Interestingness of Frequent Itemsets Using Bayesian Networks as Background Knowledge

Szymon Jaroszewicz
sj@cs.umb.edu
Dept. of Comp. Science
Technical University of Szczecin
ul. Żołnierska 49
71-210 Szczecin, Poland

Dan A. Simovici
dsim@cs.umb.edu
Dept. of Comp. Science
University of Massachusetts at Boston
100 Morrissey Blvd.
02125 Boston, U.S.A.

ABSTRACT

The paper presents a method for pruning frequent itemsets based on background knowledge represented by a Bayesian network. The interestingness of an itemset is defined as the absolute difference between its support estimated from data and from the Bayesian network. Efficient algorithms are presented for finding interestingness of a collection of frequent itemsets, and for finding all attribute sets with a given minimum interestingness. Practical usefulness of the algorithms and their efficiency have been verified experimentally.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*data mining*

Keywords

association rule, frequent itemset, background knowledge, interestingness, Bayesian network

1. INTRODUCTION

Finding frequent itemsets and association rules in database tables has been an active research area in recent years. Unfortunately, the practical usefulness of the approach is limited by huge number of patterns usually discovered. For larger databases many thousands of association rules may be produced when minimum support is low. This creates a secondary data mining problem: after mining the data, we are now compelled to mine the discovered patterns. The problem has been addressed in literature mainly in the context of association rules, where the two main approaches are sorting rules based on some interestingness measure, and pruning aiming at removing redundant rules.

Full review of such methods is beyond the scope of this paper. Overviews of interestingness measures can be found for example in [3, 13, 11, 32], some of the papers on rule pruning are [30, 31, 7, 14, 28, 16, 17, 33].

Many interestingness measures are based on the divergence between true probability distributions and distributions obtained under the independence assumption. Pruning methods are usually based on comparing the confidence of a rule to the confidence of rules related to it.

The main drawback of those methods is that they tend to generate rules that are either obvious or have already been known by the user. This is to be expected, since the most striking patterns which those methods select can also easily be discovered using traditional methods or are known directly from experience.

We believe that the proper way to address the problem is to include users background knowledge in the process. The patterns which diverge the most from that background knowledge are deemed most interesting. Discovered patterns can later be applied to improve the background knowledge itself.

Many approaches to using background knowledge in machine learning are focused on using background knowledge to speed up the hypothesis discovery process and not on discovering interesting patterns. Those methods often assume strict logical relationships, not probabilistic ones. Examples are knowledge based neural networks (KBANNs) and uses of background knowledge in Inductive Logic Programming. See Chapter 12 in [20] for an overview of those methods and a list of further references.

Tuzhilin et. al. [23, 22, 29] worked on applying background knowledge to finding interesting rules. In [29, 22] interestingness measures are presented, which take into account prior beliefs; in another paper [23], the authors present an algorithm for selecting a minimum set of interesting rules given background knowledge. The methods used in those papers are local, that is, they don't use a full joint probability of the data. Instead, interestingness of a rule is evaluated using rules in the background knowledge with the same consequent. If no such knowledge is present for a given rule, the rule is considered uninteresting. This makes it impossible to take into account transitivity. Indeed, in the presence of the background knowledge represented by the rules $A \Rightarrow B$ and $B \Rightarrow C$, the rule $A \Rightarrow C$ is uninteresting. However, this cannot be discovered locally. See [25] for a detailed discussion of advantages of global versus local methods. Some more comparisons can be found in [18].

KDD'04, August 22–25, 2004, Seattle, Washington, USA.

In this paper we present a method of finding interesting patterns using background knowledge represented by a Bayesian network. The main advantage of Bayesian networks is that they concisely represent full joint probability distributions, and allow for practically feasible probabilistic inference from that distribution [25, 15]. Other advantages include the ability to represent causal relationships, easy to understand graphical structure, as well as wide availability of modelling tools. Bayesian networks are also easy to modify by adding or deleting edges.

We opt to compute interestingness of frequent itemsets instead of association rules, agreeing with [7] that directions of dependence should be decided by the user based on her experience and not suggested by interestingness measures. Our approach works by estimating supports of itemsets from Bayesian networks and comparing thus estimated supports with the data. Itemsets with strongly diverging supports are considered interesting.

Further definitions of interestingness exploiting Bayesian network's structure are presented, as well as efficient methods for computing interestingness of large numbers of itemsets and for finding all attribute sets with given minimum interestingness.

There are some analogies between mining emerging patterns [6] and our approach, the main differences being that in our case a Bayesian network is used instead of a second dataset, and that we use a different measure for comparing supports. Due to those differences our problem requires a different approach and a different set of algorithms.

2. DEFINITIONS AND NOTATION

Database attributes will be denoted with uppercase letters $A, B, C, \ldots$, domain of an attribute A will be denoted by $\mathrm{Dom}(A)$. In this paper we are only concerned with categorical attributes, that is attributes with finite domains.

Sets of attributes will be denoted with uppercase letters $I, J, \ldots$. We often use database notation for representing sets of attributes, i.e. $I = A_1 A_2 \ldots A_k$ instead of the set theoretical notation $\{A_1, A_2, \ldots, A_k\}$. Domain of an attribute set $I = A_1 A_2 \ldots A_k$ is defined as

$$\mathrm{Dom}(I) = \mathrm{Dom}(A_1) \times \mathrm{Dom}(A_2) \times \ldots \times \mathrm{Dom}(A_k).$$

Values from domains of attributes and attribute sets are denoted with corresponding lowercase boldface letters, e.g. $\mathbf{i} \in \mathrm{Dom}(I)$.

Let P_I denote a joint probability distribution of the attribute set I. Similarly let $P_{I|J}$ be a distribution of I conditioned on J. When used in arithmetic operations such distributions will be treated as functions of attributes in I and $I \cup J$ respectively, with values in the interval $[0, 1]$. For example $P_I(\mathbf{i})$ denotes the probability that $I = \mathbf{i}$.

Let P_I be a probability distribution, and let $J \subset I$. Denote by $P_I^{\downarrow J}$ the marginalization of P_I onto J, that is

$$P_I^{\downarrow J} = \sum_{I \setminus J} P_I, \tag{1}$$

where the summation is over the domains of all variables from $I \setminus J$.

Probability distributions estimated from data will be denoted by adding a hat symbol, e.g. $\hat{P}_I$.

An *itemset* is a pair $(I, \mathbf{i})$, where I is an attribute set and $\mathbf{i} \in \mathrm{Dom}(I)$. The support of an itemset $(I, \mathbf{i})$ is defined as

$$\mathrm{supp}(I, \mathbf{i}) = \hat{P}_I(\mathbf{i}),$$

where the probability is estimated from some dataset. An itemset is called *frequent* if its support is greater than or equal to some user defined threshold minsupp. Finding all frequent itemsets in a given database table is a well known datamining problem [1].

A *Bayesian network* BN over a set of attributes $H = A_1 \ldots A_n$ is a directed acyclic graph $BN = (V, E)$ with the set of vertices $V = \{V_{A_1}, \ldots, V_{A_n}\}$ corresponding to attributes of H, and a set of edges $E \subset V \times V$, where each vertex V_{A_i} has associated a conditional probability distribution $P_{A_i|\mathsf{par}_i}$, where $\mathsf{par}_i = \{A_j : (V_{A_j}, V_{A_i}) \in E\}$ is the set of attributes corresponding to parents of V_{A_i} in G. See [25, 15] for a detailed discussion of Bayesian networks.

A Bayesian network BN over H uniquely defines a joint probability distribution

$$P_H^{BN} = \prod_{i=1}^{n} P_{A_i|\mathsf{par}_i}$$

of H. For $I \subseteq H$ the distribution over I marginalized from P_H^{BN} will be denoted by P_I^{BN}

$$P_I^{BN} = \left(P_H^{BN}\right)^{\downarrow I}.$$

3. INTERESTINGNESS OF AN ATTRIBUTE SET WITH RESPECT TO A BAYESIAN NETWORK

Let us first define the *support of an itemset* $(I, \mathbf{i})$ *in a Bayesian network* BN as

$$\mathrm{supp}_{BN}(I, \mathbf{i}) = P_I^{BN}(\mathbf{i}).$$

Let BN be a Bayesian network over an attribute set H, and let $(I, \mathbf{i})$ be an itemset such that $I \subseteq H$. The *interestingness* of the itemset $(I, \mathbf{i})$ with respect to BN is defined as

$$\mathcal{I}_{BN}(I, \mathbf{i}) = |\mathrm{supp}(I, \mathbf{i}) - \mathrm{supp}_{BN}(I, \mathbf{i})|$$

that is, the absolute difference between the support of the itemset estimated from data, and the estimate of this support made from the Bayesian network BN. In the remaining part of the paper we assume that interestingness is always computed with respect to a Bayesian network BN and the subscript is omitted.

An itemset is *ϵ-interesting* if its interestingness is greater than or equal to some user specified threshold ϵ.

A frequent interesting itemset represents a frequently occurring (due to minimum support requirement) pattern in the database whose probability is significantly different from what it is believed to be based on the Bayesian network model.

An alternative would be to use $\mathrm{supp}(I, \mathbf{i})/\mathrm{supp}_{BN}(I, \mathbf{i})$ as the measure of interestingness [6]. We decided to use absolute difference instead of a quotient since we found it to be more robust, especially when both supports are small.

One could think of applying our approach to association rules with the difference in confidences as a measure of interestingness but, as mentioned in the Introduction, we think that patterns which do not suggest a direction of influence are more appropriate.

Since in Bayesian networks dependencies are modelled using attributes not itemsets, it will often be easier to talk about interesting attribute sets, especially when the discovered interesting patterns are to be used to update the background knowledge.

DEFINITION 3.1. *Let I be an attribute set. The* interestingness *of I is defined as*

$$\mathcal{I}(I) = \max_{\mathbf{i} \in \mathrm{Dom}(I)} \mathcal{I}(I, \mathbf{i}), \tag{2}$$

analogously, I is ϵ-interesting *if $\mathcal{I}(I) \geq \epsilon$.* □

An alternative approach would be to use generalizations of Bayesian networks allowing dependencies to vary for different values of attributes, see [27], and deal with itemset interestingness directly.

3.1 Extensions to the Definition of Interestingness

Even though applying the above definition and sorting attribute sets on their interestingness works well in practice, there might still be a large number of patterns retained, especially if the background knowledge is not well developed and large number of attribute sets have high interestingness values. This motivates the following two definitions.

DEFINITION 3.2. *An attribute set I is* hierarchically ϵ-interesting *if it is ϵ-interesting and none of its proper subsets is ϵ-interesting.* □

The idea is to prevent large attribute sets from becoming interesting when the true cause of them being interesting lies in their subsets.

There is also another problem with Definition 3.1. Consider a Bayesian network

$$A \rightarrow B$$

where nodes A and B have respective probability distributions P_A and $P_{B|A}$ attached. Suppose also that A is ϵ-interesting. In this case even if $P_{B|A}$ is the same as $\hat{P}_{B|A}$, attribute sets B and AB may be considered ϵ-interesting. Below we present a definition of interestingness aiming at preventing such situations.

A vertex V is an *ancestor* of a vertex W in a directed graph G if there is a directed path from V to W in G. The set of ancestors of a vertex V in a graph G is denoted by $\mathsf{anc}(V)$. Moreover, let us denote by $\mathsf{anc}(I)$ the set of all ancestor attributes in BN of an attribute set I. More formally:

$$\mathsf{anc}(I) = \{A_i \notin I : V_{A_i} \in \mathsf{anc}(V_{A_j}) \text{ in BN, for some } A_j \in I\}.$$

DEFINITION 3.3. *An attribute set I is* topologically ϵ-interesting *if it is ϵ-interesting, and there is no attribute set J such that*

1. $J \subseteq \mathsf{anc}(I) \cup I$, *and*
2. $I \not\subseteq J$, *and*
3. J *is ϵ-interesting.*

□

The intention here is to prevent interesting attribute sets from causing all their successors in the Bayesian network (and the supersets of their successors) to become interesting in a cascading fashion.

To see why condition 2 is necessary consider a Bayesian network

$$A \leftarrow X \rightarrow B$$

Suppose that there is a dependency between A and B in data which makes AB ϵ-interesting. Now however ABX may also become interesting, (even if $P_{A|X}$ and $P_{B|X}$ are correct in the network) and cause AB to be pruned. Condition 2 prevents AB from being pruned and ABX from becoming interesting.

Notice that topological interestingness is stricter than hierarchical interestingness. Indeed if $J \subset I$ is ϵ-interesting, then it satisfies all the above conditions, and makes I not topologically ϵ-interesting.

4. ALGORITHMS FOR FINDING INTERESTING ITEMSETS AND ATTRIBUTE SETS

In this section we present algorithms using the definition of interestingness introduced in the previous section to select interesting itemsets or attribute sets. We begin by describing a procedure for computing marginal distributions for a large collection of attribute sets from a Bayesian network.

4.1 Computing a Large Number of Marginal Distributions from a Bayesian Network

Computing the interestingness of a large number of frequent itemsets requires the computation of a large number of marginal distributions from a Bayesian network. The problem has been addressed in literature mainly in the context of finding marginals for every attribute [25, 15], while here we have to find marginals for multiple, overlapping sets of attributes. The approach taken in this paper is outlined below.

The problem of computing marginal distributions from a Bayesian network is known to be NP-hard, nevertheless in most cases the network structure can be exploited to speed up the computations.

Here we use exact methods for computing the marginals. Approximate methods like Gibbs sampling are an interesting topic for future work.

Best known approaches to exact marginalizations are join trees [12] and bucket elimination [5]. We chose bucket elimination method which is easier to implement and according to [5] as efficient as join tree based methods. Also, join trees are mainly useful for computing marginals for single attributes, and not for sets of attributes.

The bucket elimination method, which is based on the distributive law, proceeds by first choosing a variable ordering and then applying distributive law repeatedly to simplify the summation. For example suppose that a joint distribution of a Bayesian network over $H = ABC$ is expressed as

$$P^{BN}_{ABC} = P_A \cdot P_{B|A} \cdot P_{C|A},$$

and we want to find P^{BN}_A. We need to compute the sum

$$\sum_B \sum_C P_A \cdot P_{B|A} \cdot P_{C|A}$$

which can be rewritten as

$$P_A \cdot \left(\sum_{b \in \mathrm{Dom}(B)} P_{B|A} \right) \cdot \left(\sum_{c \in \mathrm{Dom}(C)} P_{C|A} \right).$$

Assuming that domains of all attributes have size 3, computing the first sum directly requires 12 additions and 18 multiplications, while the second sum requires only 4 additions and 6 multiplications.

The expression is interpreted as a tree of *buckets*, each bucket is either a single probability distribution, or a sum over a single attribute taken over a product of its child buckets in the tree. In the example above a special root bucket without summation could be introduced for completeness.

In most cases the method significantly reduces the time complexity of the problem. An important problem is choosing the right variable ordering. Unfortunately that problem is itself NP-hard. We thus adopt a heuristic which orders variables according to the decreasing number of factors in the product depending on each variable. A detailed discussion of the method can be found in [5].

Although bucket elimination can be used to obtain supports of itemsets directly (i.e. $P_I(\mathbf{i})$), we use it to obtain complete marginal distributions. This way we can directly apply marginalization to obtain distributions for subsets of I (see below).

Since bucket elimination is performed repeatedly we use memoization to speed it up, as suggested in [21]. We remember each partial sum and reuse it if possible. In the example above $\sum_{b\in \mathrm{Dom}(B)} P_{B|A}$, $\sum_{c\in \mathrm{Dom}(C)} P_{C|A}$, and the computed P_A^{BN} would have been remembered.

Another method of obtaining a marginal distribution P_I is marginalizing it from P_J where $I \subset J$ using Equation (1), provided that P_J is already known. If $|J \setminus I|$ is small, this procedure is almost always more efficient than bucket elimination, so whenever some P_I is computed by bucket elimination, distributions of all subsets of I are computed using Equation (1).

DEFINITION 4.1. *Let $\mathcal{C}$ be a collection of attribute sets. The* positive border *of $\mathcal{C}$ [19], denoted by $Bd^+(\mathcal{C})$, is the collection of those sets from $\mathcal{C}$ which have no proper superset in $\mathcal{C}$:*

$$Bd^+(\mathcal{C}) = \{I \in \mathcal{C} : \textit{there is no } J \in \mathcal{C} \textit{ such that } I \subset J\}.$$

□

It is clear from the discussion above that we only need to use bucket elimination to compute distributions of itemsets in the positive border. We are going to go further than this; we will use bucket elimination to obtain supersets of sets in the positive border, and then use Equation (1) to obtain marginals even for sets in the positive border. Experiments show that this approach can give substantial savings, especially when many overlapping attribute sets from the positive border can be covered by a single set only slightly larger then the covered ones.

The algorithm for selecting the marginal distribution to compute is motivated by the algorithm from [9] for computing views that should be materialized for OLAP query processing. Bucket elimination corresponds to creating a materialized view, and marginalizing thus obtained distribution to answering OLAP queries.

We first need to define costs of marginalization and bucket elimination. In our case the cost is defined as the total number of additions and multiplications used to compute the marginal distribution.

The cost of marginalizing P_J from P_I, $J \subseteq I$ using Equation (1) is

$$cost(P_I^{\downarrow J}) = |\mathrm{Dom}(J)| \cdot (|\mathrm{Dom}(I \setminus J)| - 1)\,.$$

It follows from the fact that each value of $P_I^{\downarrow J}$ requires adding $|\mathrm{Dom}(I \setminus J)|$ values from P_I.

The cost of bucket elimination can be computed cheaply without actually executing the procedure. Each bucket is either an explicitly given probability distribution, or computes a sum over a single variable of a product of functions (computed in buckets contained in it) explicitly represented as multidimensional tables, see [5] for details. If the bucket is an explicitly given probability distribution, the cost is of course 0.

Consider now a bucket b containing child buckets $b_1, \ldots, b_n$ yielding functions $f_1, \ldots, f_n$ respectively. Let $\mathrm{Var}(f_i)$ the set of attributes on which f_i depends.

Let $f = f_1 \cdot f_2 \cdots f_n$ denote the product of all factors in b. We have $\mathrm{Var}(f) = \cup_{i=1}^n \mathrm{Var}(f_i)$, and since each value of f requires $n-1$ multiplications, computing f requires $|\mathrm{Dom}(\mathrm{Var}(f))| \cdot (n-1)$ multiplications. Let A_b be the attribute over which summation in b takes place. Computing the sum will require $|\mathrm{Dom}(\mathrm{Var}(f) \setminus \{A_b\})| \cdot (|\mathrm{Dom}(A_b)| - 1)$ additions.

So the total cost of computing the function in bucket b (including costs of computing its children) is thus

$$\begin{aligned} cost(b) &= \sum_{i=1}^{n} cost(b_i) + |\mathrm{Dom}(\mathrm{Var}(f))| \cdot (n-1) \\ &\quad + |\mathrm{Dom}(\mathrm{Var}(f) \setminus \{A_b\})| \cdot (|\mathrm{Dom}(A_b)| - 1). \end{aligned}$$

The cost of computing P_I^{BN} through bucket elimination, denoted $cost_{BE}(P_I^{BN})$, is the cost of the root bucket of the summation used to compute P_I^{BN}.

Let $\mathcal{C}$ be a collection of attribute sets. The *gain* of using bucket elimination to find P_I^{BN} for some I while computing interestingness of attribute sets from $\mathcal{C}$ can be expressed as:

$$\begin{aligned} gain(I) &= -cost_{BE}(P_I^{BN}) + \\ &\quad \sum_{J \in Bd^+(\mathcal{C}), J \subset I} \left[cost_{BE}(P_J^{BN}) - cost(P_I^{BN\downarrow J})\right]. \end{aligned}$$

An attribute set to which bucket elimination will be applied is found using a greedy procedure by adding in each iteration the attribute giving the highest increase of *gain*. The complete algorithm is presented in Figure 1.

4.2 Computing The Interestingness of a Collection of Itemsets

First we present an algorithm for computing interestingness of all itemsets in a given collection. Its a simple application of the algorithm in Figure 1. It is useful if we have a collection of itemsets (e.g. frequent itemsets) and want to select those which are the most interesting. The algorithm is given below

Input: collection of itemsets $\mathcal{K}$, supports of all itemsets in $\mathcal{K}$, Bayesian network BN
Output: interestingness of all itemsets in $\mathcal{K}$.

1. $\mathcal{C} \leftarrow \{I : (I, \mathbf{i}) \in \mathcal{K} \text{ for some } \mathbf{i} \in \mathrm{Dom}(I)\}$
2. compute P_I^{BN} for all $I \in \mathcal{C}$ using algorithm in Figure 1
3. compute interestingness of all itemsets in $\mathcal{K}$ using distributions computed in step 2

□

Input: collection of attribute sets $\mathcal{C}$, Bayesian network BN

Output: distributions P_I^{BN} for all $I \in \mathcal{C}$

1. $\mathcal{S} \leftarrow Bd^+(\mathcal{C})$
2. while $\mathcal{S} \neq \emptyset$:
3. $I \leftarrow$ an attribute set from $\mathcal{S}$.
4. for A in $H \setminus I$:
5. compute $gain(I \cup \{A\})$
6. pick $A^\star$ for which the gain in Step 5 was maximal
7. if $gain(I \cup \{A^\star\}) > gain(I)$:
8. $I \leftarrow I \cup \{A^\star\}$
9. goto 4
10. compute P_I^{BN} from BN using bucket elimination
11. compute $P_I^{BN \downarrow J}$ for all $J \in \mathcal{S}, J \subset I$ using Equation (1)
12. remove from $\mathcal{S}$ all attribute sets included in I
13. compute P_J^{BN} for all $J \in \mathcal{C} \setminus Bd^+(\mathcal{C})$ using Equation (1)

Figure 1: Algorithm for computing a large number of marginal distributions from a Bayesian network.

4.3 Finding All Attribute Sets With Given Minimum Interestingness

In this section we will present an algorithm for finding all attribute sets with interestingness greater than or equal to a specified threshold ϵ given a dataset and a Bayesian network BN.

Let us first make an observation:

OBSERVATION 4.2. *If an itemset $(I, \mathbf{i})$ has interestingness greater than or equal to ϵ with respect to a Bayesian network BN then its support must be greater than or equal to ϵ in either the data or in BN. Moreover if an attribute set is ϵ-interesting, by definition 3.1, at least one of its itemsets must be ϵ-interesting.*

It follows that if an attribute set is ϵ-interesting, then one of its itemsets must be frequent, with minimum support ϵ, either in the data or in the Bayesian network.

The algorithm works in two stages. First all frequent itemsets with minimum support ϵ are found in the dataset and their interestingness is computed. The first stage might have missed itemsets which are ϵ-interesting but don't have sufficient support in the data.

In the second stage all itemsets frequent in the Bayesian network are found, and their supports in the data are computed using an extra database scan.

To find all itemsets frequent in the Bayesian network we use the Apriori algorithm [1] with a modified support counting part, which we call `AprioriBN`. The sketch of the algorithm is shown in Figure 2, except for step 3 it is identical to the original algorithm.

Input: Bayesian network BN, minimum support minsupp.

Output: itemsets whose support in BN is $\geq$ minsupp

1. $K \leftarrow 1$
2. $Cand \leftarrow \{(I, \mathbf{i}) : |I| = 1\}$
3. compute $\text{supp}_{BN}(I, \mathbf{i})$ for all $(I, \mathbf{i}) \in Cand$ using the algorithm in Figure 1
4. $Freq \leftarrow \{(I, \mathbf{i}) \in Cand : \text{supp}_{BN}(I, \mathbf{i}) \geq \text{minsupp}\}$
5. $Cand \leftarrow$ generate new candidates from $Freq$
6. remove itemsets with infrequent subsets from $Cand$
7. goto 3

Figure 2: The `AprioriBN` algorithm

We now have all the elements needed to present the algorithm for finding all ϵ-interesting attribute sets, which is given in Figure 3.

Step 4 of the algorithm can reuse marginal distributions found in step 3 to speed up the computations.

Notice that it is always possible to compute interestingness of every itemset in step 6 since both supports of each itemset will be computed either in steps 1 and 3, or in steps 4 and 5.

The authors implemented hierarchical and topological interestingness as a postprocessing step. They could however be used to prune the attribute sets which are not interesting without evaluating their distributions, thus providing a potentially large speedup in the computations. We plan to investigate that in the future.

5. EXPERIMENTAL RESULTS

In this section we present experimental evaluation of the method. One problem we were faced with was the lack of publicly available datasets with nontrivial background knowledge that could be represented as a Bayesian network. The UCI Machine Learning repository contains a few datasets with background knowledge (Japanese credit, molecular biology), but they are aimed primarily at Inductive Logic Programming: the relationships are logical rather than probabilistic, only relationships involving the class attribute are included. These examples are of little value for our approach.

We have thus used networks constructed using our own common-sense knowledge as well as networks learned from data.

5.1 An Illustrative Example

We first present a simple example demonstrating the usefulness of the method. We use the `KSL` dataset of Danish 70 year olds, distributed with the `DEAL` Bayesian network package [4]. There are nine attributes, described in Table 1, related to the person's general health and lifestyle. All continuous attributes have been discretized into 3 levels using the equal weight method.

We began by designing a network structure based on authors' (non-expert) knowledge. The network structure is

Input: Bayesian network BN, dataset, interestingness threshold ϵ.
Output: all attribute sets with interestingness at least ϵ, and some of the attribute sets with lower interestingness.

1. $\mathcal{K} \leftarrow \{(I, \mathbf{i}) : \mathrm{supp}(I, \mathbf{i}) \geq \epsilon\}$ (using Apriori algorithm)
2. $\mathcal{C} \leftarrow \{I : (I, \mathbf{i}) \in \mathcal{K}$ for some $\mathbf{i} \in \mathrm{Dom}(I)\}$
3. compute P_I^{BN} for all $I \in \mathcal{C}$ using algorithm in Figure 1
4. $\mathcal{K}' \leftarrow \{(I, \mathbf{i}) : \mathrm{supp}_{BN}(I, \mathbf{i}) \geq \epsilon\}$ (using AprioriBN algorithm)
5. compute support in data for all itemsets in $\mathcal{K}' \setminus \mathcal{K}$ by scanning the dataset
6. compute interestingness of all itemsets in $\mathcal{K} \cup \mathcal{K}'$
7. $\mathcal{C}' \leftarrow \{I : (I, \mathbf{i}) \in \mathcal{K}'$ for some $\mathbf{i} \in \mathrm{Dom}(I)\}$
8. compute interestingness of all attribute sets I in $\mathcal{C}' \cup \mathcal{C}$:

$$\mathcal{I}(I) = \max\{\mathcal{I}(I, \mathbf{i}) : (I, \mathbf{i}) \in \mathcal{K} \cup \mathcal{K}', \mathbf{i} \in \mathrm{Dom}(I)\}$$

Figure 3: Algorithm for finding all ϵ-interesting attribute sets.

FEV	Forced ejection volume of person's lungs
Kol	Cholesterol
Hyp	Hypertension (no/yes)
BMI	Body Mass Index
Smok	Smoking (no/yes)
Alc	Alcohol consumption (seldom/frequently)
Work	Working (yes/no)
Sex	male/female
Year	Survey year (1967/1984)

Table 1: Attributes of the KSL dataset.

given in Figure 4a. Since we were not sure about the relation of cholesterol to other attributes, we left it unconnected.

Conditional probabilities were estimated directly from the KSL dataset. Note that this is a valid approach since even when the conditional probabilities match the data perfectly interesting patterns can still be found because the network structure usually is not capable of representing the full joint distribution of the data. The interesting patterns can then be used to update the network's structure. Of course if both the structure and the conditional probabilities are given by the expert, then the discovered patterns can be used to update both the network's structure and conditional probabilities.

We applied the algorithm for finding all interesting attribute sets to the KSL dataset and the network, using the ϵ threshold of 0.01. The attribute sets returned were sorted by interestingness, and top 10 results were kept.

The two most interesting attribute sets were $\{FEV, Sex\}$ with interestingness 0.0812 and $\{Alc, Year\}$ with interestingness 0.0810.

Indeed, it is known (see [8]) that a women's lungs are on average $20\% - 25\%$ smaller than men's lungs, so sex influences the forced ejection volume (FEV) much more than

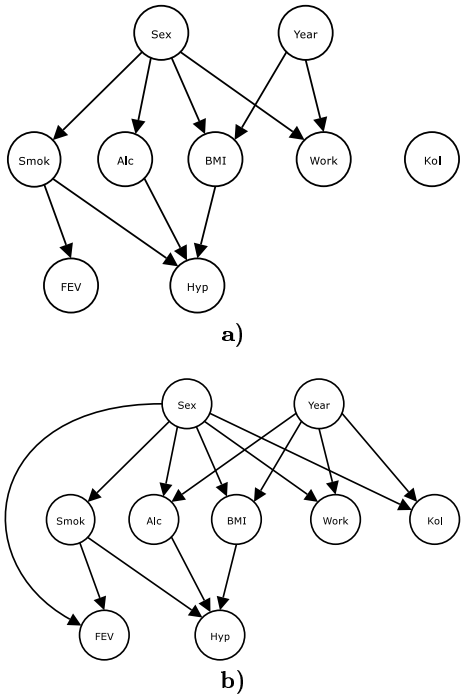

Figure 4: Network structures for the KSL dataset constructed by the authors

smoking does (which we thought was the primary influence). This fact, although not new in general, was overlooked by the authors, and we suspect that, due to large amount of literature on harmful effects of smoking, it might have been overlooked by many domain experts. This proves the high value of our approach for verification of Bayesian network models.

The data itself implied a growth in alcohol consumption between 1967 and 1984, which we considered to be a plausible finding.

We then decided to modify the network structure based on our findings by adding edges $Sex \rightarrow FEV$ and $Year \rightarrow Alc$. One could of course consider other methods of modifying network structure, like deleting edges or reversing their direction. A brief overview of more advanced methods of Bayesian network modification can be found in [15, Chap. 3, Sect. 3.5]. Instead of adapting the network structure one could keep the structure unchanged, and tune conditional probabilities in the network instead, see [15, Chap. 3, Sect. 4] for details.

As a method of scoring network structures we used the natural logarithm of the probability of the structure conditioned on the data, see [10, 26] for details on computing the score.

The modified network structure had the score of -7162.71 which is better than that of the original network: -7356.68.

With the modified structure, the most interesting attribute set was $\{Kol, Sex, Year\}$ with interestingness 0.0665. We found in the data that cholesterol levels decreased between the two years in which the study was made, and that cholesterol level depends on sex. We found similar trends in the U.S. population based on data from American Heart Association [2]. Adding edges $Year \rightarrow Kol$ and $Sex \rightarrow Kol$

improved the network score to -7095.25.

$\{FEV, Alc, Year\}$ became the most interesting attribute set with the interestingness of 0.0286. Its interestingness is however much lower than that of previous most interesting attribute sets. Also, we were not able to get any improvement in network score after adding edges related to that attribute set.

Since we were unable to obtain a better network in this case, we used topological pruning, expecting that some other attribute sets might be the true cause of the observed discrepancies. Only four attribute sets, given below, were topologically 0.01-interesting.

$\{Kol, BMI\}$	0.0144
$\{Kol, Alc\}$	0.0126
$\{Smok, Sex, Year\}$	0.0121
$\{Alc, Work\}$	0.0110

We found all those patters intuitively valid, but were unable to obtain an improvement in the network's score by adding related edges. Moreover, the interestingness values were quite small. We thus finished the interactive network structure improvement process with the final result given in Figure 4b.

The algorithm was implemented in Python and used on a 1.7GHz Pentium 4 machine. The computation of interestingness for this example took only a few seconds so an interactive use of the program was possible. Further performance evaluation is given below.

5.2 Performance Evaluation

We now present the performance evaluation of the algorithm for finding all attribute sets with given minimum interestingness. We used the UCI datasets and Bayesian networks learned from data using B-Course [26]. The results are given in Table 2.

The *max. size* column gives the maximum size of frequent attribute sets considered. The *# marginals* column gives the total number of marginal distributions computed from the Bayesian network. The attribute sets whose marginal distributions have been cached between the two stages of the algorithm are not counted twice.

The time does not include the initial run of the Apriori algorithm used to find frequent itemsets in the data (the time of the `AprioriBN` algorithm is included though). The times for larger networks can be substantial; however the proposed method has still a huge advantage over manually evaluating thousands of frequent patterns, and there are several possibilities to speed up the algorithm not yet implemented by the authors, discussed in the following section.

The *maximum interestingness* column gives the interestingness of the most interesting attribute set found for a given dataset. It can be seen that there are still highly interesting patterns to be found after using classical Bayesian network learning methods. This proves that frequent pattern and association rule mining has the capability to discover patterns which traditional methods might miss.

To give a better understanding of how the algorithm scales as the problem size increases we present two additional figures. Figure 5 shows how the computation time increases with the number of marginal distributions that must be computed from the Bayesian network. It was obtained by varying the maximum size of attribute sets between 1 and 5. The value of $\epsilon = 0.067$ was used (equivalent to one row in

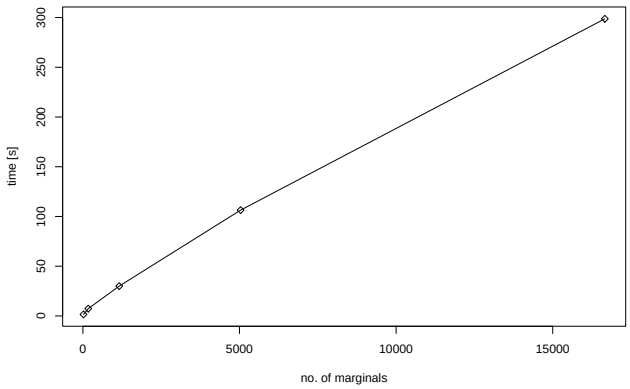

Figure 5: Time of computation depending on the number of marginal distributions computed for the `lymphography` database

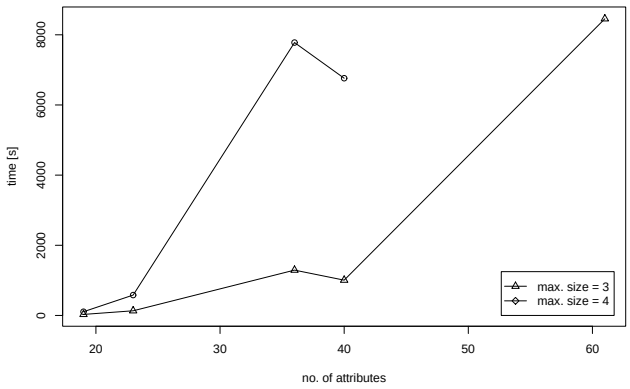

Figure 6: Time of computation depending on the number of attributes for datasets from Table 2

the database). It can be seen that the computation time grows slightly slower than the number of marginal distributions. The reason for that is that the more marginal distributions we need to compute, the more opportunities we have to avoid using bucket elimination by using direct marginalization from a superset instead.

Determining how the computation time depends on the size of the network is difficult, because the time depends also on the network structure and the number of marginal distributions computed (which in turn depends on the maximum size of attribute sets considered).

We nevertheless show in Figure 6 the numbers of attributes and computation times plotted against each other for some of the datasets from Table 2. Data corresponding to maximum attribute set sizes equal to 3 and 4 are plotted separately.

It can be seen that the algorithm remains practically usable for fairly large networks of up to 60 variables, even though the computation time grows exponentially. For larger networks approximate inference methods might be necessary, but this is beyond the scope of this paper.

dataset	#attrs	ϵ	max. size	#marginals	time [s]	max. inter.
KSL	9	0.01	5	382	1.12	0.032
soybean	36	0.075	3	7633	1292	0.064
soybean	36	0.075	4	61976	7779	0.072
breast-cancer	10	0.01	5	638	3.49	0.082
annealing	40	0.01	3	9920	1006	0.048
annealing	40	0.01	4	92171	6762	0.061
mushroom	23	0.01	3	2048	132.78	0.00036
mushroom	23	0.01	4	10903	580.65	0.00036
lymphography	19	0.067	3	1160	29.12	0.123
lymphography	19	0.067	4	5036	106.13	0.126
splice	61	0.01	3	37882	8456	0.036

Table 2: Performance evaluation of the algorithm for finding all ϵ-interesting attribute sets.

6. CONCLUSIONS AND DIRECTIONS OF FUTURE RESEARCH

A method of computing interestingness of itemsets and attribute sets with respect to background knowledge encoded as Bayesian networks was presented. We built efficient algorithms for computing interestingness of frequent itemsets and finding all attribute sets with given minimum interestingness. Experimental evaluation proved the effectiveness and practical usefulness of the algorithms for finding interesting, unexpected patterns.

An obvious direction for future research is increasing efficiency of the algorithms. Partial solution would be to rewrite the code in C, or to use some off-the-shelf highly optimized Bayesian network library like Intel's PNL. Another approach would be to use approximate inference methods like Gibbs sampling.

Adding or removing edges in a Bayesian network does not always influence all of its marginal distributions. Interactivity of network building could be imporved by making use of this property.

Usefulness of methods developed for mining emerging patterns [6], especially using borders to represent collections of itemsets, could also be investigated.

Another interesting direction (suggested by a reviewer) could be to iteratively apply interesting patterns to modify the network structure until no further improvement in the network score can be achieved. A similar procedure has been used in [24] for background knowledge represented by rules.

It should be noted however that it might be better to just inform the user about interesting patterns and let him/her use their experience to update the network. Manually updated network might better reflect causal relationships between attributes.

Another research area could be evaluating other probabilistic models such as log-linear models and chain graphs instead of Bayesian networks.

7. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. ACM SIGMOD Conference on Management of Data*, pages 207–216, Washington, D.C., 1993.

[2] American Heart Association. Risk factors: High blood cholesterol and other lipids. `http://www.americanheart.org/downloadable/heart/1045754065601FS13CH03.pdf`, 2003.

[3] R. J. Bayardo and R. Agrawal. Mining the most interesting rules. In *Proc. of the 5th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pages 145–154, August 1999.

[4] Susanne G. Bøttcher and Claus Dethlefsen. Deal: A package for learning bayesian networks. `www.math.auc.dk/novo/Publications/bottcher:dethlefsen:03.ps`, 2003.

[5] Rina Dechter. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113(1-2):41–85, 1999.

[6] Guozhu Dong and Jinyan Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proc. of the 5th Intl. Conf. on Knowledge Discovery and Data Mining (KDD'99)*, pages 43–52, San Diego, CA, 1999.

[7] William DuMouchel and Daryl Pregibon. Empirical bayes screening for multi-item associations. In *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining*, pages 67–76, 2001.

[8] H. Gray. *Gray's Anatomy*. Grammercy Books, New York, 1977.

[9] Venky Harinarayan, Anand Rajaraman, and Jeffrey D. Ullman. Implementing data cubes efficiently. In *Proc. ACM SIGMOD*, pages 205–216, 1996.

[10] David Heckerman. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Redmond, WA, 1995.

[11] R. Hilderman and H. Hamilton. Knowledge discovery and interestingness measures: A survey. Technical Report CS 99-04, Department of Computer Science, University of Regina, 1999.

[12] C. Huang and A. Darwiche. Inference in belief networks: A procedural guide. *Intl. Journal of Approximate Reasoning*, 15(3):225–263, 1996.

[13] S. Jaroszewicz and D. A. Simovici. A general measure of rule interestingness. In *5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2001)*, pages 253–265, 2001.

[14] S. Jaroszewicz and D. A. Simovici. Pruning redundant association rules using maximum entropy principle. In *Advances in Knowledge Discovery and Data Mining,*

6th Pacific-Asia Conference, PAKDD'02, pages 135–147, Taipei, Taiwan, May 2002.

[15] Finn V. Jensen. *Bayesian Networks and Decision Graphs*. Springer Verlag, New York, 2001.

[16] Bing Liu, Wynne Hsu, and Shu Chen. Using general impressions to analyze discovered classification rules. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, page 31. AAAI Press, 1997.

[17] Bing Liu, Wynne Jsu, Yiming Ma, and Shu Chen. Mining interesting knowledge using DM-II. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 430–434, N.Y., August 15–18 1999.

[18] Heikki Mannila. Local and global methods in data mining: Basic techniques and open problems. In *ICALP 2002, 29th International Colloquium on Automata, Languages, and Programming*, Malaga, Spain, July 2002. Springer-Verlag.

[19] Heikki Mannila and Hannu Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.

[20] T.M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[21] Kevin Murphy. A brief introduction to graphical models and bayesian networks. `http://www.ai.mit.edu/~murphyk/Bayes/bnintro.html`, 1998.

[22] B. Padmanabhan and A. Tuzhilin. Belief-driven method for discovering unexpected patterns. In *Proceedings. of the 4th International Conference on Knowledge Discovery and Data Mining (KDD'98)*, pages 94–100, August 1998.

[23] B. Padmanabhan and A. Tuzhilin. Small is beautiful: discovering the minimal set of unexpected patterns. In *Proceedinmgs of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'00)*, pages 54–63, N. Y., August 2000.

[24] B. Padmanabhan and A. Tuzhilin. Methods for knowledge refinement based on unexpected patterns. *Decision Support Systems*, 33(3):221–347, July 2002.

[25] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, Los Altos, CA, 1998.

[26] P.Myllymäki, T.Silander, H.Tirri, and P.Uronen. B-course: A web-based tool for bayesian and causal data analysis. *International Journal on Artificial Intelligence Tools*, 11(3):369–387, 2002.

[27] D. Poole and N. L. Zhang. Exploiting contextual independence in probablisitic inference. *Journal of Artificial Intelligence Research*, 18:263–313, 2003.

[28] D. Shah, L. V. S. Lakshmanan, K. Ramamritham, and S. Sudarshan. Interestingness and pruning of mined patterns. In *1999 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 1999.

[29] Abraham Silberschatz and Alexander Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *Knowledge Discovery and Data Mining*, pages 275–281, 1995.

[30] E. Suzuki. Autonomous discovery of reliable exception rules. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, page 259. AAAI Press, 1997.

[31] E. Suzuki and Y. Kodratoff. Discovery of surprising exception rules based on intensity of implication. In *Proc of PKDD-98, Nantes, France*, pages 10–18, 1998.

[32] P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proc of the Eighth ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD-2002)*, pages 32–41, 2002.

[33] M. J. Zaki. Generating non-redundant association rules. In *Proceedinmgs of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-00)*, pages 34–43, N. Y., August 20–23 2000.