}_{C^r}, \theta_{C^r}\right) = 0.4$ which is the same as in Example 5.

8 Empirical Tests

We performed empirical tests to assess the practical relevance of the restricted safe sets, comparing it to the (possibly) unsafe trivial projection. We mined itemset families from two data sets, and estimated boolean queries using both the safe projection and the trivial projection. The first data set, which we call *Paleo*[1], describes fossil findings: the attributes correspond to genera of mammals, the transactions to excavation sites. The *Paleo* data is sparse, and the genera and sites exhibit strong correlations. The second data set, which we call *Mushroom*, was obtained from the FIMI repository[2]. The data is relatively dense.

First we used the Apriori [2] algorithm to retrieve some families of itemsets. A problem with Apriori was that the obtained itemsets were concentrated on the attributes having high frequency. A random query conducted on such a family will be safe with high probability — such a query is trivial to solve. More interesting families would the ones having almost all variables interacting with each other, that is, their dependency graphs have only a small number of isolated nodes. Hence we modified APriori: Let A be the set containing all items and for each $a \in A$ let $m(a)$ be the frequency of a. Let m be the smallest frequency $m = \min_{a \in A} m(a)$ and define $s(a) = m(a)/m$. Let U be an itemset and let θ_U be its frequency. Define $\eta_U = \prod_{a \in U} s(a)$. We modify Apriori such that the itemset U is in the output if and only if the ratio θ_U/η_U is larger than given threshold σ. Note that this family is anti-monotonic and so Apriori can be used. By this modification we are trying to give sparse items a fair chance and in our tests the relative frequencies did produce more scattered families.

For each family of itemsets we evaluated 10000 random boolean queries. We varied the size of the queries between 2 and 4. At first, such queries seem too simple but our initial experiments showed that these queries do result large safe sets. A few examples are given in Figure 3. In most of the queries the trivial projection is safe but there are also very large safe sets. Needless to say that we are forced to use restricted safe sets.

Given a query f we calculated two intervals $i_1(f) = \mathrm{fi}\left(f \mid \mathcal{F}_B, \theta_B\right)$ and $i_2(f) = \mathrm{fi}\left(f \mid \mathcal{F}_C, \theta_C\right)$ where B contains the attributes of f and C is the restricted safe set obtained from B using Algorithm 2. In other words, $i_1(f)$ is obtained by using the trivial projection and $i_2(f)$ is obtained by projecting to the restricted safe set. As parameters for Algorithm 2 we set the upper bound $M = 8$ and the weight function w to be the mutual information.

We divided queries into two classes. A class Trivial contained the queries in which the trivial projection and the restricted safe set were equal. The rest

[1] *Paleo* was constructed from NOW public release 030717 available from [8].

[2] `http://fimi.cs.helsinki.fi`

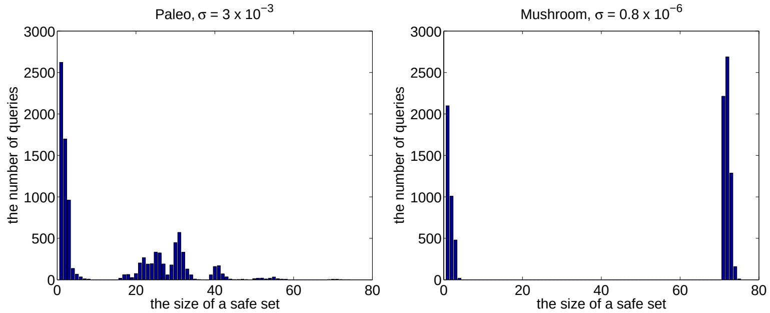

Fig. 3 Distributions of the sizes of safe sets. The left histogram is obtained from *Paleo* data by using $\sigma = 3 \times 10^{-3}$ as the threshold parameter for modified APriori. The right histogram is obtained from *Mushroom* data with $\sigma = 0.8 \times 10^{-8}$.

of the queries were labelled as Complex. We also defined a class All that contained all the queries.

As a measure of goodness for a frequency interval we considered the difference between the upper and the lower bound. Clearly $i_2(f) \subseteq i_1(f)$, so if we define a ratio $r(f) = \frac{\|i_2(f)\|}{\|i_1(f)\|}$, then it is always guaranteed that $0 \leq r(f) \leq 1$. Note that the ratio for the queries in Trivial is always 1.

The ratios were divided into appropriate bins. The results obtained from *Paleo* data are shown in the contingency table given in Tables 1 and 2 and the results for *Mushroom* data are given in Tables 3 and 4.

			$\sigma \times 10^{-3}$				
Class	$r \geq$	$r <$	3	3.25	3.5	3.75	4
Complex	0	0.2	1	0	0	0	0
	0.2	0.4	0	1	1	0	0
	0.4	0.6	15	11	10	5	4
	0.6	0.8	74	53	50	55	45
	0.8	1	238	173	124	99	68
	1		3289	1931	1353	1116	868
Trivial	1		6383	7831	8462	8725	9015

Table 1 Counts of queries obtained from *Paleo* data and classified according to the ratio $r(f)$, giving the relative tightness of the bounds from restricted safe sets compared to the trivial projections. A column represents a family of itemsets used as the constraints. The parameter σ is the threshold given to the modified APriori. The class Trivial contains the queries in which the projections were equal; Complex contains the remaining queries. For example, there were 15 complex queries having the ratios between $0.4 - 0.6$ in the first family.

By examining Tables 1 and 2 we conclude the following: If we conduct a random query of form f, then in $97\% - 99\%$ of the cases the frequency intervals are equal $i_1(f) = i_2(f)$. However, if we limit ourselves to the cases where the projections differ (the class Complex), then the frequency interval

	$\sigma \times 10^{-3}$				
Class	3	3.25	3.5	3.75	4
Complex	91.0%	89.0%	88.0%	87.5%	88.1%
All	96.7%	97.6%	98.1%	98.4%	98.8%

Table 2 Probability of $r(f) = 1$ among the complex queries and among all queries. The queries were obtained from *Paleo* data. A column represents a family of itemsets used as the constraints. The parameter σ is the threshold given to the modified APriori.

			$\sigma \times 10^{-6}$		
Class	$r \geq$	$r <$	0.8	0.9	1
Complex	0.0	0.2	46	38	42
	0.2	0.4	96	81	80
	0.4	0.6	302	261	260
	0.6	0.8	96	86	69
	0.8	1	168	118	109
	1		4738	4146	3993
Trivial	1		4554	5270	5447

Table 3 Counts of queries obtained from *Mushroom* data and classified according to the ratio $r(f)$, giving the relative tightness of the bounds from restricted safe sets compared to the trivial projections. A column represents a family of itemsets used as the constraints. The parameter σ is the threshold given to the modified APriori. The class Trivial contains the queries in which the projections were equal; Complex contains the remaining queries.

	$\sigma \times 10^{-6}$		
Class	0.8	0.9	1
Complex	87.0%	87.7%	87.7%
All	92.9%	94.2%	94.4%

Table 4 Probability of $r(f) = 1$ among the complex queries and among all queries. The queries were obtained from *Mushroom* data. A column represents a family of itemsets used as the constraints. The parameter σ is the threshold given to the modified APriori.

is equal only in about 90% of the cases. In addition, the probability of $i_1(f)$ being equal to $i_2(f)$ increases as the threshold σ grows.

The same observations apply to the results for *Mushroom* data (Tables 3 and 4): In $93\% - 94\%$ of the cases the frequency intervals are equal $i_1(f) = i_2(f)$, but if we consider only the cases where projections differ, then the percentage drops to 88%. The percentages are slightly smaller than those obtained from *Paleo* data and also there are relatively many queries whose ratios are very small.

The computational burden of a trivial query is equivalent for both trivial projection and restricted safe set. Hence, we examine complex queries in which there is an actual difference in the computational burden. The results suggest that in abt. 10% of the complex queries the restricted safe sets produced tighter interval.

9 Conclusions

We started our study by considering the following problem: Given a family $\mathcal{F}$ of itemsets, frequencies for $\mathcal{F}$, and a boolean formula find the bounds of the frequency of the formula. This can be solved by linear programming but the problem is that the program has an exponential number of variables. This can be remedied by neglecting the variables not occurring in the boolean formula and thus reducing the dimension. The downside is that the solution may change.

In the paper we defined a concept of safeness: Given an antimonotonic family $\mathcal{F}$ of itemsets a set C of attributes is safe if the projection to C does not change the solution of a query regardless of the query function and the given frequencies for $\mathcal{F}$. We characterised this concept by using graph theory. We also provided an efficient algorithm for finding the minimal safe set containing some given set.

We should point out that while our examples and experiments were focused on conjunctive queries, our theorems work with a query function of any shape

If the family of itemsets satisfies certain requirements, that is, it is triangulated and clique-safe, then we can obtain safe sets from junction trees. We also show that the factorisation obtained from a junction tree can be used to reduce the computational burden of the optimisation problem.

In addition, we provided a heuristic algorithm for finding restricted safe sets. The algorithm tries to construct a set of items such that the optimisation problem does not change for some *given* itemset frequencies.

We ask ourselves: In practice, should we use the safe sets rather than the trivial projections? The advantage is that the (restricted) safe sets always produce outcome at least as good as the trivial approach. The downside is the additional computational burden. Our tests indicate that if a user makes a random query then in abt. $93\% - 99\%$ of the cases the bounds are equal in both approaches. However, this comparison is unfair because there is a large number of queries where the projection sets are equal. To get the better picture we divide the queries into two classes TRIVIAL and COMPLEX, the first containing the queries such that the projections sets are equal, and the second containing the remaining queries. In the first class there is no improvement in the outcome *but* there is no additional computational burden (checking that the set is safe is cheap comparing to the linear programming). If a query was in COMPLEX, then in 10% of the cases projecting on restricted safe sets did produce more tight bounds.

Acknowledgements The author wishes to thank Heikki Mannila and Jouni Seppänen for their helpful comments.

References

1. Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil

Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 26–28 1993.

2. Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and Aino Inkeri Verkamo. Fast discovery of association rules. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI Press/The MIT Press, 1996.
3. Artur Bykowski, Jouni K. Seppänen, and Jaakko Hollmén. Model-independent bounding of the supports of Boolean formulae in binary data. In Pier Luca Lanzi and Rosa Meo, editors, *Database Support for Data Mining Applications: Discovering Knowledge with Inductive Queries*, LNCS 2682, pages 234–249. Springer Verlag, 2004.
4. Toon Calders. Computational complexity of itemset frequency satisfiability. In *Proceedings of the 23nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database System*, 2004.
5. Toon Calders and Bart Goethals. Mining all non-derivable frequent itemsets. In *Proceedings of the 6th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2002.
6. C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, May 1968.
7. J. Darroch and D. Ratchli. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480, 1972.
8. Mikael Forselius. Neogene of the old world database of fossil mammals (NOW). University of Helsinki, `http://www.helsinki.fi/science/now/`, 2005.
9. George Georgakopoulos, Dimitris Kavvadias, and Christos H. Papadimitriou. Probabilistic satisfiability. *Journal of Complexity*, 4(1):1–11, March 1988.
10. Bart Goethals and Mohammed Javeed Zaki, editors. *FIMI '03, Frequent Itemset Mining Implementations, Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, 19 December 2003, Melbourne, Florida, USA*, volume 90 of *CEUR Workshop Proceedings*, 2003.
11. Theodore Hailperin. Best possible inequalities for the probability of a logical function of events. *The American Mathematical Monthly*, 72(4):343–359, Apr. 1965.
12. Radim Jiroušek and Stanislav Přeušil. On the effective implementation of the iterative proportional fitting procedure. *Computational Statistics and Data Analysis*, 19:177–189, 1995.
13. Michael I. Jordan, editor. *Learning in graphical models*. MIT Press, 1999.
14. Thomas Lukasiewicz. Efficient global probabilistic deduction from taxonomic and probabilistic knowledge-bases over conjunctive events. In *Proceedings of the sixth international conference on Information and knowledge management*, pages 75–82, 1997.
15. Thomas Lukasiewicz. Probabilistic logic programming with conditional constraints. *ACM Transactions on Computational Logic (TOCL)*, 2(3):289–339, July 2001.
16. Heikki Mannila and Hannu Toivonen. Multiple uses of frequent sets and condensed representations (extended abstract). In *Knowledge Discovery and Data Mining*, pages 189–194, 1996.
17. Yossi Matias, Jeffrey Scott Vitter, and Min Wang. Wavelet-based histograms for selectivity estimation. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 448–459, 1998.
18. M. Muralikrishna and David DeWitt. Equi-depth histograms for estimating selectivity factors for multi-dimensional queries. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 28–36, 1988.
19. Nils Nilsson. Probbilistic logic. *Artificial Intelligence*.
20. Christos Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization Algorithms and Complexity*. Dover, 2nd edition, 1998.
21. Dmitry Pavlov, Heikki Mannila, and Padhraic Smyth. Beyond independence: Probabilistic models for query approximation on binary transaction data. *IEEE Transactions on Knowledge and Data Engineering*, 15(6):1409–1421, 2003.

22. Dmitry Pavlov and Padhraic Smyth. Probabilistic query models for transaction data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 164–173, 2001.

A Appendix

This section contains the proofs for the theorems presented in the paper.

A.1 Proof of Theorem 1

Let θ be any consistent frequencies for $\mathcal{F}$. Let $\mathcal{H} = \mathcal{F}_C$. To prove the theorem we will show that any distribution defined on items C and satisfying the frequencies θ_C can be extended to a distribution defined on the set A and satisfying the frequencies θ.

Let $W = A - C$. Partition W into connected blocks W_i such that $x, y \in W_i$ if and only if there is a path P from x to y such that $P \cap C = \emptyset$. Note that the items coming from the same W_i have the same frontier. Therefore, $\operatorname{front}(W_i, C)$ is well-defined. We denote $\operatorname{front}(W_i, C)$ by V_i.

Let p^{ME} be the maximum entropy distribution defined on the items A and satisfying θ. Note that there is no chord containing elements from W_i and from $C - V_i$ at the same time. This implies that we can write p^{ME} as

$$p^{ME}(X_A) = p^{ME}(X_C) \prod_i \frac{p^{ME}\left(X_{W_i}, X_{V_i}\right)}{p^{ME}\left(X_{V_i}\right)}.$$

Let p be any distribution defined on C and satisfying the frequencies θ_C. Note that $p^{ME}\left(X_{V_i}\right) = p\left(X_{V_i}\right)$, and hence we can extend p to the set A by defining

$$p(X_A) = p(X_C) \prod_i \frac{p^{ME}\left(X_{W_i}, X_{V_i}\right)}{p^{ME}\left(X_{V_i}\right)}.$$

To complete the proof we will need to prove that p satisfies the frequencies θ. Select any itemset $U \in \mathcal{F}$. There are two possible cases: Either $U \subseteq C$, which implies that $U \in \mathcal{H}$ and since p satisfies θ_C it follows that p also satisfies θ_U.

The other case is that U has elements outside C. Note that U can have elements in only one W_i, say, W_j. This in turn implies that U cannot have elements in $C - \operatorname{front}(W_j, C)$, that is, $U \subseteq W_j \cup V_i$. Note that $p^{ME}\left(X_{W_i}, X_{V_i}\right) = p\left(X_{W_i}, X_{V_i}\right)$. Since p^{ME} satisfies θ, p satisfies θ_U. This completes the theorem.

A.2 Proof of Theorem 2

Assume that we are given a family $\mathcal{F}$ of itemsets and a set C such that there exists $x \notin C$ such that $\operatorname{front}(x, C) \notin \mathcal{F}$. Select $Y \subseteq \operatorname{front}(x, C)$ to be some subset of the frontier such that $Y \notin \mathcal{F}$ and each proper subset of Y is contained in $\mathcal{F}$. We can also assume that paths from x to Y are of length 1. This is done by setting the intermediate attributes lying on the paths to be equivalent with x. We can also set the rest of the attributes to be equivalent with 0. Therefore, we can redefine $C = Y$, the underlying set of attributes to consist only of Y and x, and $\mathcal{F}$ to be

$$\mathcal{F} = \left\{Z \mid Z \subset C, Z \neq C\right\} \cup \left\{yx \mid y \in C\right\}.$$

Let $\theta = \{\theta_Z \mid Z \in \mathcal{F}\}$ be the frequencies for the itemset family $\mathcal{F}$ such that

$$\begin{array}{ll} \theta_Z = 0.5^{-|Z|} & \text{if } Z \subset C \\ \theta_Z = 0.5 & \text{if } Z = x \\ \theta_Z = c & \text{if } Z = xy \text{ for } y \in C, \end{array} \tag{4}$$

where c is a constant (to be determined later).

Define n to be the number of elements in C. Let k be the number of ones in the random bit vector X_C. Let us now consider the following three distributions defined on C:

$$\begin{aligned} p_1(X_C) &= \begin{cases} 2^{-n+1} & , n-k \text{ is even} \\ 0 & , n-k \text{ is odd} \end{cases} \\ p_2(X_C) &= 2^{-n} \\ p_3(X_C) &= \begin{cases} 2^{-n+1} & , n-k \text{ is odd} \\ 0 & , n-k \text{ is even} \end{cases} . \end{aligned}$$

Note that all three distributions satisfy the first condition in Eq. 4. Note also that $p_i(X_C)$ depends only on the number of ones in X_C. We will slightly abuse the notation and denote $p_i(k) = p_i(X_C)$, where X_C is a random vector having k ones.

Assume that we have extended $p_i(X_C)$ to $p_i(X_C, X_x)$ satisfying θ. We can assume that $p_i(X_C, X_x)$ depends only on the number of ones in X_C and the value of X_x. Define $c_i(n,k) = p_i(X_C, X_x = 1)$, where X_C is a random vector having k ones. Note that

$$0.5 = p_i(X_x = 1) = \sum_{k=0}^{n} \binom{n}{k} c_i(n,k).$$

If we select any attribute $z \in C$, then

$$c = p_i(X_z = 1, X_x = 1) = \sum_{k=1}^{n} \binom{n-1}{k-1} c_i(n,k).$$

If we now consider the conditions given in Eq. 4 and require that $p_i(X_x = 1) = \theta_x = 0.5$ and also require that $p_i(X_z = 1, X_x = 1) = c$ is the largest possible, then we get the following three optimisation problems:

$$\begin{array}{lr} \text{Problem P}_i: & \\ \text{Maximise} & c_i(n) = \sum_{k=1}^{n} \binom{n-1}{k-1} c_i(n,k) \\ \text{subject to} & c_i(n,k) \geq 0 \\ & c_i(n,k) \leq p_i(k) \\ & 0.5 = \sum_{k=0}^{n} \binom{n}{k} c_i(n,k) \end{array} \tag{5}$$

If we can show that the statement

$$c_1(n) = c_2(n) = c_3(n)$$

is false, then by setting $c = \max(c_1(n), c_2(n), c_3(n))$ in Eq. 4 we obtain such frequencies that at least one of the distributions p_i cannot be extended to x. We will prove our claim by assuming otherwise and showing that the assumption leads to a contradiction.

Note that $\binom{n-1}{k-1}/\binom{n}{k} = k/n$. This implies that the maximal solution $c_2(n)$ has the *unique* form

$$c_2(n,k) = \begin{cases} 2^{-n} & , k > \frac{n}{2} \\ 2^{-n-1} & , k = \frac{n}{2} \text{ and } n \text{ is even} \\ 0 & , \text{otherwise.} \end{cases} \tag{6}$$

Define series $b(n,k) = \frac{1}{2}\left(c_1(n,k) + c_3(n,k)\right)$. Note that $b(n,k)$ is a feasible solution for Problem P_2 in Eq. 5. Moreover, since we assume that $c_2(n) = c_1(n) = c_3(n)$, it follows that $b(n,k)$ produces the optimal solution $c_2(n)$. Therefore, $b(n,k) = c_2(n,k)$. This implies that $c_1(n,k)$ and $c_3(n,k)$ have the forms

$$c_1(n,k) = \begin{cases} 2c_2(n,k) & , n-k \text{ is even} \\ 0 & , n-k \text{ is odd} \end{cases} \tag{7}$$

$$c_3(n,k) = \begin{cases} 2c_2(n,k) & , n-k \text{ is odd} \\ 0 & , n-k \text{ is even} \end{cases} . \tag{8}$$

Assume now that n is odd. The conditions of Problems P_1 and P_3 imply that

$$\sum_{k=0}^{n} \binom{n}{k} c_1(n,k) = 0.5 = \sum_{k=0}^{n} \binom{n}{k} c_3(n,k).$$

By applying Eqs. 6– 8 to this equation we obtain, depending on n, either the identity

$$\binom{n}{n} + \binom{n}{n-2} + \ldots + \binom{n}{\frac{n+1}{2}} = \binom{n}{n-1} + \binom{n}{n-3} + \ldots + \binom{n}{\frac{n+3}{2}}$$

or

$$\binom{n}{n} + \binom{n}{n-2} + \ldots + \binom{n}{\frac{n+3}{2}} = \binom{n}{n-1} + \binom{n}{n-3} + \ldots + \binom{n}{\frac{n+1}{2}}.$$

Both of these identities are false since the series having the term $\binom{n}{\frac{n+1}{2}}$ is always larger. This proves our claim for the cases where n is odd.

Assume now that n is even. The assumption $c_1(n) = c_3(n)$ together with Eqs. 6–8 implies the identity

$$\binom{n-1}{n-1} + \binom{n-1}{n-3} + \ldots + \frac{1}{2}\binom{n-1}{\frac{n}{2}-1} = \binom{n-1}{n-2} + \binom{n-1}{n-4} + \ldots + \binom{n-1}{\frac{n}{2}}.$$

We apply the identity

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \tag{9}$$

to this equation and cancel out the equal terms from both sides. This gives us the identity

$$\frac{1}{2}\binom{n-1}{\frac{n}{2}-1} = \binom{n-2}{\frac{n}{2}-1}.$$

By applying again Eq. 9 we obtain

$$\binom{n-2}{\frac{n}{2}-2} = \binom{n-2}{\frac{n}{2}-1}.$$

This is true for no n and thus we have proved our claim.

A.3 Proof of Theorem 5

Denote by $\mathcal{E}\left(p\right)$ the entropy of a distribution p. We know that $\mathcal{E}\left(q^{ME}\right) \geq \mathcal{E}\left(p_C^{ME}\right)$. Assume now that q is a distribution satisfying the frequencies θ_C. Let us extend q as we did in the proof of Theorem 1:

$$p(X_A) = q(X_C) \prod_i \frac{p^{ME}\left(X_{W_i}, X_{V_i}\right)}{p^{ME}\left(X_{V_i}\right)}.$$

The entropy of this distribution is of the form $\mathcal{E}\left(p\right) = \mathcal{E}\left(q\right) + c$, where

$$c = \sum_i \mathcal{E}\left(p_{W_i \cup V_i}^{ME}\right) - \mathcal{E}\left(p_{V_i}^{ME}\right)$$

is a constant not depending on q. This characterisation is valid because $p_{V_i}^{ME} = q_{V_i}$. If we let $q = q^{ME}$, it follows that

$$\mathcal{E}\left(p^{ME}\right) \geq \mathcal{E}\left(p\right) = \mathcal{E}\left(q^{ME}\right) + c \geq \mathcal{E}\left(p_C^{ME}\right) + c.$$

If we now let $q = p_C^{ME}$, it follows that $p = p^{ME}$ and this implies that $\mathcal{E}\left(p^{ME}\right) = \mathcal{E}\left(p_C^{ME}\right) + c$. Thus $\mathcal{E}\left(q^{ME}\right) = \mathcal{E}\left(p_C^{ME}\right)$. The distribution maximising entropy is unique, thus $p_C^{ME} = q^{ME}$.

A.4 Proof of Theorem 6

Assume that there is $x \in Z$ such that $x \notin Y$. Let $U_x = \{u_1, \ldots, u_L\}$ be as it is defined in Algorithm 1. Let P_i be the shortest path from x to u_i and define v_i to be the first item on P_i belonging to Y. There are two possible cases: Either $v_i = u_i$ which implies that $u_i \in \operatorname{front}\left(x, Y\right)$, or u_i is blocked by some other element in Y. If $U_x \subseteq \operatorname{front}\left(x, Y\right)$, then the safeness condition is violated. Therefore, there exists u_j such that $v_j \neq u_j$.

We will prove that v_j outranks x, that is, $\operatorname{rank}\left(v_j \mid C\right) > \operatorname{rank}\left(x \mid C\right)$. It is easy to see that it is sufficient to prove that $\operatorname{rank}\left(v_j \mid U_x\right) > \operatorname{rank}\left(x \mid U_x\right)$. In order to do this note that $\{v_1, \ldots, v_L\} \subseteq \operatorname{front}\left(x, Y\right) \in \mathcal{F}$. Therefore, because of the antimonotonic property of $\mathcal{F}$, there is an edge from v_j to each v_i. This implies that there is a path R_i from v_j to u_i such that $|R_i| \leq |P_i|$, that is, the length of R_i is smaller or equal than the length of P_i. Also note, that since v_j lies on P_j, there exists a path R_j from v_j to u_j such that $|R_j| < |P_j|$. This implies that $\operatorname{rank}\left(v_j \mid U_x\right) > \operatorname{rank}\left(x \mid U_x\right)$.

Also, note that $U_x \subset N\left(v_j \mid r\right)$, where r is the search radius defined in Algorithm 1. This implies that v_j is discovered during the search phase, that is, v_j is one of the violating nodes.

To complete the proof we need to show that v_j is a neighbour of C. Since x is a neighbour of C, there is u_k such that there is an edge between x and u_k. This implies that $v_k = u_k$. Since there is an edge between v_j and v_k, it follows that v_j is neighbour of C.

A.5 Proof of Theorem 7

Let a be some item belonging to some inner clique Q but not belonging in any inner separator. The clique Q is unique and the only reachable items of C from a are the inner separators incident to Q. Since Q is a clique, it follows from the clique-safeness assumption that the frontier of a is included in $\mathcal{F}$.

Let now a be any item that is not included in any inner clique. There exists a unique inner clique Q such that all the paths from a to C go through this clique. This implies that the frontier of a is again the inner separators incident to Q.

A.6 Proof of Theorem 8

We will prove that if we have an item a coming from some inner separator and not included in the minimal safe set, then we can alter the junction tree such that the item a is no longer included in the inner separators. For the sake of clarity, we illustrate an example of the modification process in Figure 4.

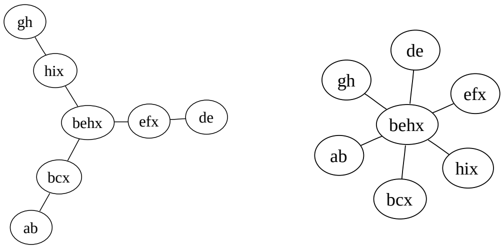

Fig. 4 Two equivalent junction trees. Our goal is to find the minimal safe set for $B = \{a, d, g\}$. The left junction tree is before the modification and the right is after the modification. We see that the attribute x is not included in the inner separators in the right tree. The sets appearing in the proof are as follows: The minimal safe set C is $adgbeh$. I consists of 3 separators bx, ex, and hx. The other separators belong to J. V consists of 4 cliques bcx, efx, hix, and $behx$. The clique Q is $behx$.

Let G be the dependency graph and T the current junction tree. Let C be the minimal safe set containing B and let $a \notin C$ be an item coming from some inner separator. Let us consider paths (in G) from a to its frontier. For the sake of clarity, we prove only the case where the paths from a to C are of length 1. The proof for the general case is similar.

Let I be the collection of inner separators containing a. Let V be the collection of (inner) cliques incident to the inner separators included in I. The pair (V, I) defines a subtree of T. Let J be the set of inner separators incident to some clique in V but not included in I. Note that each item coming from the inner separators included in J must be included in C because otherwise we have violated the assumption that the paths from a to its frontier are of length 1.

The frontier of a consists of the items of the inner separators in J and of possibly some items from the set B. By the assumption the frontier is in $\mathcal{F}$ and thus it is fully connected. It follows that there is a clique Q containing the frontier. If $Q \notin V$, a clique from V closest to Q also contains the frontier. Hence we can assume $Q \in V$.

Select a separator $E \in J$. Let $U \notin V$ be the clique incident to E. We modify the tree by cutting the edge E and reattaching U to Q. The procedure is performed to each separator in J. The obtained tree satisfies the running intersection property since Q contains the items coming from each inner separators included in J. If the frontier contained any items included in B, then Q contains these items. It is easy to see that each clique in V, except for the clique Q, becomes outer. Therefore, a is no longer included in any inner separator.

A.7 Proof of Theorem 9

Let $\hat{p}$ be the optimal distribution. Then by marginalising we can obtain $\hat{p}_i$, and $\hat{q}_j$ which produce the same solution for the reduced problem.

To prove the other direction let $\hat{p}_i$, and $\hat{q}_j$ be the optimal distributions for the reduced problem. Since the running intersection property holds, we can define the joint distribution $\hat{p}$ by $\hat{p} = \prod_i \hat{p}_i / \prod_j \hat{q}_j$. It is straightforward to see that $\hat{p}$ satisfies the frequencies. This proves the statement.