Exploiting Association and Correlation Rules Parameters for Improving the K2 Algorithm

Evelina Lamma[1] and Fabrizio Riguzzi[1] and Sergio Storari[1,2]

Abstract. A Bayesian network is an appropriate tool to deal with the uncertainty that is typical of real-life applications. Bayesian network arcs represent statistical dependence between different variables. In the data mining field, association and correlation rules can be interpreted as well as expressing statistical dependence relations. K2 is a well-known algorithm which is able to learn Bayesian networks. In this paper we present two extensions of K2 called K2-Lift and K2-X^2 that exploit two parameters normally defined in relation to association and correlation rules. The experiments performed show that K2-Lift and K2-X^2 improve K2 with respect to both the quality of the learned network and the execution time.

1 INTRODUCTION

A Bayesian network is an appropriate tool for working with the uncertainty that is typical of real-life applications. A Bayesian network [17] is a directed, acyclic graph (DAG) whose nodes represent random variables. In Bayesian networks each node, V, is conditionally independent of any subset of the nodes that are not its descendants, given its parents. Bayesian networks are sometimes called causal networks because the arcs connecting the nodes can be thought of as representing direct causal relationships. Building Bayesian networks on the basis of the intuitive (human) notion of causality usually results in networks for which the implied conditional independence assumptions are appropriate. According to [13] "… to construct a Bayesian network for a given set of variables, we draw arcs from cause variables to immediate effects. In almost all cases, doing so results in a Bayesian network whose conditional-independence implications are accurate".

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

Techniques for learning Bayesian networks have been extensively investigated (see, for instance [13]). Given a training set of examples, learning such a network is the problem of finding the structure of the direct acyclic graph and the CPTs associated with each node in the DAG 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],[11],[13],[14],[15],[20],[22],[28]). 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 a greedy approach is followed.

The K2 algorithm [11] is a typical search & 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 when the addition of a single parent cannot increase the probability of the network given the data. Other search and score methods include the algorithms based on the MDL principle [28], and the CB algorithm [24].

In this work, we propose the algorithms K2-Lift and K2-X^2 that improve the quality of learned networks and reduce the computational resources needed. These algorithms exploit parameters normally defined in relation to association rules [1] and correlation rules [8] to obtain new knowledge to be used for improving K2. Association rules describe co-occurrence of events, and can be viewed as probabilistic rules. Correlation rules instead describe correlation between events. Each association or correlation 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 parameters lift and X^2 in order to improve K2.

The paper is structured as follows. Section 2 provides an introduction to Bayesian networks. Section 3 describes the K2 algorithm. In Section 4 we briefly present association and correlation rules. Section 5 illustrates the algorithms K2-Lift and K2-X^2. In Section 6 we show an experimental comparison among K2, K2-Lift and K2-X^2 considering three of the most known Bayesian networks. Finally, in Section 7, we conclude and present future work.

2 BAYESIAN NETWORKS

A Bayesian network B is defined as a pair $B = (G,GPr)$, where G is a directed, acyclic graph $G = (V(G), A(G))$, witha set of nodes $V(G) = \{V_1, \ldots, V_n\}$, representing a set of stochastic variables and a set of arcs $A(G) \subseteq V(G) \times V(G)$, representing conditional and unconditional stochastic independences among the variables, modeled by absence of arcs among nodes [19],[21]. In the following, variables will be denoted by upper-case letters, e.g. V, whereas a variable V which takes on a value v , i.e. V = v, will be abbreviated to v.

1 Dipartimento di Ingegneria, Università di Ferrara, via Saragat 1, 44100, Ferrara, Italy
{elamma, friguzzi, sstorari}@ing.unife.it
2 Dipartimento di Elettronica, Informatica e Sistemistica, Università di Bologna, 40136, Viale Risorgimento 2, Bologna, Italy

When it is not necessary to refer to a specific value of a variable, we will usually just refer to a variable, which thus stands for any value of the variable.

The basic property of a Bayesian network is that any variable corresponding to a node in the graph G is (conditionally) independent of its non-descendants given its parents; this is called the local Markov property. A joint probability distribution $Pr(V_1, \ldots, V_n)$ is defined on the variables. As a consequence of the local Markov property, the following decomposition property holds:

$$\Pr(V_1,...,V_n) = \prod_{i=1}^{n} \Pr(V_i \mid \pi(V_i)) \qquad (1)$$

where $\pi(V_i)$ denotes the conjunction of variables corresponding to the parents of V_i, for i=1, …, n. A directed graph that respects all independence information in a probability distribution is called an I-map; if it respects all dependence information, it is called a D-map. A graph that is both an I-map and a D-map, is called a perfect map.

Once the network is built, probabilistic statements can be derived from it by probabilistic inference, using one of the inference algorithms described in the literature (e.g. [19],[21]).

3 THE K2 ALGORITHM

In the literature, we find different approaches for Bayesian network learning. Some of them are based on the search and score methodology [3],[11],[13],[14],[15],[20],[22],[28], and the others follow an information theory based approach [9],[24].

A frequently used procedure for Bayesian network structure construction from data is the K2 algorithm [11]. Given a database D, this algorithm searches for the Bayesian network structure G* with maximal Pr(G,D), where Pr(G,D) is determined as described below. 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_{i_k} k=1, … ,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 Bayesian network structure 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 jth unique instantiation of $\pi(V_i)$ relative to D. Suppose there are q_i such unique instantiations of $\pi(V_i)$. Define N_{ijk} to be the number of cases in D in which variable V_i has the value v_{i_k} and $\pi(V_i)$ is instantiated as w_{ij}. Let

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk} \qquad (2)$$

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

$$\Pr(G,D) = \Pr(G) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \prod_{k=1}^{r_i} N_{ijk}! \qquad (3)$$

The K2 algorithm assumes that an ordering on the variables is available and that all structures are equally likely. For every node V_i it searches for the set of parent nodes that maximizes the following function:

$$g(V_i, \pi(V_i)) = \prod_{j=1}^{q_i} \frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \prod_{k=1}^{r_i} N_{ijk}! \qquad (4)$$

K2 adopts a greedy heuristic method. It starts by assuming that a node lacks parents, after which in every step it adds incrementally the parent whose addition most 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 (i.e. the function $g(V_j,\pi(V_i))$).

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

```
1. for i=1 to n
2. {
3.   π(Vi)=0;
4.   repeat
5.   {
6.        select Vj∈ {V1, … ,Vi-1} - π(Vi) that
            maximizes g(i, π(Vi) ∪ {Vj});
7.        Δ = g(Vj, π(Vi) ∪ {Vj}) - g(Vj, π(Vi));
8.        if Δ > 0 then π(Vi) = π(Vi) ∪ {Vj};
9.   } until Δ < 0  or π(Vi) = {V1, … ,Vi-1};
10.}
```

Figure 1. Pseudo code representation of the K2 algorithm

4 ASSOCIATION AND CORRELATION RULES

Association rules [1] describe co-occurrence of events and can be regarded as probabilistic rules. Good examples from real life are databases of sales transactions. In this case the aim is to find the items that 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 [1] is a rule of the form

$$A_1=v_{A1},\ A_2=v_{A2},\ \ldots,\ A_j=v_{Aj} \Rightarrow B_1=v_{B1},\ B_2=v_{B2},\ \ldots,\ B_k=v_{Bk}$$

where $A_1, A_2, \ldots, A_j, B_1, B_2, \ldots, B_k$ are attribute names and $v_{A1}, v_{A2}, \ldots, v_{Aj}, v_{B1}, v_{B2}, \ldots, v_{Bk}$ are values such that v_{Ai} (v_{Bh}) belongs to the domain of the attribute A_i (B_h).

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, i.e. it is a set *X* such that $X \subseteq M$. 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 $X \cap Y \neq \emptyset$.

For an association rule $X \Rightarrow Y$ we define the following parameters:

- The *support* of $X \Rightarrow Y$ (indicated by $support(X \Rightarrow Y)$) is $support(X \cup Y)$.
- The *lift* [5] of $X \Rightarrow Y$ (indicated by $lift(X \Rightarrow Y)$) is given by $lift(X \Rightarrow Y) = support(X \cup Y) / (support(X) \times support(Y))$.

A *correlation rule* [8] is a set of variable names $\{A_1, A_2,..., A_n\}$. Correlation rule $\{A_1, A_2,...,A_n\}$ means that the variables A_1, A_2, ..., A_n are correlated.

With respect to correlation rules, the Pearson's X^2 statistic [8],[12] can be defined. This statistic measures the degree of correlation: if the statistic is 0, then the variables in the rule are uncorrelated. If it bigger that 0 then there is a certain degree of correlation. In the case of a rule with two variables X and Y, X^2 can be defined as follows. Let X assume I different values $x_1,...x_I$ and Y assume J different values $y_1,...y_J$. Moreover, let us define the following parameters: $n=|D|$, $n_{ij}=support(\{X=x_i,Y=y_j\})n$, $n_{i\bullet}=support(\{X=x_i\})n$, $n_{\bullet j}=support(\{Y=y_j\})n$ and $n^*_{ij}=n_{i\bullet}n_{\bullet j}/n$. X^2 is then given by

$$X^2 = \sum_{i=1}^{I}\sum_{j=1}^{J}\frac{(n_{ij}-n^*_{ij})^2}{n^*_{ij}} \qquad (5)$$

The X^2 test is based on the X^2 distribution with $(I-1)(J-1)$ degrees of freedom. The hypothesis that X and Y are uncorrelated can be rejected with a certain level of significance if X^2 is above a threshold obtained from the distribution. For example, for 1 degree of freedom (the case of binary variables) and a significance level of 95% the threshold for X^2 is 3.84. Thus, if X^2 is above 3.84, we are 95% sure that X and Y are correlated.

5 K2-LIFT AND K2-X^2 ALGORITHMS

In this section, we describe the algorithms K2-Lift and K2-X^2 that improve the K2 algorithm described in Section 3 by exploiting association and correlation rules parameters. The K2 algorithm, in order to work, requires the total ordering of the nodes. This ordering is not always simple to obtain especially in complex domains characterized by many attributes. In addition to this limitation, the algorithm has a high computational cost and produces a significant number of extra arcs in the learned network.

The high computational cost is due to function (4) (see Section 3) 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 function (4). The algorithm will add at least one arc to root nodes because the value of the heuristics for this new structure is always better than the value of the previous structure.

The new proposed approach uses the knowledge represented by association and correlation rules parameters in order to reduce the set of nodes from which the K2 algorithm tries to identify the best set of parents.

We consider only association rules with only one antecedent and only one consequent (named one-to-one rules) because they represent dependence/independence relation between two variables. Each rule is characterized by a value for the lift parameter described in Section 4.

K2-Lift is based on the following observation. In the case in which two nodes Q and P are maximally dependent then $support(Q=q_i \cup P=p_j)=support(Q=q_i)=support(P=p_j)$ and the lift for the rule $P=p_j \Rightarrow Q=q_i$ would be $1/support(P=p_j)=1/support(Q=q_i)$. In the case that P and Q are not maximally dependent then $1/support(Q=q_i) \neq 1/support(P=p_j)$. We consider in this case the average of these two values:

$$LiftMD = \frac{\frac{1}{s(P=p_j)}+\frac{1}{s(Q=q_i)}}{2} \qquad (6)$$

where $s(X=x)$ stands for $support(X=x)$. We use this parameter as a measure of the lift in the case of maximal dependency and we compare the actual lift of the rule $P=p_j \Rightarrow Q=q_i$ with this value by computing the formula

$$LiftNorm = \frac{|LiftMD-1|-|Lift-1|}{|LiftMD-1|} \qquad (7)$$

where the -1 addendum is used because we want to measure the departure of lift from the case of independence in which lift is equal to 1.

We compute the value of *LiftNorm* for all possible rules involving P and Q and we take the minimum value *MinLiftNorm* for *LiftNorm*. Then we compare this value to a threshold: if *MinLiftNorm* is greater or equal to the threshold we discard P from the possible parents of node Q. In the experiments we have used a threshold of 97%.

We compare *MinLiftNorm* with a threshold instead of comparing directly the lift of a rule with a threshold because the lift has a minimum, which is 1, but has not a fixed maximum, because its theoretical maximum is $1/support(P=p_j)=1/support(Q=q_i)$. Therefore, in order to use the lift parameter, it was necessary to consider a measure that takes into account the theoretical maximum value for it.

K2-X^2 differs from K2 because it deletes from the set of allowable parents of a node Q all those nodes P for which the X^2 statistic of the correlation rule $\{P,Q\}$ is below the threshold value given by a 95% significance.

In both cases, if *MinLiftNorm* is below the threshold for many couples of variables and if X^2 is above the threshold for many correlation rules, then K2-Lift and K2-X^2 will not remove many variables from the list of parents and the successive execution of K2 will require more time and will possibly incur in more errors.

6 EXPERIMENTAL COMPARISONS

We compared K2, K2-X^2 and K2-Lift on three different Bayesian networks:

- The "Visit to Asia" network: A belief network for a fictitious medical example about whether a patient has tuberculosis, lung cancer or bronchitis on the basis of their X-ray, dyspnea, visit-to-Asia and smoking status. It has 8 nodes and 8 arcs, and is described in [19].
- "Boelarge92": A belief network for a particular scenario of neighborhood events that shows how even distant concepts can have some connection. It has 24 nodes and 35 arcs. It is described in [6];
- The "ALARM" network: ALARM stands for "A Logical Alarm Reduction Mechanism". This is a medical diagnostic system for patient monitoring. It is a nontrivial belief network with 8 diagnoses, 16 findings and 13 intermediate variables (37 nodes and 46 arcs), and is described in [4].

The dataset of examples used for learning has been obtained with Hugin [16]. This tool, given the structure and the CPTs of a Bayesian network is able to automatically generate a dataset of N examples representing its probability relations. Each experiment was conducted by first generating a dataset from one of the above networks and then trying to learn back the network using K2, K2-X^2 and K2-Lift. For each network, two datasets of examples were generated, one with 5000 examples and another with 20000 examples, except for the ALARM network for which the datasets have 5000 and 10000 examples. This was done because the ALARM network was really computationally demanding. The learned networks are compared with the original network in Table 1. For each algorithm we indicate: the numbers of missing and extra arcs; the number of computations of the function $g(V_i,\pi(V_i))$ (Num g); the number of computation of N_{ijk} (Num N_{ijk}). The last two parameters represent the computational resources needed by the algorithms.

Analysing these experimental results we can observe that both K2-Lift and K2-X^2 improve with respect to K2 both in terms of the quality of the learned network and in terms of the used computational resources. Moreover, it must be noted that in the case of the ALARM network K2-Lift finds a network with a much higher quality with respect to both K2 and K2-X^2.

From these experiments we can observe that K2-Lift and K2-X^2 work reasonably well on medium networks where the nodes are not too interconnected. Further experiments are required in order to test the two algorithms on larger and denser networks.

We have also compared K2-Lift and K2-X^2 with the algorithm K3 [7] on the Visit to Asia network and preliminary experiments showed that our algorithms achieve an improvement as regards the quality of the resulting network.

7 CONCLUSIONS

In this work we describe two methods for improving K2 [11], one of the most known algorithm for learning Bayesian network, by exploiting association and correlation rules parameters.

The K2 algorithm 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 does not increase the probability of the resulting network given the data.

In this work, we reduce the set of allowable parents from which the algorithm selects actual parents and avoid extra arc insertions. This new methodology uses data mining techniques, and in particular the computation of association and correlation rules parameters from a database of examples, in order to learn the structure of a Bayesian network. We have presented the K2-X^2 and K2-Lift algorithms that exploit the X^2 and lift parameter of, respectively, correlation and association rules in order to improve the performance of the K2 algorithm.

Experiments discussed in the paper show that the proposed approach allows to obtain networks that have a higher quality with respect to K2 in a shorter time.

In the future, we plan to compare K2-X^2 and K2-Lift with an algorithm based on MDL [28] and other Bayesian network learning algorithms.

ACKNOWLEDGEMENTS

.This work was partially funded by the Information Society Technologies programme of the European Commission under the IST-2001-32530 SOCS project in the context of the Global Computing initiative, and by the Ministero dell'Istruzione,della Ricerca e dell'Università under the COFIN2003 project "La gestione e la negoziazione automatica dei diritti sulle opere dell'ingegno digitali: aspetti giuridici e informatici". Fabrizio Riguzzi and Sergio Storari would like to thank ECCAI for a travel award. The authors would like to thank Rossella Bozzini for her help with the experiments.

REFERENCES

[1] Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Buneman, P., Jajodia, S. (eds.): Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993. ACM Press 22(2) (1993) 207-216.

[2] Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.): Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94). Morgan Kaufmann, ISBN 1-55860-153-8 (1994) 487-499.

[3] Akaike, H.: A new Look at Statistical Model Identification. IEEE Trans. Automatic Control 19 (1974) 716-723.

[4] Beinlich, I.A., Suermondt, H.J., Chavez, R.M., Cooper, G.F.: The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In: Hunter, J. (ed.): Proceedings of the Second European

Table 1. Comparison between K2, K2-X^2 and K2-Lift.

Data Set	K2				K2-X^2				K2-Lift			
	Missed Arcs	Extra Arcs	Num g	Num. N_{ijk}	Missed Arcs	Extra Arcs	Num g	Num. N_{ijk}	Missed Arcs	Extra Arcs	Num g	Num. N_{ijk}
Asia 5000	0	2	61	1608	1	1	37	816	1	0	34	732
Asia 20000	0	1	57	1224	0	0	47	1020	0	0	28	588
Boelarge92 5000	7	2	585	14244	6	0	234	5316	6	0	226	5064
Boelarge92 20000	7	4	615	17172	6	0	314	7620	6	0	196	4476
Alarm 5000	1	11	1771	204462	1	7	853	124098	1	0	725	110397
Alarm 10000	1	11	1771	204462	1	10	964	146694	1	0	701	103683

Conference on Artificial Intelligence in Medicine (AIME 89). Springer, Berlin (1989) 247-256.
[5] Berry, J.A., Linoff, G.S.: Data Mining Techniques for Marketing, Sales and Customer Support. John Wiley & Sons Inc., New York (1997).
[6] Boerlage, Brent: Link Strength in Bayesian Networks. MSc Thesis, Dept. Computer Science, Univ. of British Columbia, BC (1992).
[7] Bouckaert, R.: Belief networks construction using the minimum description length principle. In: Clarke, M., Kruse, R., Moral, S. (ed.)., Symbolic and Quantitative Approaches to Reasoning and Uncertainty, Lecture Notes in Computer Science 747, Springer Verlag, Berlin, (1993), 41-48.
[8] Brin, S., Motwani, R., Silverstein, C.: Beyond market baskets: Generalizing association rule to correlations. In: Proceedings ACM SIGMOD International Conference on Management of Data. SIGMOD (1997) 265-276.
[9] Cheng, J., Greiner, R., Kelly, J., Bell, D., Liu, W.: Learning Bayesian networks from data: An information-theory based approach, Artificial Intelligence 137 (2002) 43–90.
[10] Chow, C.K., Liu, C.N.: Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory 14 (1968) 462-467
[11] Cooper, G., Herskovits, E.: A Bayesian method for the induction of probabilistic networks from data. Machine Learning 9 (1992) 309-347.
[12] Giudici, P.: Data mining - Metodi statistici per le applicazioni aziendali. McGraw-Hill, Milano (2001).
[13] Heckerman, D., Geiger, D., Chickering, D.: Learning Bayesian Networks: the combination of knowlegde and statistical data. Machine Learning 20 (1995) 197-243.
[14] Heckerman, D.: Tutorial on learning in Bayesian networks. In: Jordan, M. (ed.): Learning in Graphical Models. MIT Press, Cambridge, MA (1999).
[15] Herskovits, E.H.: Computer-based probabilistic-network construction. Doctoral Dissertation, Medical Informatics, Stanford University (1991).
[16] Hugin. http://www.hugin.com.
[17] Kim, J. H., Pearl, J.: 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) 190-193.
[18] Lamma, E., Riguzzi, F., Stambazzi, A., Storari, S.: Improving the SLA algorithm using association rules. Italian Congress on Artificial Intelligence (AIIA 2003), (2003).
[19] Lauritzen, S.L., Spiegelhalter D.J.: Local computations with probabilities on graphical structures and their application to expert systems. J. Royal Statistics Society B 50(2) (1988) 157-194.
[20] Madigan, D., Raftery A.:. Model Selection and Accounting for Model Uncertainty in Graphical Models Using Occam's Window. J. Am. Statist. Association 89 (1994) 1535-1546.
[21] Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Francisco (1988).
[22] Rissanen, J.: Stochastic Complexity (with discussion). J. Roy. Statist. Soc. B 49 (1987) 223-239.
[23] Schwarz, G.: Estimating the Dimension of a Model. Annals of Statistics 6 (1978) 461-464.
[24] Sigh, M., Valtorta, M.: Construction of Bayesian Network Structures from Data: a Brief Survey and an Efficient Algorithm. International Journal of Approximate Reasoning 12 (1995), 111-131.
[25] Silverstein, C., Brin, S., Motwani, R., Ullman, J.D.: Scalable Techniques for Mining Causal Structures. Data Mining and Knowledge Discovery 4(2/3) (2000) 163-192.
[26] Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. Lecture Notes in Statistics. Springer, Berlin (1993).
[27] Spirtes, P., Glymour, C., Scheines, R.: An algorithm for fast recovery of sparse causal graphs. Social Science Computer Review 9 (1991) 62-72.
[28] Suzuki, J.: 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 (1999).