Iterative bayesian network implementation by using annotated association rules

Clément Fauré[1,2], Sylvie Delprat[1], Jean-François Boulicaut[2], and Alain Mille[3]

[1] EADS CCR, Learning Systems Department, Centreda 1, F-31700 Blagnac,
{clement.faure, sylvie.delprat}@eads.net
[2] LIRIS UMR 5205, INSA Lyon, Bâtiment Blaise Pascal, F-69621 Villeurbanne
[3] LIRIS UMR 5205, Université Lyon 1, Nautibus, F-69622 Villeurbanne
{amille, jboulica}@liris.cnrs.fr

Abstract. This paper concerns the iterative implementation of a knowledge model in a data mining context. The proposed approach relies on coupling a bayesian network design with an association rule discovery technique. First, discovered association rule relevancy is enhanced by exploiting the expert knowledge encoded within a bayesian network, i.e., avoiding to provide trivial rules w.r.t. the available expertise. Moreover, the bayesian network can be updated thanks to an expert-driven annotation process on computed association rules. Our approach is experimentally validated on both synthetic and real data. We sketch a practical case study for which the data report on operational interruptions in the aeronautic industry.

1 Introduction

One major goal of the knowledge discovery from databases (KDD) community is to support the discovery of valuable information or patterns within the data. In the so-called transactional data sets (say 0/1 data), the association rule mining technique is quite popular. It has been studied extensively from the computational perspective. Many researchers have been considered the relevancy as well. Clearly, valuable patterns have to be valid statements (e.g., w.r.t. some objective interestingness criteria like confidence or lift). It is also needed that we support the discovery of useful ones w.r.t. expert expectation, i.e., the so-called subjective interestingness criteria. In this research paper, we consider that expert expectation is related to novelty, i.e., patterns like association rules are valuable if they provide some information which is somehow new given the encoded domain knowledge. Furthermore, the encoded knowledge for Process n might be updated for Process $n + 1$ by using the expert-driven annotation of patterns extracted during Process n. More concretely, we focus on association rule mining when a bayesian network (BN) captures domain knowledge. Given the BN, some extracted rules can be filtered out. Then, we suggest to perform an expert-driven annotation of the presented rules and these annotations are then used to perform updates on the BN. Doing so, we propose a methodology which iteratively improves both the model for expert domain knowledge and the relevancy of the extracted patterns.

[1] introduces an interestingness measure for frequent itemsets, i.e., the computationally difficult step for the association rule mining task [2]. They propose to use a BN to specify the "expected" distribution of the data. It is then possible to filter truly interesting frequent sets when their frequencies are somehow surprising given the BN. In the preliminary paper [3], we have extended this approach to support relevant association rule mining when we assume that (a) a BN captures expert knowledge about domain dependencies, and (b) we compute a sub-collection of the frequent and valid association rules, the so-called δ-strong association rules, i.e., rules with at most δ exceptions. So far, the implementation and the update of such BN models remain open problems. For instance, an expert on aircraft operational interruption data may find difficult to build and update such models. Indeed, this is harder when considering huge amount of high dimensional heterogeneous data. Our proposal is illustrated on the well-known "Asia" dataset, and on real-life data from an industrial application case dealing with aircraft operational interruptions.

The paper is organized as follows. Section 2 introduces our approach to support knowledge discovery by means of association rule extraction and analysis. Section 3 is dedicated to the experimental validation of the added-value. Section 4 is a brief conclusion.

2 Modelling and using expert knowledge

Our proposal relies on five steps:

1. Modelling an initial BN which represents *a priori* expert domain knowledge.
2. Computing concise collections of association rules with high confidence.
3. Supporting rule post-processing (i.e., filtering) by using the knowledge model.
4. Supporting expert annotation of the most interesting rules.
5. Updating the BN structure and parameters given the collected annotations.

Figure 1 provides an overview of the whole KDD process. In this paper, we focus on Points 2 to 4.

First, we are convinced that nuggets of knowledge can indeed be captured via association rule computations. We assume the reader is familiar with the popular definitions concerning this data mining task. Many algorithms have been designed for computing frequent and valid association rules since the popular Apriori algorithm proposal [2]. When computing all the rules with enough frequency and enough confidence, it is well-known that Apriori-like algorithms can not cope with dense and/or strongly correlated 0/1 datasets, at least for the desired frequency thresholds. When the computation is tractable, the huge number of extracted rules which include many irrelevant ones is known as a real bottleneck for association rule based KDD processes. Part of the problem concerns redundancy.

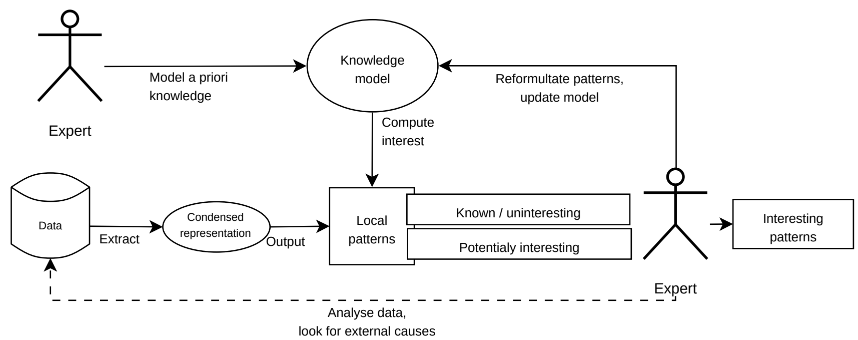

Fig. 1. Process overview

Let us first consider application-independent redundancy, i.e., sets of rules which provides fundamentally the same information. This has been addressed seriously by means of the closed sets and related approaches on the so-called condensed representations (see, e.g., [4] for an overview). Our approach to this problem is to use a concise subset of frequent and valid association rules called the δ-strong rules [5, 6]. The given technique computes the so-called frequent δ-free sets which will lead to minimal left-hand sides (LHS) for the rules. Their right-hand sides are computed as the δ-closure of the LHSs. The δ parameter determines the number of exceptions tolerated for the rules and we assume its value is low w.r.t. the specified frequency threshold. As a result, this technique provides a subset of all the frequent association rules with high confidence.

A second problem concern application-dependant redundancy. Apart from simple *template* based strategies or the exploitation of taxonomies on attributes (see, e.g., [7]), few authors have been considering how to remove rules which do not provide valuable information given an explicitly encoded model for available knowledge. To address this second issue, we started to investigate in [3] the integration of expert knowledge into the computation of rule objective interestingness. Our idea is that available knowledge might be captured within a bayesian network. The modelled dependencies can help us to filter out the extracted patterns that reflect these dependencies. Doing so, we support the presentation of more interesting and unexpected patterns. To address the inherent complexity and the dynamics of knowledge discovery process, we also consider that the knowledge model (BN) has to be iteratively refined and updated: the initial model can be improved by exploiting expert evaluations on extracted rules.

2.1 Association rule post processing using a bayesian network

Modelling and exploiting knowledge to support the discovery of relevant association rules has already given rise to some proposals. [8] studies the exploitation of expert knowledge elicited by expert rules. This has then been formalized into a belief system [9] that might enable the extraction of a kind of minimal set of association rules. However this kind of approach has a major limitation. Indeed, a rule is said to be interesting if it differs from the rules according to what is currently defined in the belief system, but not by looking at what could be inferred from those rules. Jaroszewicz et al. [1, 10] have tackled this issue by modelling bayesian networks for which "inference" is obviously integrated within the model. They describe the use of a BN to compute the interest of all the frequent itemsets an Apriori-like algorithm can compute. For each frequent itemset, the difference between the support value estimated on the data and the support value inferred from the BN is computed. The more interesting patterns are the ones with the higher absolute difference value between these two measures. These itemsets can be submitted to the expert for a potential manual update of the structure and parameters of the BN.

We have a similar approach but we exploit further the interesting complementarities between bayesian networks and association rules, namely dependency links between variables (directed arcs of the graph, association relationship expressed by a rule) and frequencies for specific events (conditional probabilities defined in a BN, support of an association rule). We address "separately" these two relationships. First, we define an interestingness measure on association rule w.r.t. a BN. Then we propose an algorithm for computing relations of independence in the association rules w.r.t. the structure of the BN.

Interestingness measure of an association rule given a bayesian network. Let DB be a boolean database (i.e. a database where each record is a set of boolean values), and $H = \{A_1, A_2, \ldots, A_n\}$ the set of boolean attributes. H is defined on $D_H = D_{A_1} \times D_{A_2} \times \ldots \times D_{A_n}$. $P_I^{DB}(i)$ denotes the probability that a given set of attributes $I \subseteq H$ takes the vector i of boolean values. An itemset is represented by a pair (I, i) with $I \in H$ a non empty set of attributes and i a set of values taken by the attributes of I. When it will not be strictly necessary itemset (I, i) will be simply designated by I.

A bayesian network BN is a *directed acyclic graph*, or DAG, defined by a set of nodes corresponding to the attributes of H and by $E \subset H \times H$ the set of arcs in the graph. Each node is associated with a conditional probability distribution $P_{A_i|\Pi_{A_i}}$, where $\Pi_{A_i} = \{A_j|(V_{A_j}, V_{A_i}) \in E\}$ are the parents of node A_i. For a more detailed discussion on bayesian networks the reader may consult [11]. One of the most important properties of bayesian networks is that they uniquely define the joint probability distribution over H.

$$P_H^{BN} = \prod_{i=1}^{n} P_{A_i|\Pi_{A_i}} \tag{1}$$

On the other hand, an association rule R is a pattern $X \Rightarrow Y$, where X and Y are itemsets such as $Y \neq \emptyset$ and $X \cap Y = \emptyset$. The support of an itemset I in DB, noted $supp_{DB}(I)$, is the set of all records of DB that includes I.

Thus, given a database DB defined on a set of attributes H and a bayesian network BN, it is possible to compute the confidence of an association rule $R = X \Rightarrow Y$ (see [12] for an example of an inference algorithm). By extending [1], we have defined a metric that relates the potential interest of a given association rule w.r.t. knowledge defined in by the bayesian network. This metric is based on the difference between the confidence of the rule estimated on the data and the one inferred by the bayesian network. For any association rule $R = X \Rightarrow Y$ this measure is expressed as follow:

$$\mathrm{I}nt(R) = |conf_{DB}(R) - conf_{BN}(R)| \tag{2}$$
$$\text{where} \quad conf_{DB}(R) = \frac{supp_{DB}(X \cup Y)}{supp_{DB}(X)}$$
$$\text{and} \quad conf_{BN}(R) = \prod_{i=1}^{m} P_{Y_i|\Pi_{Y_i}}$$

Computation of the structural differences between an association rule and a bayesian network. So far, we can compute concise collections of association rules, we have a formalism to express a priori expert knowledge, and a metric which takes into account the described knowledge to measure the interest of a given association rule. What is missing is a way to exploit the information of conditional independence implicitly captured by the bayesian network. The goal is to highlight which parts of an association rule really contribute -according to the network- to the observation of the whole rule, and which parts are not.

Let us first define the d-separation property which has been introduced by J. Pearl [11]. We start by specifying *active paths*. A path between two sets of nodes is said to be *active* if it carries some information (or dependence). More formally,

Definition 1 (Active / blocked path). *Let $G = (V, E)$ be a DAG. Let $C = (x_i)_{i \in I}$ be a path in G and Z be a subset of nodes of G. C is an active path with respect to Z if the two following conditions are met:*

- *Every convergent node of C has one of its descendants, or himself, in Z.*
- *Among the elements of C where the path is not convergent, none is in Z.*

A non active path is said to be blocked.

Definition 2 (D-separation). *For every triplet (X, Y, Z) of disjoints subsets of a DAG G, X is d-separated from Y by Z in G (noted $< X|Z|Y >_G$) if and only if every path $(x_i)_{i \in \{1 \ldots p\}}$ with $x_1 \in X$ and $x_p \in Y$ is blocked by Z.*

We want to apply the notion of *d-separation* -which is a graphical property- on association rules w.r.t. the BN structure. For any association rule $R = X \Rightarrow Y$ we will compute the d-separation test $< X_i \,|\, X \backslash X_i \,|\, Y_j >$, where $X_i \in X$ and $Y_j \in Y$. We end up with a matrix that sums up the results of all the d-separations tests. If an item of the rule (X_i or Y_i) has a "true" value for all its d-separation tests, then it will be highlighted as being in the *d-separated part* of the rule. It means that thanks to the rule, an informative association has been found in the data which is not modelled in the current BN structure (see Section 3 for experimental results).

2.2 Post-processing and annotation of association rules

Let say that we have a BN that reflects most of the domain (in)dependencies. This network could have been defined either from scratch by an expert or through a mixed approach involving expert but also automatic learning. Also note that the initial BN does not have to be "complete". For instance, it can capture the obvious dependencies that underly the domain data, including known taxonomies over the attributes. As we go through the KDD process, this initial BN can be updated to capture more and more domain knowledge, thus supporting the presentation of more and more valuable association rules. At each iteration, the expert might annotate the rules by labeling which parts represent what kind of information. This annotation can be used to improve the BN and support focus on the next iteration.

Once our δ-strong rules have been extracted, we compute their interest as well as their "topological" differences w.r.t. the current BN. These measures are used to filter uninteresting rules (interest compared with a user-defined threshold ϵ). It divides the rules in two classes. A first class contains the rules that do not provide further information w.r.t. the BN (interest below ϵ). The expert who is inspecting the rules can decide to ignore them. The second class represents the rules that we call ϵ-interesting. They express that some dependencies observed on the data are not described properly by the BN. The goal here is to remove rules that are ϵ-interesting but are already known by the domain expert or have been found to contain non valid patterns. The idea is to refine the knowledge model by integrating step by step dependencies that were not identified at the very beginning of the process. Understanding what information is contained by a given association rule is however a difficult task. This is why we want to highlight rule-like sub patterns of an association rule that represent a notion of d-separation between items on the LHS and items on the RHS given the BN structure and the observation of all the LHS items. We can further divide the association among the ϵ-interesting rules in three different types:

- **K** The rule contains a sub pattern already known by the expert but that is not modelled in the current BN. It means that the structure and the parameters of the network have to be updated to integrate the causality related to this pattern. Doing so, this pattern will not be presented as ϵ-interesting in the next iteration.
- **NV** The rule contains a pattern which appears not valid given the expert knowledge. This might be due to statistic coincidences (false positive) but this is not a valuable information and it will be labelled as being "non valid".
- **I** The rule holds a pattern that is potentially interesting. It has been "surprising" for the expert, and a deeper analysis has confirmed its relevancy.

(NV) and (I) categories are both resembling patterns that are *a priori* interesting. The distinction between these two groups has to be decided thanks to expert subjective interestingness. It might also need further validity assessment. In a real world data mining process, the number of association rules that fit in these different categories can be huge. Moreover, a relation of association may contain mixed kinds of patterns. For example, it is possible to have a rule that presents one or more patterns of type K (already known information), one or more patterns of type NV (non valid), and possibly an interesting sub pattern (I). This might lead to tedious analysis tasks. We propose to ask the domain expert for annotations on the most interesting extracted association rules. He/she has to perform annotation following a precise method. These annotations can then be exploited to update the structure and the parameters of the BN.

One of the issues to prepare association rule annotation is the definition of a syntax that support the description of rule information content. We use a simple notation specified by means of a BNF grammar as follows:

```
annotation-list ::= annotation-list annotation | annotation
annotation      ::= '('left-hand-side '=>' element ';' category')'
element-list    ::= element-list 'and' element | element
element         ::= attribute | attribute'='value
category        ::= K':'verbal-probability | NV | I
```

This notation enables the expert to:

- specify whether an association rule contains one or more known patterns (K), non valid patterns (NV) or a potentially interesting one (I).
- define without ambiguity the "shape" of that patterns through the definition of a list of patterns which can only have one item in the right-hand side of the rule.
- be generic concerning the description of the detected patterns (providing only the name of the attribute or an attribute-value pair).
- define when needed a conjunction of attributes or items in the left-hand side of the pattern.
- associate a verbal-probability to patterns labeled as "already known" by following the idea of probability-ladder presented in [13].

2.3 Bayesian network updating by using annotated association rules

Setting and exploiting rule annotations has to be performed by one or more experts because both application domain and bayesian network expertise are needed. We have defined three categories for the annotation of association rules. This categorization provides guidelines for further uses of the annotations.

First, (NV) labeled patterns can serve for a specific post-processing filtering. They are actually handled outside our BN framework because they represent information that we do not want to model. Discussing further this issue is out of the scope of this paper. Then, patterns labeled as (K) represent already known information not modeled in the current BN. It is possible to perform a modification of the BN structure and/or of the conditional associated probability tables. Interesting patterns labeled (I) contain previously unknown (or contradictory) information that has been validated by the expert in charge of the annotation. These patterns can be exploited as (K) patterns.

The modifications that are made in this step of the process have to carefully reflect the discoveries related to the domain knowledge. From that perspective, they will have an "indirect" impact on the association rule analysis. Our main goal remains however the iterative implementation of a knowledge model. Moreover, all the modifications have to be validated by human experts because we do not believe that this can be done automatically.

Finally, Figure 2 provides a detailed view on our KDD process. Examples of the different patterns and their integration in the BN are given in the next section.

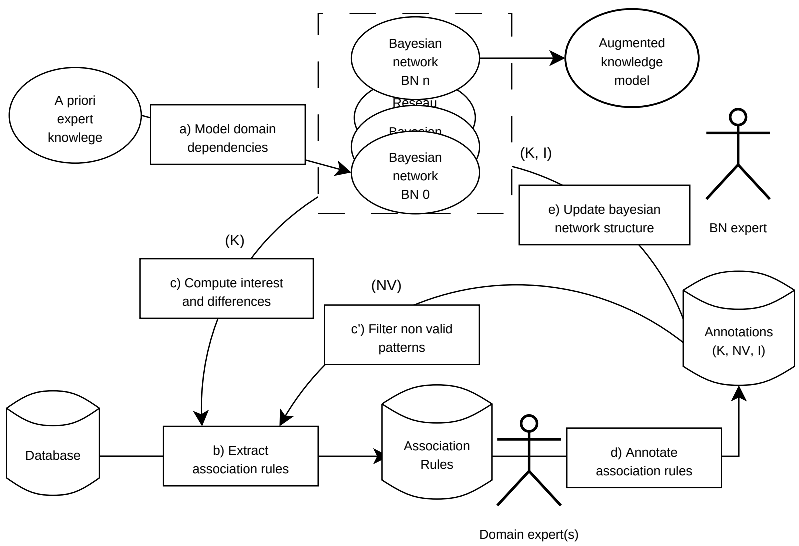

Fig. 2. A more detailed view on our data mining process

3 Experimental validation

3.1 Asia dataset

Let us first consider some synthetic data. We choose an initial BN that already captures a good view on a particular domain knowledge for the Asia dataset known within the BN community. From this network, we produced a dataset

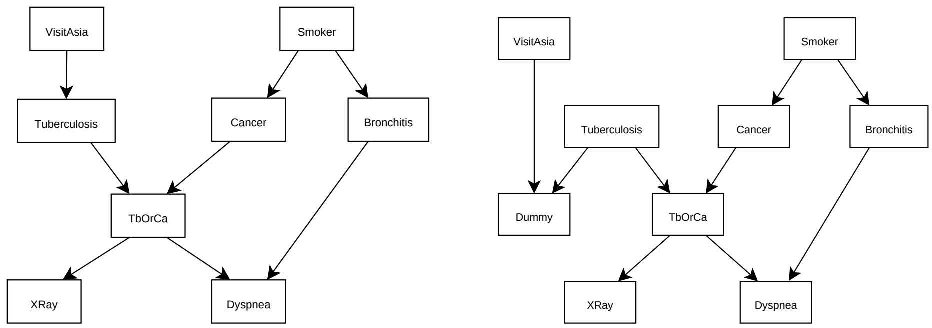

Fig. 3. Original Asia network (left) and modified one (right)

of $10,000$ records. As we look for association rules, we focus on the *presence* of events. The initial BN structure is then modified so that the "VisitAsia" node is no longer directly connected to the "Tuberculosis" node. The goal of this experiment is to apply our methodology and see whether we can find the right "Asia" network structure. Both networks are given in Figure 3.

Let us now follow the method described in Figure 2 step by step.

(a) The modified "Asia" BN serves as a basis for our experiments.

(b) From the generated dataset, we extract a concise collection of association rules (minimum absolute support value of 100, i.e., 0.01% of the database and maximal number of exceptions $\delta = 10$, i.e., a guarantee that the minimal confidence is $0,9$). A total of 16 association rules are extracted immediately.

(c) Interest measure and d-separations are computed on these rules w.r.t. the modified "Asia" BN. **(c')** Filtering out non valid sub patterns is optional and depends on the identification of such patterns. Obtained results are shown in Table 1. By looking at these results, let us recall that association rule mining capture only patterns over true values, i.e., presence of particular events. For example, the third rule should be read as "when we observe that a person is a smoker, a presence of dyspnea and bronchitis diagnosis and special node 'TbOrCa' is activated, then it is often associated with abnormal x-ray result". The underlined part of the rules denotes what we call the *core dependencies* of the rule w.r.t. our BN structure. The last column, however, shows a sub pattern of the association rule which contains a missing information within the current network. Looking at these results, only one association rule has a relatively high interest value. This association rule states that "when we observe that a person has visited Asia then it is associated with abnormal x-ray results". Clearly, this rule brings an information which is not modeled as a dependence in our modified BN structure while it was represented in the original one. It is thus possible to find rules that exhibit a difference between the available knowledge model and the data. We can however wonder whether such discovered associations are truly interesting. Furthermore, if this is the case, what are the modifications to be made to the model to reflect these observations in the data? This is of course where an expert judgment is crucially needed.

(d) An expert can now perform annotations. For our running example, assume that he/she has to put down that the rule which contains the "VisitAsia" relation belongs to the interesting category.

(e) Finally, this annotation is forwarded to the expert who is in charge of BN updating. By looking at the interesting pattern, it leads to a structural modification that provides the initial BN structure. We consider that the association rule actually found is sufficient for an expert to suggest the "right" revision of the BN. Notice that if we compute the rules on the same dataset but using the initial Asia network, we observe that the "VisitAsia" association rule no longer holds a d-separated pattern.

We also tried a similar approach but using an Apriori algorithm with the same constraints and the same dataset. A total of 115 association rules were generated. Among all these rules three mentioned different variants of the relation between "VisitAsia" and "abnormal x-ray", also including the "Dyspnea" attribute. The main difference between our approach and this naive one is that in the second case it is much more harder, due to redundant association rules, to focus on the potentially interesting patterns: the expert will have to go through all the rules to find out about an association that involves "VisitAsia".

Table 1. Association rules extracted from the Asia dataset. Underlined items belong to the *core dependencies* of the rule

Association rule	Interest	D-separated part
Tuberculosis ⇒ XRay Dyspnea TbOrCa	0,04	
VisitAsia ⇒ XRay Dyspnea	0,46	VisitAsia ⇒ Xray Dyspnea
Smoking Dyspnea Bronchitis TbOrCa ⇒ XRay	0,03	Smoking Dyspnea Bronchitis ⇒
Dyspnea Bronchitis TbOrCa ⇒ XRay	0,02	Dyspnea Bronchitis ⇒
Smoking Bronchitis TbOrCa ⇒ XRay	0,02	Smoking Bronchitis ⇒
Bronchitis TbOrCa ⇒ XRay	0,02	Bronchitis ⇒
Smoking Dyspnea TbOrCa ⇒ XRay	0,02	Smoking Dyspnea ⇒
Smoking TbOrCa ⇒ XRay	0,02	Smoking ⇒
Dyspnea TbOrCa ⇒ XRay	0,02	Dyspnea ⇒
TbOrCa ⇒ XRay	0,02	
Smoking Dyspnea Cancer ⇒ XRay TbOrCa	0,02	
Dyspnea Cancer ⇒ XRay TbOrCa	0,02	
Dyspnea Bronchitis Cancer ⇒ Smoking XRay TbOrCa	0,00	
Smoking Cancer ⇒ XRay TbOrCa	0,02	Smoking ⇒
Cancer ⇒ XRay TbOrCa	0,02	
Bronchitis Cancer ⇒ Smoking XRay TbOrCa	0,01	

3.2 Operational Interruption dataset

Aircraft development process is currently based on *concurrent engineering* principles to reduce as much as possible the aircraft development cycle. One of the consequences is that operational performance of the aircraft such as operational reliability has to be predicted even earlier in the product development process so that customer requirements can really drive the product design. An Operational Interruption (OI) happens when an aircraft can not take-off during a mission fifteen minutes after the scheduled departure time due to a technical problem (fault, dysfunction). These events are considered to be important for the airlines as the cost induced by these operational interruptions is not negligible. Thus, aircraft engineers must predict, early in the aircraft development process, a realistic estimate of the frequency of operational interruptions that will happen when aircraft will be commercially exploited. These predictions - along with specific engineering constraints- initialize, guide and validate design choices. For this, engineers use a tool that implements a stochastic model that integrates all the parameters known to impact the frequency of operational interruptions. This tool is calibrated and customized by the return of experience obtained from in service aircraft, system and equipment that share similarities with the undergoing project.

Nowadays, research needs tend to focus on the improvement of the computational models used by the operational interruption prediction tool. In this context, in service data mining is interesting because it aims at discovering - previously unknown- factors that could possibly be integrated in these models to achieve even more accurate predictions. We propose to support the discovery process by applying our framework on in service data corresponding to the details of operational interruptions.

An extract of the operational interruption database is presented in Table 2. The data is mainly composed of categorial attributes (attribute-value pairs). Also, for each incident a free text description of what happened is provided. This text is parsed to extract keywords that will be used to enrich the initial database. Once all pre-processing have been made we obtain a binary matrix of about $15,000$ rows for $2,300$ columns. As it was previously stated, association rule mining only deals with observed events (e.g. true values in the database). All computations were made on a standard desktop computer (2 GHz processor, 1 Gb of memory).

Let us now consider the whole process on this concrete use case.

(a) An initial BN has been defined by the domain expert. The resulting network structure is shown in Figure 4. We assume that it captures important dependencies on this domain.

(b) Using the algorithm from [5], we have computed a collection of delta-strong association rules (relative minimum support of $0,001\%$ and a maximum number of exceptions $\delta = 20$). After one minute, a total of $1,811$ δ-strong association rules has been generated.

(c) Interestingness measure of association rules w.r.t. the current BN has been computed. For this, an algorithm based on Kruskal's polytree reduction [14], implemented in the BNJ library [15], was used to make approximate inference computations on the network. This process took about 3 hours for our $1,811$ rules (mean time of 6 seconds per rule). This step also involved the computation of d-separated parts of the association rules w.r.t. the network structure (neglibible run time) Table 3 shows some of the rules.

Table 2. Extract of operational interruption database

ATA	Date	Operator	MSN	Engine Type	Station	Phase	Effect	Delay	Class
0	29/12/1998	OP1	11	EngineXXA	ST3	TX	DY///	0.50	NM
0	30/12/1998	OP1	29	EngineXXA	ST4	CS	DY///	0.83	NA
212351	03/02/1998	OP2	11	EngineXXA	ST4	CS	DY///	0.68	
212600	07/10/1998	OP1	50	EngineXXA	ST1	CS	DY///	0.39	
212634	21/03/1998	OP2	142	EngineXXA	ST4	TX	DY///	0.85	
212634	23/03/1998	OP1	34	EngineXXA	ST3	CS	DY///	1.15	
212634	09/07/1998	OP1	87	EngineXXA	ST3	CS	DY///	0.25	
212634	04/09/1998	OP3	50	EngineXXA	ST8	TO	DY///	16.00	NM
212634	13/09/1998	OP4	42	EngineXXA	ST2	CS	DY///	2.37	
212651	07/09/1998	OP3	151	EngineXXA	ST1	CS	DY///	0.51	NS
212651	16/10/1998	OP5	170	EngineXXA	ST3	CS	DY///	0.42	

Table 3. Examples of association rules extracted on operational interruption dataset with respect to the initial bayesian network

Id	Association rule	Interest	D-separated part
1	SYSTEM, remove, MB, ST1, CS ⇒ DY, OP2	0.14	ata keyword ⇒
2	SYSTEM, none, MB, ST1, CS ⇒ DY, OP2	0.08	ata keyword ⇒
3	ENGINE, MB, ST1, CS ⇒ DY, OP2	0.06	ata ⇒
4	MB, ST2 ⇒ OP4, DY	0.35	⇒ effect
5	MB, ST3 ⇒ OP5, DY	0.26	⇒ effect
6	MB, ST1 ⇒ OP2	0.16	⇒
7	MB, F05 ⇒ OP2, ST1, DY, CS	0.41	⇒ effect phase
8	leak, delay> 6, CS ⇒ CN	0.97	leak delay ⇒
9	remove, otherMB, delay> 6, CS ⇒ CN	0.94	keyword sCat delay ⇒
10	CS, 3753 ⇒ SYSTEM	0.00	⇒

(d) A domain expert has annotated the rules using the syntax presented in Section 2.2. An interesting side effect of our additional computations is that it is possible to sort rules according to their *core dependencies* and/or to filter rules with empty right hand side in the *d-separated* part. After some discussions with the domain expert, it has been clear that this kind of "post-processing" can enhance the annotation process.

Let us consider the rules 1, 2 and 3. They share the same *core dependencies* and thus are displayed together. Moreover, their interestingness measures tell that they are already described by the BN. Thus, they do not need further investigations. Furthermore, by looking at the left-hand side of the d-separated part, the domain expert has decided that an association between `ata` and `operator` was not valid (NV).

The next group includes the rules 4, 5 and 6. Depending on the ϵ threshold chosen (in our exemple 0.25) rules 4 and 5 are ϵ-interesting, while 6 is not. This can be explained by the fact that the conditional probability tables between the attributes `station`, `stationCat` and `operator` were initially slightly under-estimated by the domain expert. Subsequently, the relation "`stationCat and station` $\Rightarrow$ `operator`" can be annotated known (K) and a probability for this event has been estimated.

Finally rules 8 and 9 both show a high measure of interest. By looking more closely at them, the domain expert has told that, for the rule 8 the association between the keyword `leak` and the effect `CN` (cancelation) is not valid. Thus has been labeled by (NV).

The rule 9 holds a potentially interesting association. Indeed the fact that: "when we observe a `remove` operation at a station which is not the main base (`otherMB`) of the company, then it is often associated with a flight cancelation (`CN`)", is a potentially interesting knowledge nugget. To check the validity of this pattern, the expert had to look for exceptions of this rule in the database. We already know that rule 9 has a strong confidence but it is interesting to figure out how many times a remove operation *only* truly lead to a cancelation. After some investigations on the data, it appeared that the confidence on the association was stronger when the event `otherMB` was observed. The expert decided to annotate this patterns as being interesting (I).

Table 4. Summary of the annotation made on our example

Rules	Expert annotation
{1, 2, 3}	(ata $\Rightarrow$ operator ; NV)
{4}	(ST2 $\Rightarrow$ OP4 ; K:'certain')
{5}	(ST3 $\Rightarrow$ OP5 ; K:'almost certain')
{7}	(F05 $\Rightarrow$ OP5 ; K:'certain')
{8}	(leak $\Rightarrow$ CN ; NV)
{9}	(remove and otherMB $\Rightarrow$ CN ; I)

(e) In the last step of our process, the previously written annotations (summarized in Table 4) have been handled by another expert whose task was to integrate them into the knowledge model. Here, we decided to make a straightforward (although a bit rough) integration of all the (K) and (I) annotations. The probability tables have been modified accordingly but they can not be displayed for obvious practical reasons. The modified BN is presented in Figure 4. There is clearly more to say about this step but this is out of the scope of this paper.

After the model update, another iteration started and we went back to **(c)** and **(c')** steps. Table 5 shows the same set of rules, after the model update. In

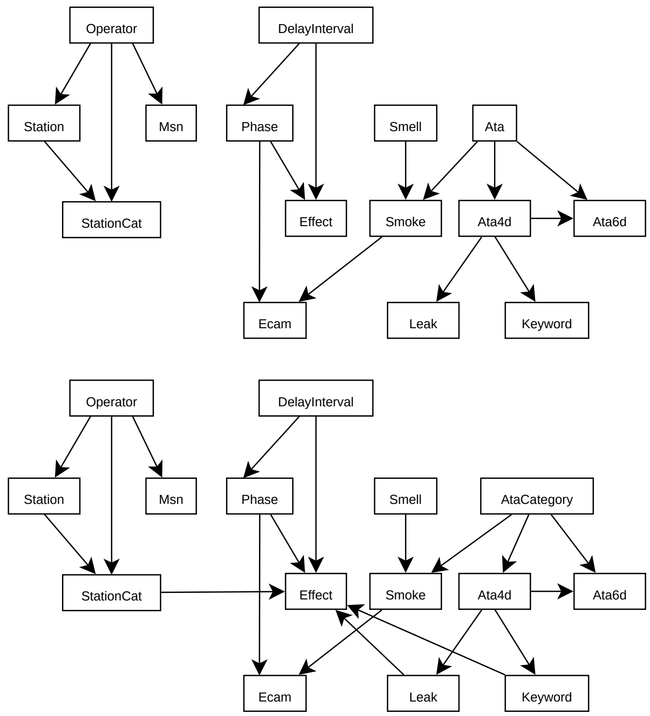

Fig. 4. Initial version of our bayesian network on aircraft operational interruption data (above) and updated one (below)

this example, it appears that already known dependencies have been taken into account. Indeed interestingness values for rules that contained (K) patterns were lowered $(4, 5, 7)$. Rules $1, 2, 4$ and 8 show examples of post-processing based on (NV) patterns. The (NV) items that are not part of the *core dependencies* of the rule are simply stroked-through. That way, potential information are not lost, and non valid patterns are spotted and processed more easily by the domain expert. For rule 9, it shows the effect of integrating interesting knowledge in the knowledge model.

Our results, give rise to some open issues. The rule 8 had a non valid pattern and a high interest value. Here, we managed to get rid only of the non-valid pattern, but the interest remained high, thus inducing a false-positive statement. Another problem to be addressed in future work is related to the potential side effects that a specific modification on the network structure or parameters can have on the other rules.

This concrete example already shows promising results as it has permitted to discover a contributor of the frequency of operational interruption.

Table 5. Examples of association rules extracted on operational interruption dataset w.r.t. the modified BN

Id	Association rule	Interest	D-separated part
1	~~SYSTEM~~, remove, MB, ST1, CS ⇒ DY, OP2	0.14	~~ata~~ keyword ⇒
2	~~SYSTEM~~, none, MB, ST1, CS ⇒ DY, OP2	0.08	~~ata~~ keyword ⇒
3	~~ENGINE~~, MB, ST1, CS ⇒ DY, OP2	0.06	~~ata~~ ⇒
4	MB, ST2 ⇒ OP4, DY	0.12	⇒ effect
5	MB, ST3 ⇒ OP5, DY	0.07	⇒ effect
6	MB, ST1 ⇒ OP2	0.16	⇒
7	MB, F05 ⇒ OP2, ST1, DY, CS	0.03	⇒ effect phase
8	~~leak~~, delay> 6, CS ⇒ CN	0.91	~~leak~~ delay ⇒
9	remove, otherMB, delay> 6, CS ⇒ CN	0.12	delay ⇒
10	CS, 3753 ⇒ SYSTEM	0.00	⇒

4 Conclusion

Looking for relevant local patterns in categorical data sets (i.e., 0/1 data), we have been considering application-dependant redundancy. Our approach concerns association rule filtering when expert knowledge is encoded within a model, namely a bayesian network, and the rule is not enough informative w.r.t. the available knowledge. Then, this paper has focused on the possible revision of such a knowledge model by using discoveries derived from inspection and annotation of selected association rules. The idea is that such a KDD process somehow converges towards truly interesting and valuable information or nuggets of knowledge: discovering new and valid statements in the data suggest refinement on the knowledge model which better captures important dependencies and thus enable to iterate on a more focused pattern discovery phase. While this approach might be considered with various types of local patterns and knowledge models, we have studied in detail an instance where we compute concise collections of a priori interesting association rules (the so-callled δ-strong rules) and we consider that the knowledge model is a bayesian network. An interestingness measure that takes into account expert knowledge and an algorithm that extracts d-separated parts of association rules w.r.t. the BN structure has been proposed. We applied the approach on the “Asia” dataset for a toy illustration of the whole process. We also applied the method on a practical case involving aircraft operational interruption data. The results are promising as the preliminary iterations already lead to an interesting discovery. Further experimentations are however needed. We are also working on a systematic study of the various degrees of freedom with this approach, typically the type of extracted patterns (e.g., other forms of rules), alternative interestingness measures for extracted patterns, or alternative knowledge modeling formalisms.

References

1. Jaroszewicz, S., Simovici, D.A.: Interestingness of frequent itemsets using bayesian networks as background knowledge. In: Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, ACM Press (2004) 178–186
2. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: 12. In: Fast discovery of association rules. American Association for Artificial Intelligence, Menlo Park, CA, USA (1996) 307–328
3. Fauré, C., Delprat, S., Mille, A., Boulicaut, J.F.: Utilisation des réseaux bayésiens dans le cadre de l extraction de règles d association. In: Proceedings EGC'2006. (2006) In French.
4. Goethals, B., Zaki, M.J.: Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations FIMI 2003, Melbourne, USA (2003)
5. Boulicaut, J.F., Bykowski, A., Rigotti, C.: Approximation of frequency queries by means of free-sets. In: Proceedings of the 2000 PKDD European Conference on Principles and Practice of Knowledge Discovery in Databases. (2000) 75–85
6. Boulicaut, J.F., Bykowski, A., Rigotti, C.: Free-sets: a condensed representation of boolean data for the approximation of frequency queries. Data Mining and Knowledge Discovery journal **7**(1) (2003) 5–22
7. Han, J., Fu, Y.: Discovery of multiple-level association rules from large databases. In: VLDB '95: Proceedings of the 21th International Conference on Very Large Data Bases, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1995) 420–431
8. Padmanabhan, B., Tuzhilin, A.: A belief-driven method for discovering unexpected patterns. In: Proceedings of the 1998 KDD International Conference on Knowledge Discovery and Data Mining. (1998) 94–100
9. Padmanabhan, B., Tuzhilin, A.: Small is beautiful: discovering the minimal set of unexpected patterns. In: Proceedings of the 2000 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, ACM Press (2000) 54–63
10. Jaroszewicz, S., Scheffer, T.: Fast discovery of unexpected patterns in data, relative to a bayesian network. In: Proceedings of the 2005 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, ACM Press (2005)
11. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1988)
12. Dechter, R.: Bucket elimination: A unifying framework for reasoning. Artificial Intelligence **113**(1-2) (1999) 41–85
13. Druzdzel, M.J., Diez, F.: Criteria for combining knowledge from different sources in probabilistic networks (2000)
14. Kruskal, J.: On the shortest spanning subtree and the traveling salesman problem. In: American Mathematical Society. Volume 7. (1956) 48–50
15. : Bayesian network tools in java BNJ. http://bnj.sourceforge.net/ (2004) Open-source suite of software tools for research and development.