The editors

Improving the K2 Algorithm Using Association Rule Parameters

Evelina Lamma, Fabrizio Riguzzi and Sergio Storari *

Dipartimento di Ingegneria, Università di Ferrara, Via Saragat 1, 41100 Ferrara, Italy

Abstract

A Bayesian network is an appropriate tool to work with the uncertainty that is typical of real-life applications. Bayesian network arcs represent statistical dependence between different variables and can be automatically elicited from database by Bayesian network learning algorithms such as K2. In the data mining field, association rules can also be interpreted as expressing statistical dependence relations. In this paper we present an extension of K2 called K2-rules that exploits a parameter normally defined in relation to association rules for learning Bayesian networks. We compare K2-rules with K2 and TPDA on the problems of learning four Bayesian networks. The experiments show that K2-rules improves both K2 and TPDA with respect to the quality of the learned network and K2 with respect to the execution time.

Key words: Bayesian Networks. Machine Learning. Association Rules.

1. Introduction

A Bayesian network [1,2] is an appropriate tool to work with the uncertainty that is typical of real-life applications. A Bayesian network is a directed, acyclic graph (DAG) whose nodes represent random variables. In a Bayesian network each node

* Corresponding author.
Email addresses: `elamma@ing.unife.it` (Evelina Lamma), `friguzzi@ing.unife.it` (Fabrizio Riguzzi), `sstorari@ing.unife.it` (Sergio Storari).

is conditionally independent from any subset of nodes that are not its descendants, given its parents.

By means of Bayesian networks, we can use information about the values of some variables to obtain probabilities for other variables. A probabilistic inference takes place once the probabilities functions of each node conditioned to just its parents are given. These are usually represented in a tabular form, called Conditional Probability Table (CPT).

Given a training set of examples, learning a Bayesian network is the problem of finding the structure of the direct acyclic graph and the CPT associated with each node that best match (according to some scoring metric) the dataset. Optimality is evaluated with respect to a given scoring metric (e.g., description length or posterior probability [3–10]). A procedure for searching among possible structures is needed. However, the search space is so vast that any kind of exhaustive search cannot be considered, and often a greedy approach is followed.

The K2 algorithm [4] is a typical search and score method. It starts by assuming that a node has no parents, after which, in every step it adds incrementally the parent whose addition mostly increases the probability of the resulting structure. K2 stops adding parents to the nodes when the addition of a single parent cannot increase the probability of the network given the data. Other search and score methods include the MDL algorithm [10] and the CB algorithm [11].

In this work, we propose the algorithm K2-rules that improves the quality of learned networks and reduces the computational resources needed. This algorithm uses data mining techniques, and in particular the computation of parameters normally defined in relation to association rules [12], to obtain new knowledge to be used for improving some of the steps of K2. Association rules describe correlation of events, and can be viewed as probabilistic rules. Two events are "correlated" when they are frequently observed together. Both Bayesian network arcs and association rules represent dependence relations among variables so it is natural to integrate these methodologies in order to improve Bayesian network learning. Each association rule is characterized by several parameters which can be used to identify the absence of dependence among the nodes. In this work, we exploit in particular the leverage parameter.

The paper presents the results of a comparison between K2, K2-rules and TPDA [13], another well-known learning algorithm. TPDA computes the mutual information of each pair of nodes as a measure of dependence and creates the network using this information.

The paper is structured as follows. Section 2 describes the K2 algorithm. In Section 3, we briefly present association rules. Section 4 illustrates the algorithm K2-rules. In Section 5, we show an experimental comparison between K2, K2-rules and TPDA considering four of the most known Bayesian networks. Section 6 discusses related works. Finally, in Section 7, we conclude and present future work.

2. The K2 algorithm

In the literature, there are different approaches for Bayesian network learning. Some of them are based on the search and score methodology [3–10], and the others follow an information theory based approach [11,13].

A procedure frequently used for learning the structure of a Bayesian network from data is the K2 algorithm [4]. Given a database D, this algorithm searches for the Bayesian network structure G^* with maximal $\Pr(G^*|D)$, where $\Pr(G|D)$ is the probability of network structure G given the database D. Since $\Pr(G_1|D)/\Pr(G_2|D) = \Pr(G_1, D)/\Pr(G_2, D)$ (where G_1 and G_2 are two Bayesian network structures), the authors look for a method to compute $\Pr(G, D)$. Let $V(G)$ be a set of n discrete variables, where a variable $V_i \in V(G)$ has r_i possible value assignments v_{ik} $k = 1, \ldots, r_i$. Let D be a database of m cases, where each case contains a value assignment for each variable in $V(G)$. Let G denote a directed acyclic graph representing the structure of a Bayesian network containing just the variables in $V(G)$, and let GPr be the associated set of conditional probability distributions. Each node $V_i \in V(G)$ has a set of parents $\pi(V_i)$. Let w_{ij} denote the j-th unique instantiation of $\pi(V_i)$ relative to D. Suppose there are q_i unique instantiations of $\pi(V_i)$, so $j = 1, \ldots, q_i$. Define N_{ijk} to be the number of cases in D in which variable V_i has the value v_{ik} and $\pi(V_i)$ is instantiated as w_{ij}. Let

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$$

Given a Bayesian network structure G, assuming that the cases occur independently and the conditional probability density function $f(GPr|G)$ is uniform, then it follows that

$$\Pr(G, D) = \Pr(G) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!$$

The K2 algorithm looks for a network structure G that maximizes $\Pr(G, D)$. In particular, assuming that an ordering on the variables is available and that all structures are equally similar, it adopts a greedy method for maximizing $\Pr(G, D)$. This method consists in searching, for each node V_i, for the set of parent nodes that maximizes the function:

$$g(i, \pi(V_i)) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{1}$$

K2 starts by assuming that a node lacks parents, after which in every step it adds incrementally the parent whose addition mostly increases $g(i, \pi(V_i))$. K2 stops adding parents to a node when the addition of a single parent cannot increase $g(i, \pi(V_i))$.

A pseudo code representation of K2 algorithm is shown in Figure 1.

1 For $i = 1$ to n
{
  1.1 $\pi(V_i) = \emptyset$
  1.2 Repeat
  {
    1.2.1 Select $V_j \in \{V_1, \ldots, V_{i-1}\} - \pi(V_i)$ that
      maximizes $g(i, \pi(V_i) \cup \{V_j\})$
    1.2.2 $\Delta = g(i, \pi(V_i) \cup \{V_j\}) - g(i, \pi(V_i))$
    1.2.3 If $\Delta > 0$ then $\pi(V_i) = \pi(V_i) \cup \{V_j\}$
  } until $\Delta < 0$ or $\pi(V_i) = \{V_1, \ldots, V_{i-1}\}$
}

Fig. 1. Pseudo code representation of the K2 algorithm

3. Association Rules

Association rules [12] describe co-occurrence of events, and can be regarded as probabilistic rules. A good example of association rules is taken from the domain of sale transactions: an association rule in this domain expresses what items are usually bought together, information that is used for developing successful marketing strategies.

Consider a database D consisting of a single table. An association rule [12] is a rule of the form

$$A_1 = v_{A_1}, A_2 = v_{A_2}, \ldots, A_j = v_{A_j} \Rightarrow B_1 = v_{B_1}, B_2 = v_{B_2}, \ldots, B_k = v_{B_k}$$

where $A_1, A_2, \ldots, A_j, B_1, B_2, \ldots, B_k$ are attribute names and $v_{A_1}, v_{A_2}, \ldots, v_{A_j}, v_{B_1}, v_{B_2}, \ldots, v_{B_k}$ are values such that v_{A_i} (v_{B_h}) belongs to the domain of the attribute $A_i(B_h)$ for $i = 1, \ldots, j$ ($h = 1, \ldots, k$).

More formally, an association rule can be defined as follows.

An *item* is a literal of the form $Attribute_i = v_{Attribute_i}$ where $v_{Attribute_i}$ belongs to the domain of $Attribute_i$. Let M be the set of all the possible items. A *transaction* T is a record of the database.

An *itemset* X is a set of items that is consistent, that is a set X such that $X \subseteq M$ and an attribute $Attribute_i$ does not appear twice in X. We say that a transaction T *contains* an itemset X if $X \subseteq T$ or, alternatively, if T satisfies all the literals in X.

The *support* of an itemset X (indicated by $support(X)$) is the fraction of transactions in D that contain X.

An *association rule* is an implication of the form $X \Rightarrow Y$, where X and Y are itemsets and $X \cap Y = \emptyset$. For each association rule $X \Rightarrow Y$ we define the following parameters:

- The *support* of $X \Rightarrow Y$ (represented by $support(X \Rightarrow Y)$) is defined as $support(X \cup Y)$;
- The *leverage* [14] of $X \Rightarrow Y$ (represented by $leverage(X \Rightarrow Y)$) is defined as

$leverage(X \Rightarrow Y) = support(X \cup Y) - support(X) \times support(Y)$. (this parameter is similar to the Absolute Confidence Difference to Prior defined in [15]). It can assume positive and negative values. Since $support(X)$ can be interpreted as $\Pr(X)$, the leverage can be interpreted as $\Pr(X, Y) - \Pr(X) \times \Pr(Y)$. Therefore the more the leverage is close to 0 the more X and Y are statistically independent from each other.

In this paper we consider association rules where both X and Y contain a single item. In this way the leverage of the rule can be interpreted as a measure of the dependence between the two items contained respectively in X and Y.

4. K2-rules algorithm

In this section we describe the K2-rules algorithm which improves the K2 algorithm described in Section 2 by exploiting the leverage parameter of association rules. In order to work, the K2 algorithm requires the total ordering of the nodes. K2 has a high computational cost and produces a significant number of extra arcs in the learned network.

The high computational cost is due to Formula 1 (see Section 2) which requires many computational resources especially for nodes characterized by a great number of parents.

The extra arc problem arises especially when the network is characterized by a lot of root nodes (nodes without parents). During network learning, the algorithm tries to add parents to each of these nodes until it maximizes $g(i, \pi(V_i))$. The algorithm will add at least one arc to a root node because the value of the heuristic for this new structure is always better than the value of the previous structure.

The new proposed approach considers all the association rules containing a single item in the body and a single item in the head. In order to obtain the leverage of these rules, we do not employ an algorithm that learns association rules (such as APRIORI [16]), but we consider all the possible two items rules and for each we compute the leverage. The K2-rules algorithm first computes, for each stochastic variable V_i, the maximum and the minimum of the leverage of the association rules that have an item that refers to V_i. Let $MaxLev(V_i)$ and $MinLev(V_i)$ be these figures.

Then K2-rules finds the minimum of all the $MaxLev(V_i)$ and the maximum of all the $MinLev(V_i)$. Let $MaxLeverage$ and $MinLeverage$ be these figures. Using these parameters, K2-rules deletes nodes from the list of possible parents of a node Q (those that precede it in the order). These parameters are used as thresholds for considering a couple of nodes statistically independent: if the leverage of a rule involving the two nodes is between $MinLeverage$ and $MaxLeverage$, then the two variables are considered independent. Therefore the node that precedes the other in the given order can be removed from the list of parents of the other node.

This method is implemented by the function $FindAllowableParents$ that, given a node, returns the set of allowable parents. We have also considered a more con-

Given the set of network nodes V and the set of learned association rules AR:
1 For $i = 1$ to n
{
    1.1 Select the subset $AR(V_i)$ of association rules from AR
        which involve V_i;
    1.2 Find the minimum and maximum value of leverage of rules in $AR(V_i)$
        and call them $MinLev(V_i)$ and $MaxLev(V_i)$;
}
2 Find the global minimum and maximum for all the network nodes:
    $MinLeverage{=}max_{V_i \in V}\{MinLev(V_i)\}$ and
    $MaxLeverage{=}min_{V_i \in V}\{MaxLev(V_i)\}$
3 For $i = 1$ to n
{
    3.1 $\pi(V_i) = \emptyset$
    3.2 Compute $FindAllowableParents(V_i)$ or $FindAllowableParents\_All(V_i)$
    which return a list $AllowableParents$ of allowable nodes
    3.3 Repeat
    {
        3.3.1 Select $V_j \in AllowableParents - \pi(V_i)$ that
            maximizes $g(i, \pi(V_i) \cup \{V_j\})$
        3.3.2 $\Delta = g(i, \pi(V_i) \cup \{V_j\}) - g(i, \pi(V_i))$
        3.3.3 If $\Delta > 0$ then $\pi(V_i) = \pi(V_i) \cup \{V_j\}$
    } until $\Delta < 0$ or $\pi(V_i) = \{V_1, \ldots, V_{i-1}\}$
}

Fig. 2. Pseudo code representation of the K2-rules algorithm

Given the ordered list of network nodes L, a node Q and the set of learned association rules AR, the $FindAllowableParents$ function:
    1 Associates a list $AllowableParents$ of possible parents to Q
        composed by the nodes that precede Q in the list L
    2 Selects the subset $AR(Q)$ of association rules from AR which involve Q
    3 For each node P in $AllowableParents$
        3.1 if at least a rule R in $AR(Q)$
            exists that involves P and Q, and that has
            $MinLeverage < leverage(R) < MaxLeverage$ then
            3.1.1 Removes P from $AllowableParents$
    4 Return $AllowableParents$

Fig. 3. *FindAllowableParents* function

servative method to remove nodes from the set of allowable parents.

This method is implemented by the function *FindAllowableParents_All*. This

Given the ordered list of network nodes L, a node Q and the set of learned association rules AR, the *FindAllowableParents_All* function:

1 Associates a list $AllowableParents$ of possible parents to Q
  composed by the nodes that precede Q in the list L
2 Selects the subset $AR(Q)$ of association rules from AR which involve Q
3 For each node P in $AllowableParents$
  3.1 if all the rules R in $AR(Q)$
    which involve P and Q have
    $MinLeverage < leverage(R) < MaxLeverage$ then
    3.1.1 Removes P from $AllowableParents$
4 Return $AllowableParents$

Fig. 4. *FindAllowableParents_All* function

function deletes a node P from the list of parents of a node Q only if all the rules involving Q and P are such that their leverage is between $MinLeverage$ and $MaxLeverage$. The K2-rules algorithm is described in pseudo code in Figure 2. The functions $FindAllowableParents$ and *FindAllowableParents_All* are presented in Figures 3 and 4 respectively.

5. Experimental comparisons

We compared K2 and K2-rules on four different Bayesian networks:

– "Visit to Asia": a network for a fictitious medical example about whether a patient has tuberculosis, lung cancer or bronchitis, depending on their X-ray, dyspnea, visit-to-Asia and smoking status. It has 8 nodes and 8 arcs, and is described in [17].
– "Car_diagnosis": a network to diagnose the reason why a car does not start, based on spark plugs, headlights, main fuse, etc. It has 18 nodes and 20 arcs, and is described in [18];
– "ALARM": ALARM stands for "A Logical Alarm Reduction Mechanism". This is a medical diagnostic network to monitor patients. It is a nontrivial network with 8 diagnoses, 16 findings and 13 intermediate variables (37 nodes and 46 arcs), and is described in [19].
– "Boelarge92": A subjective belief network for a particular scenario of neighborhood events, that shows how even distant concepts have some connection. It has 24 nodes and 35 arcs. It is described in [20];

The dataset of examples used for rule learning has been obtained with the NETICA tool [18]. This tool, given the structure and the CPTs of a Bayesian network is able to generate automatically a dataset of N examples. Each experiment was conducted by first generating a dataset from one of the networks above and then trying to learn back the network using K2, K2-rules using $FindAllowableParents$ (K2-r-

	K2				K2-r-FAP				K2-r-FAPA				TPDA	
Data Set	MA	EA	LS	NN	MA	EA	LS	NN	MA	EA	LS	NN	MA	EA
1,000	0	4	66	2280	1	2	52	1980	1	2	52	1980	1	0
5,000	0	2	61	1608	0	1	50	1416	0	1	50	1416	1	0
10,000	0	1	57	1224	1	0	40	900	1	2	44	960	1	0
20,000	0	1	57	1224	1	0	39	852	1	1	47	1020	1	0

Table 1
Comparison of K2, K2-r-FAP and K2-r-FAPA on the Visit-to-Asia network.

	K2				K2-r-FAP				K2-r-FAPA				TPDA	
Data Set	MA	EA	LS	NN	MA	EA	LS	NN	MA	EA	LS	NN	MA	EA
1,000	3	9	396	21705	4	6	270	16710	4	9	331	20679	8	0
5,000	1	7	395	26817	2	2	214	17631	1	7	325	24375	6	0
10,000	1	7	395	26817	2	1	201	16716	1	5	334	24837	7	0
20,000	1	7	395	26817	1	0	206	17490	1	4	311	23745	6	0

Table 2
Comparison of K2, K2-r-FAP and K2-r-FAPA on the Car_diagnosis network.

	K2				K2-r-FAP				K2-r-FAPA				TPDA	
Data Set	MA	EA	LS	NN	MA	EA	LS	NN	MA	EA	LS	NN	MA	EA
1,000	3	13	1793	252120	23	3	143	11919	2	1	937	155178	37	37
5,000	1	11	1771	204462	27	4	119	20163	1	0	685	91572	37	36
10,000	1	11	1771	204462	22	3	148	23782	1	0	771	95235	37	35

Table 3
Comparison of K2, K2-r-FAP and K2-r-FAPA on the ALARM network.

FAP), K2-rules using *FindAllowableParents_All* (K2-r-FAPA) and TPDA. The learned networks are compared to the original network in tables 1, 2, 3 and 4. For each algorithm we indicate: the numbers of missing and extra arcs (MA and EA, respectively); the Log Score (LS) indicating the number of computations of the function $g(i, \pi(V_i))$; the number of computation of N_{ijk} (NN). The last two parameters represent the computational resources needed by the K2, K2-r-FAP and K2-r-FAPA algorithms.

Analyzing these experimental results we can observe that K2-r-FAP and K2-r-FAPA have a number of missing arcs comparable with that of K2 (apart from K2-r-FAP applied to ALARM) but have lower numbers of extra arcs. Moreover, the computational costs of both K2-rules algorithms are significantly lower than those required for K2. In particular, K2-r-FAP is more selective than K2-r-FAPA so it requires the least amount of resources.

Comparing K2-r-FAP and K2-r-FAPA with TPDA we can observe the following: TPDA underestimates the probabilistic relations so it produces a high number of missing arcs while K2-r-FAP and K2-r-FAPA overestimate the probabilistic rela-

	K2				K2-r-FAP				K2-r-FAPA				TPDA	
Data Set	MA	EA	LS	NN	MA	EA	LS	NN	MA	EA	LS	NN	MA	EA
1000	10	3	554	1194	10	1	196	4056	10	1	196	4056	13	1
5,000	7	2	585	14244	7	0	172	4068	7	0	176	4140	12	0
10,000	7	4	601	15732	8	2	199	5136	7	2	202	5284	13	0
20,000	7	4	615	17172	8	3	161	4320	7	3	268	4500	12	0

Table 4
Comparison of K2, K2-r-FAP and K2-r-FAPA on the Boelarge network.

tions so they produce a low number of missing arcs but introduce some extra arcs. The total number of erroneous (missing and extra) arcs of TPDA is higher than the one of the new K2-rules algorithms (especially K2-r-FAPA) except for Visit to Asia and the first two datasets of Car_diagnosis.

6. Related Works

In this paper we present an approach for exploiting parameters related to association rules in order to improve the performance of an algorithm for learning Bayesian networks. Such an approach is not limited to the K2 algorithm only. In fact, in [21], we have applied the same methodology to the TPDA algorithm [13] that is based on an information theory approach rather than on a search and score methodology. In particular, we have exploited association rules parameters for improving the drafting phase of TPDA. This phase is devoted to learn an initial sketch of the network structure.

In [21] we have used the parameters leverage, conviction, lift, Pearson X^2 and Cramer index. We have tested our algorithm (called BNL-rules) with each parameter on four networks together with TPDA. In each test three dimensions of the database were considered: 5,000, 20,000 and 100,000. Of all the algorithm tested, BNL-rules with Pearson X^2 gave the best results outperforming TPDA in five cases in terms of number of missing and extra arcs, equating it in six cases and having a lower performance in only one case.

7. Conclusions

In this work we describe a method for improving K2, one of the most known algorithm for learning Bayesian network, by exploiting association rules parameters.

The K2 algorithm starts by assuming that a node has no parents, after which in every step it incrementally adds the parent whose addition mostly increases the probability of the resulting structure. K2 stops adding parents to the nodes when the addition of a single parent cannot increase the probability of the resulting network structure given the data.

In this work, we propose a method for improving the K2 algorithm, reducing the set of allowable parents from which the algorithm selects actual parents and avoiding extra arc insertions. This new methodology uses data mining techniques, and in particular the computation of association rules parameters from a database of examples, in order to learn the structure of a Bayesian network. Association rules describe correlation of events, and are characterized by several parameters that can be used in structure learning. We have presented the K2-rules algorithm (K2 with association rules) that exploits the leverage parameter of association rules in order to improve the performance of the K2 algorithm.

Experiments discussed in the paper have shown that the proposed approach solves the problem of extra arcs and also notably reduces the computational cost. They also showed the validity of the new algorithms with respect to TPDA.

In future, we plan to compare K2-rules with MDL [10] and other Bayesian network learning algorithms.

Acknowledgments

We would like to thank Andrea Stambazzi and Rossella Bozzini for their help with the experiments. This work is partially funded by the Information Society Technologies programme of the European Commission under the IST-2001-32530 SOCS project.

References

1. J. Pearl, Probabilistic reasoning in intelligent systems: networks of plausible inference, Morgan Kaufmann, San Francisco, 1988.
2. J. H. Kim, J. Pearl, A computational model for combined causal and diagnostic reasoning in inference systems, in: Proceedings of the Eight International Joint Conference on Artificial Intelligence (IJCAI83), Los Angeles, 1983, pp. 190–193.
3. H. Akaike, A new look at statistical model identification, IEEE Trans. Automatic Control 19 (1974) 716–723.
4. G. Cooper, E. Herskovits, A bayesian method for the induction of probabilistic networks from data, Machine Learning 9 (1992) 309–347.
5. D. Heckerman, D. Geiger, D. Chickering, Learning bayesian networks: the combination of knowlegde and statistical data, Machine Learning 20 (1995) 197–243.
6. D. Heckerman, Tutorial on learning in bayesian networks, in: M. Jordan (Ed.), Learning in Graphical Models, MIT Press, Cambridge, MA, 1999, pp. 301–354.
7. E. H. Herskovits, Computer-based probabilistic-network construction, Ph.D. thesis, Medical Informatics, Stanford University (1991).
8. D. Madigan, A. Raftery, Model selection and accounting for model uncertainty in graphical models using occam's window, J. Am. Statist. Association 89 (1994) 1535–1546.

9. J. Rissanen, Stochastic complexity (with discussion), J. Roy. Statist. Soc. B 49 (1987) 223–239.
10. J. Suzuki, Learning bayesian belief networks based on the mdl principle: An efficient algorithm using the branch and bound technique, IEICE Transactions on Communications Electronics Information and Systems.
11. M. Singh, M. Valtorta, Construction of bayesian network structures from data: a brief survey and an efficient algorithm, International Journal of Approximate Reasoning 12 (1995) 111–131.
12. R. Agrawal, T. Imielinski, A. Swami, Mining association rules between sets of items in large databases, in: P. Buneman, S. Jajodia (Eds.), Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993, ACM Press, 1993, pp. 207–216.
13. J. Cheng, R. Greiner, J. Kelly, D. Bell, W. Liu, Learning bayesian networks from data: An information-theory based approach, Artificial Intelligence 137 (1–2) (2002) 43–90.
14. I. Witten, E. Frank, Weka (2003).
URL `http://www.cs.waikato.ac.nz/~ml/weka/doc_gui/weka.associations.ItemSet.html#leverageForRule`
15. C. Borgelt, Absolute confidence difference to prior (2004).
URL `http://fuzzy.cs.uni-magdeburg.de/~borgelt/doc/apriori/apriori.html#diff`
16. R. Agrawal, R. Srikant, Fast algorithms for mining association rules, in: J. Bocca, M. Jarke, C. Zaniolo (Eds.), Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94), Morgan Kaufmann, 1994, pp. 487–499.
17. S. L. Lauritzen, D. J. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems, J. Royal Statistics Society B 50 (2) (1988) 157–194.
18. Norsys, Netica, `http://www.norsys.com` (2004).
19. I. Beinlich, H. Suermondt, R. Chavez, G. Cooper, The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks, in: J. Bocca, M. Jarke, C. Zaniolo (Eds.), Proceedings of the Second European Conference on Artificial Intelligence in Medicine (AIME 89), Springer, 1989, pp. 247–256.
20. B. Boerlage, Link strength in bayesian networks, Master's thesis, Dept. Computer Science, Univ. of British Columbia (1992).
21. E. Lamma, F. Riguzzi, A. Stambazzi, S. Storari, Improving the SLA algorithm using association rules, in: A. Cappelli, F. Turini (Eds.), Ottavo Congresso Nazionale dell'Associazione Italiana per l'Intelligenza Artificiale (AI*IA2003), Pisa, 23-26 September 2003, no. 2829 in LNAI, Springer Verlag, Heidelberg, Germany, 2003, pp. 165–175.