Exploiting Association and Correlation Rules Parameters for Learning Bayesian Networks

Sergio Storari Fabrizio Riguzzi[*]
Evelina Lamma
ENDIF-Dipartimento di Ingegneria,
Università di Ferrara, via Saragat 1
44100, Ferrara, ITALY
fabrizio.riguzzi@unife.it
sergio.storari@unife.it
evelina.lamma@unife.it

November 26, 2008

Abstract

In data mining, association and correlation rules are inferred from data in order to highlight statistical dependencies among attributes. The metrics defined for evaluating these rules can be exploited to score relationships between attributes in Bayesian network learning. In this paper, we propose two novel methods for learning Bayesian networks from data that are based on the K2 learning algorithm and that improve it by exploiting parameters normally defined for association and correlation rules. In particular, we propose the algorithms K2-Lift and K2-X^2, that exploit the lift metric and the X^2 metric respectively. We compare K2-Lift, K2-X^2 with K2 on artificial data and on three test Bayesian networks. The experiments show that both our algorithms improve K2 with respect to the quality of the learned network. Moreover, a comparison of K2-Lift and K2-X^2 with a genetic algorithm approach on two benchmark networks show superior results on one network and comparable results on the other.

[*] Corresponding author, Tel. +390532974836, Fax +390532974870

Keywords Bayesian Networks Learning, K2, Association Rules, Correlation Rules

1 Introduction

Bayesian networks are very effective tools for representing uncertain knowledge and performing reasoning on it. However, building a Bayesian network for a domain is a time consuming and difficult task. Therefore, techniques for automatically inferring a Bayesian network from data have recently received a lot of attention [3, 32, 25, 18, 11, 17, 34, 17, 23, 16, 35, 10, 7, 23]. Given a training set of examples, learning a Bayesian network is the problem of finding the structure of the network together with the conditional probability tables (CPTs for short) that best match the dataset. The quality of the match is evaluated using a scoring metric such as description length or posterior probability [3, 32, 25, 18, 11, 17, 16, 35]. Usually a greedy search in the space of possible structures is adopted.

Among the several different approaches for learning Bayesian networks, K2 [11] is one of the fastest: it takes as input a topological sort of the nodes and, for each node, it repeatedly adds a previous node as a parent if the resulting structure increases a score given by the joint probability of the data and the network structure. K2 stops adding parents when no addition can increase the score.

[29] shows that K2 has some problems in dealing with root nodes (i.e., nodes without parents) as it erroneously generates many extra arcs pointing to them. One way to avoid this problem is to identify all root nodes before starting learning.

In order to reach this goal, we propose the use of parameters normally defined in relation to association and correlation rules. In data mining, association rules [2] and correlation rules [9] are used for representing dependencies among variables and are automatically inferred from data. Each association or correlation rule is characterized by a number of parameters which can be used to identify independence among the nodes.

In this paper we present two algorithms that use these parameters to improve the quality of the networks learned by K2 and to further reduce the computational resources needed:

- K2-Lift exploits the lift parameter in order to improve K2;
- K2-X^2 exploits Pearson's X^2 index in order to improve K2;

This paper summarizes the techniques described in [20, 21] and [13] and presents new experimental results.

The paper is structured as follows. Section 2 provides an introduction to the problem of learning Bayesian networks. In Section 3 we present association and correlation rules. Section 4 describes the algorithms K2-Lift and K2-X^2. In Section 5 we show experimental comparisons among K2, K2-Lift and K2-X^2. In Section 6 we present some related works and discuss experiments comparing K2-Lift and K2-X^2 with the genetic algorithms of [23]. Finally, in Section 7, we conclude and present future work.

2 Learning Bayesian Networks

Given a set of discrete random variables $\mathcal{V}$, a Bayesian network represents probabilistic dependencies among the variables of $\mathcal{V}$. Formally, a Bayesian network is a couple $(\mathcal{G}, \Theta)$ where $\mathcal{G}$ is a directed acyclic graph with a node per variable and Θ is a set of parameters expressing the dependency of a variable from its parents in the graph.

A Bayesian network can be built by interviewing a domain expert. In the case in which no expert is available and a set of observations regarding the domain variables is available, Bayesian network learning algorithms can be used to infer the parameters of the network, the structure or both.

A widely used approach for learning Bayesian networks consists in performing local search in the space of possible structures guided by a scoring function. Usually, local search is performed by starting from a user defined structure (possibly empty) and by repeatedly adding, removing or reversing an arc. The new structure is then scored and the best modification is kept.

Various scoring functions have been proposed in the literature, based on different principles: Bayesian inference [11, 17, 31], entropy [18, 14], minimum description length [19, 35] and minimum message length [37].

The K2 algorithm [11] is a search and score algorithm that uses a Bayesian scoring function. Given a database D, K2 searches for the structure $\mathcal{G}$ that maximizes the joint probability of the data and the structure $P(D, \mathcal{G})$. In [11] the joint probability is computed in closed form given a dataset provided that a number of assumptions hold:

1. the examples of the dataset are independent, identically distributed and complete;
2. the parameters for different configurations of the parents of a node are

independent given the structure;

3. the prior distribution of the parameters for a parent configuration given the structure is uniform.

In order to present the scoring function, we need the following definitions. Variable V_i has r_i possible value assignments v_{ik} for $k = 1, \ldots, r_i$. Let D be a database of m cases. Each node $V_i \in \mathcal{V}$ 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 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_{ik} and $\pi(V_i)$ is instantiated as w_{ij}. Let

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

be the number of cases in which $\pi(V_i)$ take value w_{ij}.

Given a Bayesian network model, from the assumption 1, 2 and 3 it follows that

$$P\left(\mathcal{G}, D\right) = P\left(\mathcal{G}\right) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{2}$$

This function can be computed with one scan over the data in which the sufficient statistics N_{ijk} are computed. By defining

$$g\left(V_i, \pi\left(V_i\right)\right) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{3}$$

we can write

$$P\left(\mathcal{G}, D\right) = P\left(\mathcal{G}\right) \prod_{i=1}^{n} g\left(V_i, \pi\left(V_i\right)\right) \tag{4}$$

In this formula, the individual variables provide independent contributions, so the score can be optimized variable by variable.

The K2 algorithm assumes that an ordering on the variables is available and that all structures are a priori equally likely. For every node V_i, it searches for the set of parent nodes $\pi(V_i)$ that maximizes the function $g\left(V_i, \pi\left(V_i\right)\right)$

K2 adopts a greedy heuristic method. It starts by assuming that a node has no parents, and then, at every step, it adds the parent whose addition mostly increases the function $g(V_i, \pi(V_i))$. K2 stops adding parents to the nodes when the addition of a single parent does not increase $g(V_i, \pi(V_i)))$. A pseudo code representation of K2 algorithm is shown in Figure 1.

[29] observed that, under particular conditions, the K2 algorithm introduces learning errors adding many extra arcs between root nodes. The article points out that these errors may be reduced by performing an analysis on the dataset aimed at identifying the root nodes before starting K2 learning.

3 Association and Correlation Rules

Association rules [2] relate events that are frequently observed together. 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.

An *item* is a literal of the form $V = v$ where V is a variable of the domain (attribute of the dataset) and v is a valule that belongs to the domain of V. Let M be the set of all the possible items. An *itemset* X is a consistent set of items, that is a set X such that $X \subseteq M$ and $V = v_1 \in X, V = v_2 \in X \Rightarrow v_1 = v_2$.

A *transaction* T is a record of the database. 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 $\mathit{Support}(X)$) is the fraction of transactions in D that contain X. The support of the opposite of an itemset X (indicated by $\mathit{Support}(!X)$) is the fraction of transactions in D that do not contain X. Thus, $\mathit{Support}(!X) = 1 - \mathit{Support}(X)$.

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

- The *support* of $X \Rightarrow Y$ (represented by $\mathit{Support}(X \Rightarrow Y)$) is defined as $\mathit{Support}(X \cup Y)$;
- The *lift* [6] of $X \Rightarrow Y$ (represented by $\mathit{Lift}(X \Rightarrow Y)$) is defined as $\mathit{Support}(X \cup Y)/(\mathit{Support}(X) \times \mathit{Support}(Y))$;
- The *leverage* [30] of $X \Rightarrow Y$ (represented by $\mathit{leverage}(X \Rightarrow Y)$) is defined as $\mathit{Support}(X \cup Y) - \mathit{Support}(X) \times \mathit{Support}(Y)$.

A *correlation rule* [9] is a set of variables $\{V_1, \ldots, V_m\}$. The Pearson's X^2 statistic [9] can be defined with respect to a correlation rule. This statistic measures the degree of correlation among the variables: if the statistic is 0, then the variables in the rule are uncorrelated. If it is bigger than 0, then there is a certain degree of correlation. In the case of a rule with two variables P and Q, X^2 can be defined as follows. Suppose P assumes I different values $p_1, \ldots, p_I$ and suppose Q assumes J different values $q_1, \ldots, q_J$. Moreover, let us define the following parameters: $N = |D|$, $N_{ij} = \mathit{Support}(\{p = p_i, Q =$

$q_j\}) \times N$, $N_{i\bullet} = Support(\{P = p_i\}) \times N$, $N_{\bullet j} = Support(\{Q = q_j\}) \times N$ and $N^*_{ij} = N_{i\bullet} \times N_{\bullet j}/N$. X^2 is then given by

$$X^2 = \sum_{i=1}^{I} \sum_{j=1}^{J} \frac{\left(N_{ij} - N^*_{ij}\right)^2}{N^*_{ij}} \tag{5}$$

N^*_{ij} can be interpreted as the number of records of D that are expected to have $P = p_i$ and $Q = q_j$ given that P and Q are independent while N_{ij} is the actual number of such records. Thus X^2 measures the difference between the expected number of such records in the case that P and Q are independent and the actual number of records. The X^2 test is based on the χ^2 distribution with $(I-1)(J-1)$ degrees of freedom. The hypothesis that P and Q 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 most common significance level) the threshold for X^2 is 3.84. Thus, if X^2 is above 3.84, we are 95% sure that P and Q are correlated.

4 Proposed Algorithms

This section describes how the K2 learning algorithm has been improved by exploiting parameters defined in relation to association and correlation rules. On the basis of these parameters, the set of nodes from which the K2 algorithm tries to identify the best set of parents is reduced, thus mitigating the problem of extra arcs discussed in [29].

We consider only binary association rules, with one item in the body and one item in the head. Each rule is characterized by a value for the lift parameter described in Section 3.

4.1 K2-Lift

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

$$\mathit{LiftMD}_{ij} = \frac{\frac{1}{\mathit{Support}(\{P=p_i\})} + \frac{1}{\mathit{Support}(\{Q=q_j\})}}{2} \tag{6}$$

We use this parameter (LiftMD) as a measure of the lift in the case of Maximal Dependency, and we compare the actual lift Lift_{ij} of the rule $P = p_j \Rightarrow Q = q_i$ with this value by computing the formula

$$\mathit{LiftNorm}_{ij} = \frac{\mathit{Lift} - 1}{\mathit{LiftMD}_{ij} - 1} \tag{7}$$

where the -1 term is used because we want to measure the departure of lift from the case of independence in which lift is equal to 1. We use the normalized version of the lift because in this way we can compare $\mathit{LiftNorm}_{ij}$ for all possible values p_i and q_j of P and Q. Let $\mathit{MaxLiftNorm}$ be

$$\max_{ij}\{\mathit{Lift}_{ij}\}$$

We then compare $\mathit{MaxLiftNorm}$ to a threshold: if $\mathit{MaxLiftNorm}$ is greater than or equal to the threshold, we add P to the possible parents of node Q as we cannot exclude a possible correlation between P and Q.

4.2 K2-X^2

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 for the correlation rule $\{P, Q\}$ is below the threshold value given by a 95% significance.

In both cases, if *MaxLiftNorm* is above 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 execution will require more time and will possibly incur in more errors.

5 Experimental Comparisons

In order to evaluate the new algorithms, we selected a number of Bayesian networks, we generated datasets from them by random sampling, we applied K2, K2-Lift and K2-X^2 and then we compared the learned networks. The comparison is performed by computing the number of extra arcs (EA for short in the following), i.e. arcs that are present in the learned network but absent from the original network, and missing arcs (MA for short in the following), i.e. arcs that are present in the original network but absent from the learned network. Note that in computing EA and MA we took into account the directionality of the arcs, therefore if the original network contains an arc from P to Q and the learned network contains an arc from Q to P this is counted as one missing and one extra arcs. As a measure of performance of an algorithm we used the total number of wrong arcs (WA for short in the following) computed as $WA = EA + MA$ (also called Hamming distance in [23]).

We considered both artificially generated and benchmark Bayesian networks.

5.1 Artificially Generated Networks

We generated a number of networks by using a random procedure inspired to the one of [33]. The procedure is shown in Figure 2 and takes as input the number of nodes n, the maximum number of parents MP and the probability p that a node is a parent.

The CPT of a network was generated by filling each cell of the table with a random number between 0 and 1 and by dividing each cell of a row corresponding to a combination of parent values by the sum of the row values.

A number of networks were generated by varying the parameters: n was set to 10, 15 and 20 while MP and p assume the values from the following configurations:

1. $MP = 0$, $p = 0$ (empty DAG for evaluating the EA), or

2. $MP = 2$, $p = 0.25$, or

3. $MP = 3$, $p = 0.5$

For each combination of n, MP and p we generated 20 networks. So, overall, 180 networks were generated.

A number of datasets were sampled with the following number of cases: 1000, 5000, 10000 and 20000. For each size and each network, we performed an experiment using 10 randomly generated datasets. For each dataset, one random sort of the attributes was given as input to each learning algorithm.

Therefore, we generated overall 7.200 datasets and 21.600 learned networks. For each learned network we computed WA.

For each combination of network parameters, dataset size and couple of algorithms we applied the Student's t two-tailed test: we computed the value of the t statistics for the WA value over the 10 dataset - ordering couples. We declare two algorithm equivalent in an experiment if the null hypothesis can not be rejected with a 97.5% significance, otherwise we identify a winner and a loser.

Table 1, Table 2 and Table 3 report the number of wins, ties and losses for networks with 10, 15 and 20 attributes respectively.

As can be seen from the tables, K2-Lift and K2-X^2 never lose against K2.

Comparing in more details K2-X^2 with K2, we notice that the performance improvement of K2-X^2 is relevant for small datasets and decreases when the dataset dimension increases. The improvement increases with the number of network attributes (e.g., for 5000 cases, K2-X^2 wins on 3 datasets for 10 attributes, it wins on 7 dataset for 15 attributes and it wins on 12 datasets for 20 attributes).

Comparing in more details K2-Lift with K2, we notice that the performance improvement of K2-Lift is not relevant for small datasets, gets larger for medium datasets (e.g., 5000 cases) and decreases again for large datasets. As for K2-X^2, the improvement relevance increases with the number of network attributes (e.g., for 5000 cases, K2-Lift wins on 3 datasets for 10 attributes, it wins on 6 datasets for 15 attributes and in wins on 12 datasets for 20 attributes).

Comparing K2-Lift with K2-X^2, we notice that K2-X^2 is better for small datasets while K2-Lift is better for large datasets, especially for complex networks.

5.2 Benchmark Networks

We have considered three different benchmark networks:

- "Asia": a network for a fictitious medical example in which 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. It is shown in Figure 3 and is described in [24].
- "Alarm": a medical diagnostic network for patient monitoring. It is a nontrivial belief network with 8 diagnoses, 16 findings and 13 intermediate variables (36 nodes and 46 arcs). It is described in [5];
- "Boelarge92": a 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 [8];

From these networks we generated datasets by random sampling with the following sizes: 5000 and 20000 for "Asia" and "Boelarge92", 5000 and 10000 for "Alarm". A smaller upper size was chosen for "Alarm" because for 20000 cases the learning algorithms exhausted the memory. For each dataset size, 10 datasets were generated.

Table 4 shows the average EA and MA for the three algorithms, while Table 5 shows the values of the t statistics. For a 99% significance the two-tailed threshold is 2.756, so K2-Lift and K2-X^2 are always significantly superior to K2, while K2-Lift is always signficantly superior to K2-X^2.

5.3 Bioinformatics

In [13] K2 and K2-Lift were applied to the problem of learning genetic networks starting from microarray datasets based on experiments performed on Acute Myeloid Leukemia. The analyzed dataset, described in [28], is available on-line in the ArrayExpress repository of the European Bioinformatics Institute[1]. The dataset groups the results of 20 microarray experiments, divided as follows:

10 Acute Myeloid Leukemia (AML) samples;

10 MyeloDysplastic Syndrome (MDS) samples.

AML may develop de novo or secondarily to MDS. Large-scale profiling of gene expression by DNA microarray analysis is a promising approach for identifying genetic markers specific to de novo or MDS-related AML.

Given a dataset, the analysis protocol followed in the experiments in [13] consists of 3 steps:

1. Generate a set of 20 random attribute orderings named SAO_i, with $i = 1, .., 20$.
2. For each learning algorithm $La \in \{K2, K2 - Lift\}$:
   (a) For i=1,..,20
      i. Learn the Bayesian network $BN_{La,i}$ by using La on SAO_i

[1] `http://www.ebi.ac.uk/arrayexpress/`, access code `E-MEXP-25`

  ii. Compute the Bayes score $BS_{La,i}$ of $BN_{La,i}$

(b) Rank the learned network $BN_{La,i}$ according to their score $BS_{La,i}$

(c) Analyze the first five learned networks $BN_{La,i}$ and identify the genes that are frequently parents of other genes

Results in [13] show that, in most cases, K2-Lift creates a more synthetic network than K2.

K2-Lift identified a number of genes that are frequent parents, i.e. that have a strong influence on other genes. The strong influence of these genes is confirmed by biological literature on studies performed on AML:

- MDM2 was found by K2-Lift as a frequent parent. MDM2 is a target gene of the transcription factor tumor protein p53. Over-expression of this gene can result in excessive inactivation of tumor protein p53, diminishing its tumor suppressor function. Faderl et al [12] showed that over-expression of MDM2 is common in AML and is associated with shorter complete remission duration and event free survival rate.

- TOB1 was found by K2-Lift as a frequent parent. TOB1 is strictly related to ERBB2 which is a receptor protein tyrosine kinase frequently mutated in human cancer. The protein-kinase family is the most frequently mutated family found in human cancer and faulty kinase enzymes are being investigated as promising targets for the design of anti-tumor therapies. Zhou et al [38] showed that ERBB2-mediated resistance to DNA-damaging agents requires the activation of Akt, which enhances MDM2-mediated ubiquitination and degradation of TP53.

- HGF was found by K2-Lift as a frequent parent. HGF has been widely implicated in tumor scattering and invasive growth and is of prognostic importance in AML [36].

6 Related Works

Parameters of correlation rules are used in learning networks of binary variables also in [15]: the support of the rules is used in order to select subsets of variables with cardinality ≥ 2. These subset are built starting from cardinality 2 and are stored in a data structure called Edgedump if they represent interactions among the variables that are not described by lower order rules. To this purpose, a local search for a network structure is performed among the variables of each subset X using the BDeu score: if m is the cardinality of the subset and the resulting structure has a node with $m-1$ parents, then X is added to the Edgedump.

After having built the Edgedump, edges are progressively added to an empty network starting from those that appear in the highest number of m-way interactions. The addition of edges stop when no improvement of BDeu is obtained. The authors show that this algorithm, called Screen-based Bayes Net Structure search (SBNS) is able to achieve good performances also in large networks. However, SBNS requires the variables to be binary, while we make no such assumption.

In [1] the authors propose an algorithm called BENEDICT for learning Bayesian networks that exploits the Kullback-Leibler cross entropy as a measure of independence between nodes. The algorithm searches the space of possible networks by means of local search and, at each step, it computes the global discrepancy between the current network and the data. The global discrepancy is simply the sum of the independence measure over all the couples of non-adjacent nodes. The search stops when the discrepancy is below a user-defined threshold or when the discrepancy improvement is below a user-defined threshold. As an optimization, BENEDICT eliminates a node from the set of possible parents of another node if the independence measure between the two nodes is close to zero. In this aspect, BENEDICT is similar to K2-Lift and K2-X^2.

In [23] the authors propose an approach for learning Bayesian networks that is based on genetic algorithms (GA for short in the following). The approach is tested on datasets generated by random sampling from the Asia and Alarm networks containing 500, 1000, 2000 and 3000 cases. In order to compare K2-Lift and K2-X^2 with the algorithm in [23], we applied them to datasets of the same size obtained by random sampling from the above networks. For each algorithm and each dataset size, we performed 30 learning experiments, each time providing a different random variable order to the algorithms. The average numbers of wrong arcs (WA) for these experiments are shown in Table 6. Comparing these results with those of [23] we can observe that, for the Asia network, K2-Lift and K2-X^2 obtained a lower number of wrong arcs with respect to the best GA approach for all dataset size apart from 500. As regards the Alarm network, K2-X^2 is always superior to K2-Lift, while for the GA approach the best algorithms are “hybrid GA with the simple reduction criterion, population size

50, low mutation and crossover rate" (GA1) and "hybrid GA with the elitist reduction criterion, population size 50, low mutation rate and high crossover rate" (GA2). With respect to GA1, K2-X^2 achieves lower WA in two out of four cases and, with respect to GA2, in one out of four cases.

In the future, we plan to investigate the application of the techniques presented for ruling out parents within genetic algorithms. In particular, we plan to modify the generation of the new population by excluding the network structures that contain parents ruled out by our heuristics. This approach can be applied both to classical genetic algorithms, such as those presented in [23], and to the class of Estimation of Distribution Algorithms [22, 27] where the new population is generated by sampling from a probability distribution that is estimated from the selected individuals. Univariate Marginal Distribution Algorithm [26] and Population-Based Incremental Learning [4] are two algorithms in this class to which we plan to apply our approach, since they have been shown experimentally [7] to perform better than traditional genetic algorithms when learning Bayesian networks.

7 Conclusions

In this work we have described a method for improving the Bayesian network learning algorithm K2 by exploiting a number of parameters of association and correlation rules.

Our method improves K2 performance by reducing the set of allowable parents from which the algorithm selects the actual parents. We have presented the K2-Lift and K2-X^2 algorithms that exploit the Lift and X^2 parameters of, respectively, association and correlation rules.

We have compared the algorithms on a number of randomly generated networks and on three benchmark networks. On the randomly generated networks K2-Lift and K2-X^2 never lose against K2 and are often significantly superior. The improvement increases with the number of network attributes. As regards the dependence on the dataset size, K2-X^2 improvement is highest for small numbers of examples, while K2-Lift improvement is highest for medium size networks. Overall, K2-X^2 is best for small datasets while K2-Lift is best for large datasets, especially with a high number of attributes.

On the benchmark networks, K2-Lift and K2-X^2 are always significantly superior to K2 and K2-Lift is always significantly superior to K2-X^2.

Experiments conducted on a real biological dataset, described in detail in [13], show that, in most cases, K2-Lift creates a more synthetic network than K2.

We have also compared K2-Lift and K2-X^2 with the genetic algorithm approach proposed in [23]: on the Asia network, K2-Lift and K2-X^2 obtained a lower number of wrong arcs for three dataset dimensions out of four with respect to the best GA parametrization, while on the Alarm network K2-X^2 improves parametrization "hybrid GA with the simple reduction criterion, population size 50, low mutation and crossover rate" in two cases out of four and parametrization "hybrid GA with the elitist reduction criterion, population size 50, low mutation rate and high crossover" in one case out of four.

In the future we plan to investigate the contribution that association and correlation rules can provide to learning algorithms based on genetic search.

8 Acknowledgments

This work has been partially funded by the FAR projects of the University of Ferrara. The authors would like to thank Francesco Marsilla for his help with the experiments.

References

[1] S. Acid and L. M. de Campos. BENEDICT: An algorithm for learning probabilistic belief networks. In *Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU96)*, 1996.

[2] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, *ACM SIGMOD International Conference on Management of Data (SIGMOD93)*, pages 207–216. ACM Press, 1993.

[3] H. Akaike. A new look at statistical model identification. *IEEE Trans. Automatic Control*, 19:716–723, 1974.

[4] S. Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical report, School of Computer Science, Carnegie Mellon University, 1994.

[5] I.A. Beinlich, H.J. Suermondt, R.M. Chavez, and G.F. Cooper. The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In J.B. Bocca, M. Jarke, and C. Zaniolo, editors, *Second European Conference on Artificial Intelligence in Medicine (AIME 89)*, pages 247–256. Springer, 1989.

[6] J.A. Berry and G.S. Linoff. *Data Mining Techniques for Marketing, Sales and Customer Support.* John Wiley & Sons Inc., New York, 1997.

[7] R. Blanco, I. Inza, and P. Larrañaga. Learning bayesian networks in the space of structures by estimation of distribution algorithms. *Int. J. Intell. Syst.*, 18(2):205–220, 2003.

[8] B. Boerlage. Link strength in bayesian networks. Master's thesis, Dept. Computer Science, Univ. of British Columbia, 1992.

[9] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rule to correlations. In J. Peckham, editor, *ACM SIGMOD International Conference on Management of Data (SIGMOD97)*, pages 265–276, 1997.

[10] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu. Learning bayesian networks from data: An information-theory based approach. *Artif. Intell.*, 137(1–2):43–90, 2002.

[11] G. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Mach. Learn.*, 9:309–347, 1992.

[12] S. Faderl, H. M. Kantarjian, E. Estey, T. Manshouri, C. Y. Chan, A. Rahman Elsaied, S. M. Kornblau, J. Cortes, A. Thomas, S. Pierce, M. J. Keating, Z. Estrov, and M. Albitar. The prognostic significance of p16(ink4a)/p14(arf) locus deletion and mdm-2 protein expression in adult acute myelogenous leukemia. *Cancer*, 89(9):1976–82, 2000.

[13] G. Gamberoni, E. Lamma, F. Riguzzi, S. Storari, and S. Volinia. Bayesian networks learning for gene expression datasets. In *Advances in Intelligent Data Analysis VI: 6th International Symposium on Intelligent Data Analysis, (IDA05)*, number 3646 in Lecture Notes in Computer Science, pages 109–120, Heidelberg, Germany, September 2005. Springer Verlag.

[14] D. Geiger. An entropy-based learning algorithm of bayesian conditional trees. In *6th Annual Conference on Uncertainty in Artificial Intelligence (UAI92)*, pages 92–97, 1992.

[15] A. Goldenberg and A. Moore. Tractable learning of large bayes net structures from sparse data. In *21st International Conference on Machine Learning (ICML04)*, volume 69 of *ACM International Conference Proceeding Series*. ACM, 2004.

[16] D. Heckerman. Tutorial on learning in bayesian networks. In M. Jordan, editor, *Learning in Graphical Models*. MIT Press, Cambridge, MA, 1999.

[17] D. Heckerman, D. Geiger, and D. Chickering. Learning bayesian networks: the combination of knowlegde and statistical data. *Mach. Learn.*, 20:197–243, 1995.

[18] E. H. Herskovits. *Computer-based probabilistic-network construction.* PhD thesis, Medical Informatics, Stanford University, 1991.

[19] W. Lam and F. Bacchus. Learning bayesian belief networks: An approach based on the mdl principle. *Comput. Intell.*, 10(4):269–293, 1994.

[20] E. Lamma, F. Riguzzi, and S. Storari. Exploiting association and correlation rules parameters for improving the k2 algorithm. In Ramon Lopez de Mantaras and Lorenza Saitta, editors, *16th European Conference on Artificial Intelligence (ECAI04)*, pages 500–504, Amsterdam, Holland, August 2004. IOS Press.

[21] E. Lamma, F. Riguzzi, and S. Storari. Improving the k2 algorithm using association rules parameters. In B. Bouchon-Meunier, G. Coletti, and R. R. Yager, editors, *Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU04)*, pages 1667–1674, Roma, Italy, July 2004. Editrice Università La Sapienza.

[22] P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, volume 2 of *Genetic Algorithms and Evolutionary Computation.* Springer.

[23] P. Larrañaga, M. Poza, Y. Yurramendi, R. H. Murga, and C. M. H. Kuijpers. Structure learning of bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(9):912–926, 1996.

[24] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *J. Royal Statist. Soc. B*, 50(2):157–194, 1988.

[25] D. Madigan and A. Raftery. Model selection and accounting for model uncertainty in graphical models using occam's window. *J. Am. Statist. Association*, 89:1535–1546, 1994.

[26] H. Mühlenbein. The equation for response to selection and its use for prediction. *Evol. Comput.*, 5:303–346, 1998.

[27] H. Mühlenbein and G. Paass. From recombination of genes to the estimation of distributions i. binary parameters. In *Parallel Problem Solving from Nature - PPSN IV, International Conference on Evolutionary Computation*, volume 1141 of *Lecture Notes in Computer Science*, pages 178–187. Springer, 1996.

[28] Y. Oshima, M. Ueda, Y. Yamashita, Y. L. Choi, J. Ota, S. Ueno, R. Ohki, R. Koinuma, T. Wada, K. Ozawa, A. Fujimura, and H. Mano. Dna microarray analysis of hematopoietic stem cell-like fractions from individuals with the m2 subtype of acute myeloid leukemia. *Leukemia*, 17(10):1900–1997, 2003.

[29] Y. Peng, Z. Zhou, and S. Cho. Constructing belief networks from realistic data. *Int. J. of Intell. Syst.*, 14(7):671–695, 1999.

[30] G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, pages 229–248. AAAI/MIT Press, 1991.

[31] M. Ramoni and P. Sebastiani. Robust learning with missing data. Technical Report KMI-TR-28, Knowledge Media Institute, The Open University, 1996.

[32] J. Rissanen. Stochastic complexity (with discussion). *J. Roy. Statist. Soc. B*, 49:223–239, 1987.

[33] P. Shaughnessy and G. Livingston. Evaluating the causal explanatory value of bayesian network structure learning algorithms. Technical Report WS-06-06, University of Massuchussets Lowell, 2006.

[34] M. Singh and M. Valtorta. Construction of bayesian network structures from data: a brief survey and an efficient algorithm. *Int. J. of Approx. Reas.*, 12:111–131, 1995.

[35] J. Suzuki. Learning bayesian belief networks based on the mdl principle: An efficient algorithm using the branch and bound technique. *IEICE Trans. on Comm. Elec. Inform. and Syst.*, 1999.

[36] S. Verstovsek, H. Kantarjian, E. Estey, A. Aguayo, F. J. Giles, T. Manshouri, C. Koller, Z. Estrov, E. Freireich, M. Keating, and M. Albitar. Plasma hepatocyte growth factor is a prognostic factor in patients with acute myeloid leukemia but not in patients with myelodysplastic syndrome. *Leukemia*, 15(8):1165–70, 2001.

[37] C. Wallace, K. B. Korb, and H. Dai. Causal discovery via MML. In *13th International Conference on Machine Learning (ICML96)*, pages 516–524. Morgan Kaufmann, 1996.

[38] B.P. Zhou, Y. Liao, W. Xia, Y. Zou, B. Spohn, and M.C. Hung. Her-2/neu induces p53 ubiquitination via akt-mediated mdm2 phosphorylation. *Nat Cell Biol.*, 3(11):973–82, 2001.

Table 1: K2, K2-Lift, K2-X^2 comparison on artificial networks with 10 variables.

		With respect to K2			With respect to K2-X^2		
Algorithm	Size	Wins	Ties	Losses	Wins	Ties	Losses
K2-X^2	1000	5	15	0	-	-	-
	5000	3	17	0	-	-	-
	10000	4	16	0	-	-	-
	20000	1	19	0	-	-	-
K2-Lift	1000	0	20	0	0	16	4
	5000	3	17	0	0	20	0
	10000	5	15	0	0	20	0
	20000	1	19	0	0	19	1

Table 2: K2, K2-Lift, K2-X^2 comparison on artificial networks with 15 variables.

		With respect to K2			With respect to K2-X^2		
Algorithm	Size	Wins	Ties	Losses	Wins	Ties	Losses
K2-X^2	1000	11	9	0	-	-	-
	5000	7	13	0	-	-	-
	10000	7	13	0	-	-	-
	20000	3	17	0	-	-	-
K2-Lift	1000	0	20	0	0	11	9
	5000	6	14	0	0	20	0
	10000	11	9	0	0	20	0
	20000	5	15	0	2	17	1

Table 3: K2, K2-Lift, K2-X^2 comparison on artificial networks with 20 variables.

		With respect to K2			With respect to K2-X^2		
Algorithm	Size	Wins	Ties	Losses	Wins	Ties	Losses
K2-X^2	1000	13	7	0	-	-	-
	5000	12	8	0	-	-	-
	10000	10	10	0	-	-	-
	20000	8	12	0	-	-	-
K2-Lift	1000	2	18	0	0	8	12
	5000	12	8	0	0	19	1
	10000	13	7	0	0	20	0
	20000	15	5	0	7	13	0

Table 4: Average MA and EA of K2, K2-Lift and K2-X^2 on benchmark networks.

		K2		K2-Lift		K2-X^2	
Network	Size	MA	EA	MA	EA	MA	EA
Asia	5000	2.17	5.03	1.83	3.20	1.83	3.77
	20000	1.37	5.60	0.93	2.60	0.97	4.10
Alarm	5000	4.67	39.77	4.63	28.86	4.63	29.40
	10000	3.30	45.17	3.07	29.07	3.07	30.23
Boelarge92	5000	7.46	9.56	7.46	7.70	7.40	8.20
	20000	6.10	15.10	6.70	11.33	6.10	12.90

Table 5: t statistics values of the comparisons between K2, K2-Lift, K2-X^2 on benchmark networks.

Network	Size	K2-Lift Vs K2	K2-X^2 Vs K2	K2-X^2 Vs. K2-Lift
Asia	5000	10.63	8.73	-6.16
	20000	11.37	10.11	-8.33
Alarm	5000	18.26	18.49	-3.76
	10000	18.22	17.48	-4.97
Boelarge92	5000	10.50	9.78	-4.01
	20000	10.43	10.43	-3.38

Table 6: Average WA of K2, K2-Lift and K2-X^2 on benchmark networks.

		K2	K2-Lift	K2-X^2
Network	Size	WA	WA	WA
Asia	500	7.80	6.00	5.80
	1000	6.67	4.76	5.26
	2000	6.47	4.10	4.63
	3000	6.84	4.93	5.50
Alarm	500	40.86	39.06	32.90
	1000	41.54	36.84	32.03
	2000	44.07	34.07	33.87
	3000	44.93	33.00	32.53

```
for $i = 1$ to $n$
{
    $\pi(V_i) = \emptyset$
    repeat
    {
        select $V_j \in \{V_1, \ldots, V_{i-1}\} - \pi(V_i)$ that maximizes
            $g(V_i, \pi(V_i) \cup \{V_j\})$
        $\Delta = g(V_i, \pi(V_i) \cup \{V_j\}) - g(V_i, \pi(V_i))$
        if $\Delta > 0$ then $\pi(V_i) = \pi(V_i) \cup \{V_j\}$
    } until $\Delta < 0$ or $\pi(V_i) = \{V_1, \ldots, V_{i-1}\}$
}
```

Figure 1: K2 algorithm

```
1. Chose randomly a a topological sort of the nodes.
2. Let $\mathcal{G}$ be a graph containing the nodes and no edges.
3. For each node $C$:
    (a) Choose $MP$ other nodes (possible parents).
    (b) For each of these nodes $P$:
        i. If $P$ precedes $C$ in the topological sort and
           a randomly generated number from 0 to 1 is less than $p$,
           add an edge from $P$ to $C$ to the graph $\mathcal{G}$.
4. If $\mathcal{G}$ has no edges, go to 2 otherwise, return $\mathcal{G}$.
```

Figure 2: Procedure for generating the networks.

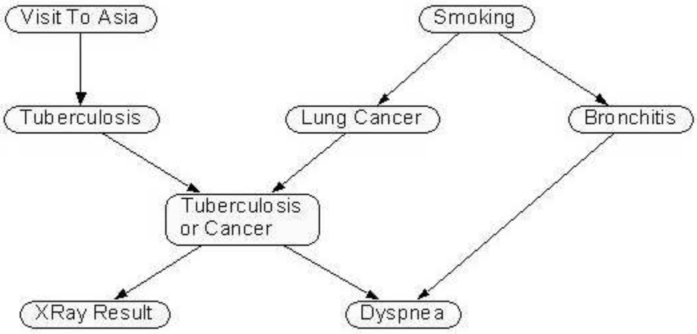

Figure 3: Asia network