An Empirical Investigation of the K2 Metric

Christian Borgelt and Rudolf Kruse

Department of Knowledge Processing and Language Engineering
Otto-von-Guericke-University of Magdeburg
Universitätsplatz 2, D-39106 Magdeburg, Germany
E-mail: {borgelt,kruse}@iws.cs.uni-magdeburg.de

Abstract. The K2 metric is a well-known evaluation measure (or scoring function) for learning Bayesian networks from data [7]. It is derived by assuming uniform prior distributions on the values of an attribute for each possible instantiation of its parent attributes. This assumption introduces a tendency to select simpler network structures. In this paper we modify the K2 metric in three different ways, introducing a parameter by which the strength of this tendency can be controlled. Our experiments with the ALARM network [2] and the BOBLO network [17] suggest that—somewhat contrary to our expectations—a slightly stronger tendency towards simpler structures may lead to even better results.

1 Introduction

Probabilistic inference networks—especially Bayesian networks [15] and Markov networks [14]—are well-known tools for reasoning under uncertainty in multidimensional domains. The idea underlying them is to exploit independence relations between the attributes used to describe a domain—an approach which has been studied extensively in the field of graphical modeling, see e.g. [12]—in order to decompose a multivariate probability distribution into a set of (conditional or marginal) distributions on lower-dimensional subspaces. Early efficient implementations include HUGIN [1] and PATHFINDER [9].

In this paper we focus on Bayesian networks. Formally, a Bayesian network represents a factorization of a multivariate probability distribution that results from an application of the product theorem of probability theory and a simplification of the factors achieved by exploiting conditional independence statements of the form $P(A \mid B, X) = P(A \mid X)$, where A and B are attributes and X is a set of attributes. Hence the represented joint distribution can be computed as

$$P(A_1, \ldots, A_n) = \prod_{i=1}^{n} P(A_i \mid \mathrm{par}(A_i)),$$

where $\mathrm{par}(A_i)$ is the set of parents of attribute A_i in a directed acyclic graph that is used to represents the factorization.

Bayesian networks provide excellent means to structure complex domains and to draw inferences. However, constructing a Bayesian network manually can

be tedious and time-consuming. Considerable expert knowledge—domain knowledge as well as mathematical knowledge—is necessary to get it right. Therefore an important line of research is the automatic construction of Bayesian networks from a database of sample cases. Most algorithms for this task consist of two ingredients: a *search method* to traverse the possible network structures and an *evaluation measure* or *scoring function* to assess the quality of a given network.

In this paper we consider only the latter component, i.e., the scoring function. A desirable property of a scoring function is *decomposability*, i.e., that it can be computed by aggregating local assessments of subnetworks or even single edges. Intuitively, a decomposable scoring function assesses the significance of dependences between attributes in the database, in order to decide which edges between attributes are needed in the Bayesian network. An example for a decomposable scoring functions is mutual information [13, 6]. Decomposable scoring functions are often used to select parents for each attribute, for example, in a greedy manner as in the K2 algorithm [7].

Due to the analogy of selecting parents for an attribute to the induction of a decision tree, there is a large variety of scoring functions [11, 19, 3]. Each of them exhibits a different sensitivity w.r.t. dependences in the data: Some scoring functions tend to select more edges/parents than others. Since in a cooperation with DaimlerChrysler, in which we work on fault diagnosis, it turned out that it is of practical importance to be able to control this sensitivity, we searched for parameterized families of scoring functions, where the parameter controls the sensitivity. In this paper we report the results of this research, which led us to certain variants of the K2 metric.

2 The K2 Metric

The K2 metric was derived first in [7], where it was used in the K2 algorithm, and later generalized in [10] to the Bayesian Dirichlet metric. It is the result of a Bayesian approach to learning Bayesian networks from data. The idea is as follows [7]: We are given a database D of sample cases over a set of attributes, each having a finite domain. It is assumed (1) that the process that generated the database can be accurately modeled as a Bayesian network, (2) that given a Bayesian network model cases occur independently, and (3) that cases are complete. Given these assumption we can compute from a given network structure B_S and a set of conditional probabilities B_P associated with it the probability of the database, i.e., we can compute $P(D|B_S, B_P)$. Adding an assumption about the prior probabilities of the network structures and the probability parameters and integrating over all possible sets of conditional probabilities B_P for the given structure B_S yields $P(B_S, D)$:

$$P(B_S, D) = \int_{B_P} P(D|B_S, B_P) f(B_P|B_S) P(B_S) \mathrm{d}B_P,$$

where f is the density function on the space of possible conditional probabilities and $P(B_S)$ is the prior probability of the structure B_S. $P(B_S, D)$ can be used

to rank possible network structures, since obviously

$$\frac{P(B_{S_i}|D)}{P(B_{S_j}|D)} = \frac{P(B_{S_i}, D)}{P(B_{S_j}, D)}.$$

With the additional assumption that the density functions f are marginally independent for all pairs of attributes and for all pairs of instantiations of the parents of an attribute, we arrive at (see [7] for details):

$$P(B_S, D) = P(B_S) \prod_{k=1}^{n} \prod_{j=1}^{q_k} \int_{\theta_{ijk}} \cdots \int \left(\prod_{i=1}^{r_k} \theta_{ijk}^{N_{ijk}} \right) f(\theta_{1jk}, \ldots, \theta_{r_k jk}) \mathrm{d}\theta_{1jk} \ldots \mathrm{d}\theta_{r_k jk}.$$

Here n is the number of attributes of the network, q_k is the number of distinct instantiations of the parents attribute k has in the structure B_S, and r_k is the number of values of attribute k. θ_{ijk} is the probability that attribute k assumes the i-th value of its domain, given that its parents are instantiated with the j-th combination of values, and N_{ijk} is the number of cases in the database, in which the attribute k is instantiated with its i-th value and its parents are instantiated with the j-th value combination.

In the following we confine ourselves to single factors of the outermost product and thus drop the index k. That is, we consider only single attribute scores. This is justified because of the factorization property of Bayesian networks. Using a uniform prior density on the parameters θ_{ij}, namely $f(\theta_{1j}, \ldots, \theta_{rj}) = (r-1)!$, and assuming that the possible networks structures are equally likely yields as a scoring function [7]:

$$\mathrm{K2}(A|\operatorname{par}(A)) = \prod_{j=1}^{q} \frac{(r-1)!}{(N_{\cdot j} + r - 1)!} \prod_{i=1}^{r} N_{ij}!,$$

where A is a child attribute and $\operatorname{par}(A)$ is the set of its parents. r is the number of values of the attribute A and q is the numbers of distinct instantiations of its parent attributes. N_{ij} is the number of cases in which attribute A is instantiated with its i-th value and its parents are instantiated with their j-th value combination; $N_{\cdot j} = \sum_{i=1}^{r} N_{ij}$. Note that in the derivation of the above function the solution of Dirichlet's integral [8]

$$\int_{\theta_{ij}} \cdots \int \prod_{i=1}^{r} \theta_{ij}^{N_{ij}} \mathrm{d}\theta_{1j} \ldots \mathrm{d}\theta_{rj} = \frac{\prod_{i=1}^{r} N_{ij}!}{(N_{\cdot j} + r - 1)!}$$

was used, which we need again below.

The higher the value of the above scoring function K2 (i.e., its product over all attributes), the better the corresponding network structure. To simplify the computation of this measure often the logarithm of the above function is used:

$$\log_2(\mathrm{K2}(A|\operatorname{par}(A))) = \sum_{j=1}^{q} \log_2 \frac{(r-1)!}{(N_{\cdot j} + r - 1)!} + \sum_{j=1}^{q} \sum_{i=1}^{r} \log_2 N_{ij}!.$$

As already said above, the K2 metric was generalized to the Bayesian Dirichlet metric in [10]. This more general scoring function is defined as

$$\mathrm{BD}(A|\operatorname{par}(A)) = \prod_{j=1}^{q} \frac{\Gamma(N'_{\cdot j})}{\Gamma(N_{\cdot j} + N'_{\cdot j})} \prod_{i=1}^{r} \frac{\Gamma(N_{ij} + N'_{ij})}{\Gamma(N'_{ij})},$$

where Γ is the well-known generalized factorial,

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} \mathrm{d}t, \qquad \forall n \in \mathbb{N} : \Gamma(n+1) = n!.$$

It is used to take care of the fact that N'_{ij} and $N'_{\cdot j} = \sum_{i=1}^{r} N'_{ij}$, which represent a prior distribution (see [10] for details), may not be integer numbers. Obviously, the K2 metric results for the simple choice $\forall i, j : N'_{ij} = 1$, which very clearly signifies the assumption of a uniform prior distribution.

This representation also makes it plausible why the K2 metric has a tendency to select simpler network structures, i.e., why algorithms using it are somewhat reluctant to add parent attributes. By the prior $N'_{ij} = 1$ the frequency distributions are somewhat "leveled out" and the more so, the more parent attributes there are. The reason is that the number of cases in the database for a given instantiation of the parent attributes is the smaller, the more parents there are, simply because each parent introduces an additional constraint. Hence the influence of the data frequencies N_{ij} is smaller for a larger number of parents and consequently an attribute seems to be less strongly dependent on its parents. The result is an inclination to reject a(nother) parent.

Analogously, we can see why the Bayesian Dirichlet likelihood equivalent uniform (BDeu) metric [5, 10], which has $\forall i, j : N'_{ij} = \frac{s}{r \cdot q}$, where s is a parameter called the *equivalent sample size*, has a tendency to select more complex network structures and tends to connect attributes with many possible values. Due to the product $r \cdot q$ in the denominator the influence of the prior is reduced by an additional parent and by parents with many possible values. The result is an increased influence of the data frequencies N_{ij} for more parents and thus a tendency to add a(nother) parent attribute.

3 Modifications of the K2 Metric

In this section we introduce three modifications of the K2 metric, all of which contain a parameter through which the strength of the tendency of the K2 metric towards simpler network structures can be controlled.

3.1 Weighted Data

The argument given above to explain the tendency of the K2 metric directly suggests an idea to control this tendency. Since the tendency depends on the relation of the data frequencies N_{ij} and the prior $N'_{ij} = 1$ one may consider

weighting either of them. Due to the numerical properties of the Γ-function, especially its behavior for arguments less than 1, weighting the data frequencies seems to be preferable. That is, we simply multiply the data frequencies with a factor β, which we write as $\beta = (\alpha_1 + 1)^2$, since this form is advantageous for the presentation of our experimental results (see below).

This factor can also be made plausible as follows: Formally the factor β is equivalent to the assumption that we observed the data β times and thus we artificially increase or reduce the statistical basis of the network induction. Of course, a larger statistical basis allows us to justify a more complex structure, whereas a smaller basis allows us only to justify a simpler one. It should be noted, though, that we introduce this factor here only to study the properties of the K2 metric, not as a statistically justifiable correction factor.

With such a factor we get the following family of scoring functions:

$$\mathrm{K2}^{(1)}_{\alpha_1}(A|\operatorname{par}(A)) = \prod_{j=1}^{q} \frac{\Gamma(r)}{\Gamma((\alpha_1+1)^2 N_{\cdot j} + r)} \prod_{i=1}^{r} \Gamma((\alpha_1+1)^2 N_{ij} + 1),$$

Obviously, for $\alpha_1 = 0$ we have the standard K2 metric as it was described above. For $\alpha_1 < 0$ we get a stronger, for $\alpha_1 > 0$ we get a weaker tendency to select simpler network structures.

3.2 Modified Prior

In the derivation of the K2 metric it is assumed that the density functions on the spaces of conditional probabilities are uniform. However, after we found the best network structure w.r.t. the K2 metric, we no longer integrate over all conditional probabilities (e.g. when we propagate evidence in the induced network). Although, of course, it is possible in principle to average over several network structures, a single network is often preferred. Hence we fix the structure and compute estimates of the probabilities using, for example, Bayesian or maximum likelihood estimation. Therefore the idea suggests itself to reverse these steps. That is, we could estimate first for each structure the best conditional probability assignments and then select the best structure based on these, then fixed, assignments. Formally, this can be done by choosing the density functions in such a way that the estimated probabilities have probability 1. Using maximum likelihood estimation of a multinomial distribution we thus get

$$f(\theta_{1j}, \ldots, \theta_{rj}) = \prod_{i=1}^{r} \delta\left(\theta_{ij} - \frac{N_{ij}}{N_{\cdot j}}\right)$$

where δ is Dirac's δ-function (or, more precisely, δ-distribution, since it is not a classical function), which is defined to have the following properties:

$$\delta(t) = \begin{cases} +\infty & \text{for } t = 0, \\ 0 & \text{for } t \neq 0, \end{cases} \qquad \int_{-\infty}^{+\infty} \delta(t)\mathrm{d}t = 1, \qquad \int_{-\infty}^{+\infty} \delta(t)\varphi(t)\mathrm{d}t = \varphi(0).$$

Inserting this density function into the formula for $P(B_S, D)$ derived above, we get as a scoring function:

$$\begin{aligned}\mathrm{K2}^{(2)}_{\infty}(A|\operatorname{par}(A)) &= \prod_{j=1}^{q} \int_{\theta_{ij}}^{\cdots} \int \left(\prod_{i=1}^{r} \theta_{ij}^{N_{ij}}\right) \left(\prod_{i=1}^{r} \delta\left(\theta_{ij} - \frac{N_{ij}}{N_{\cdot j}}\right)\right) \mathrm{d}\theta_{1j} \ldots \mathrm{d}\theta_{r_k j} \\ &= \prod_{j=1}^{q} \left(\prod_{i=1}^{r} \left(\frac{N_{ij}}{N_{\cdot j}}\right)^{N_{ij}}\right)\end{aligned}$$

An interesting thing to note about this function is that obviously

$$N_{\cdot\cdot} \cdot H(A|\operatorname{par}(A)) = -\log_2 \mathrm{K2}^{(2)}_{\infty}(A|\operatorname{par}(A)),$$

where $N_{\cdot\cdot} = \sum_{j=1}^{q} N_{\cdot j}$ and $H(A|\operatorname{par}(A))$ is the expected entropy of the probability distribution on the values of attribute A given its parents. Note that we get the well-known *mutual information* (also called *cross entropy* or *information gain*) [13, 6, 16] if we relate the value of this measure to its value for a structure in which attribute A has no parents, i.e.,

$$N_{\cdot\cdot} \cdot I_{\mathrm{gain}}(A, \operatorname{par}(A)) = \log_2 \frac{\mathrm{K2}^{(2)}_{\infty}(A|\operatorname{par}(A))}{\mathrm{K2}^{(2)}_{\infty}(A|\emptyset)}.$$

In other words, mutual information turns out to be equivalent to a so-called *Bayes factor* of this metric.

This Bayesian justification of mutual information as a scoring function may be doubted, since in it the database is—in a way—used twice to assess the quality of a network structure, namely once directly and once indirectly through the estimation of the parameters of the conditional probability distribution. Formally this approach is not strictly correct, since the density function on the parameter space should be a prior distribution whereas the estimate we used clearly is a posterior distribution (since it is computed from the database). However, the fact that mutual information results—a well-known and well-founded scoring function—is very suggestive evidence that this approach is worth to be examined.

The above derivation of mutual information as a scoring function assumes Dirac pulses at the maximum likelihood estimates for the conditional probabilities. However, we may also consider the likelihood function directly, i.e.,

$$f(\theta_{1j}, \ldots, \theta_{rj}) = c_1 \prod_{i=1}^{r} \theta_{ij}^{N_{ij}}, \qquad c_1 = \frac{(N_{\cdot j} + r - 1)!}{\prod_{i=1}^{r} N_{ij}!}.$$

where the value of the normalization constant c_1 results from the solution of Dirichlet's integral (see above) and the fact that the integral over $\theta_{1j}, \ldots, \theta_{rj}$ must be 1 (since f is a probability density function).

With this consideration a family of scoring functions suggests itself, which can be derived as follows: First we normalize the likelihood function, so that the maximum value of this function becomes 1. This is easily achieved by dividing the

likelihood function by the maximum likelihood estimate raised to the power N_{ij}. Then we introduce an exponent α_2, through which we can control the "width" of the function around the maximum likelihood estimate. Thus, if the exponent is zero, we get a constant function, if it is one, we get a function proportional to the likelihood function, and if it approaches infinity, it approaches Dirac pulses at the maximum likelihood estimate. That is, we get the family:

$$f_{\alpha_2}(\theta_{1j}, \ldots, \theta_{rj}) = c_2 \cdot \left(\left(\prod_{i=1}^{r} \left(\frac{N_{ij}}{N_{\cdot j}} \right)^{-N_{ij}} \right) \left(\prod_{i=1}^{r} \theta_{ij}^{N_{ij}} \right) \right)^{\alpha_2} = c_3 \cdot \prod_{i=1}^{r} \theta_{ij}^{\alpha_2 N_{ij}}.$$

c_2 and c_3 are normalization factors to be chosen in such a way that the integral over $\theta_{1j}, \ldots, \theta_{rj}$ is 1. Thus we find, using again the solution of Dirichlet's integral,

$$c_3 = \frac{\Gamma(\alpha_2 N_{\cdot j} + r)}{\prod_{i=1}^{r} \Gamma(\alpha_2 N_{ij} + 1)}.$$

Inserting the derived parameterized density into the function for the probability $P(B_S, D)$ and evaluating the formula using Dirichlet's integral yields the family of scoring functions

$$\mathrm{K2}^{(2)}_{\alpha_2}(A|\operatorname{par}(A)) = \prod_{j=1}^{q} \frac{\Gamma(\alpha_2 N_{\cdot j} + r)}{\Gamma((\alpha_2 + 1) N_{\cdot j} + r)} \prod_{i=1}^{r} \frac{\Gamma((\alpha_2 + 1) N_{ij} + 1)}{\Gamma(\alpha_2 N_{ij} + 1)}.$$

From the derivation above it is clear that we get the K2 metric for $\alpha_2 = 0$. Since α_2 is, like α_1, a kind of data weighting factor, we have a measure with a stronger tendency towards simpler network structures for $\alpha_2 < 0$ and a measure with a weaker tendency for $\alpha_2 > 0$. However, in order to keep the argument of the Γ-function positive, negative values of α_2 cannot be made arbitrarily large. Actually, due to the behavior of the Γ-function for arguments less than 1, only positive values seem to be useful.

3.3 Weighted Coding Penalty

It is well-known that Bayesian estimation is closely related to the minimum description length (MDL) principle [18]. Thus it is not surprising that the K2 metric can also be justified by means of this principle. The idea is as follows (see e.g. [11], where it is described w.r.t. decision tree induction): Suppose the database of sample cases is to be transmitted from a sender to a receiver. Both know the number of attributes, their domains, and the number of cases in the database[1], but at the beginning only the sender knows the values the attributes are instantiated with in the sample cases. Since transmission is costly, it is tried to code the values using a minimal number of bits. This can be achieved by exploiting

[1] A strict application of the MDL principle would assume that these numbers are unknown to the receiver. However, since they have to be transmitted in any case, they do not change the ranking and thus are neglected or assumed to be known.

properties of the value distributions to construct a good coding scheme. However, the receiver cannot know this coding scheme without being told and thus the coding scheme has to be transmitted, too. Therefore the total length of the description of the coding scheme and the description of the values based on the chosen coding scheme has to be minimized.

The transmission is carried out as follows: The values of the sample cases are transmitted attribute by attribute. That is, at first the values of the first attribute are transmitted for all sample cases, then the values of the second attribute are transmitted, and so on. Thus the transmission of the values of an attribute may exploit dependences between this attribute and already transmitted attributes to code the values more efficiently. Using a coding based on absolute value frequencies (for coding based on relative frequencies, see [11, 3]) and exploiting that the values of a set $\mathrm{par}(A)$ of already transmitted attributes are known, the following formula can be derived for the length of a description of the values of attribute A:

$$L(A|\,\mathrm{par}(A)) = \log_2 S + \sum_{j=1}^{q} \log_2 \frac{(N_{\cdot j}+r-1)!}{N_{\cdot j}!\,(r-1)!} + \sum_{j=1}^{q} \log_2 \frac{N_{\cdot j}!}{\prod_{i=1}^{r} N_{ij}!}.$$

Here S is the number of possible selections of a set $\mathrm{par}(A)$ from the set of already transmitted attributes. The lower the value of the above function (that is, its sum over all attributes), the better the corresponding network structure.

The above formula can be interpreted as follows: First we transmit which subset $\mathrm{par}(A)$ of the already transmitted attributes we use for the coding. We do so by referring to a code book, in which all possible selections are printed, one per page. This book has S pages and thus transmitting the page number takes $\log_2 S$ bits. (This term is usually neglected, since it is the same for all selections of attributes.) Then we do a separate coding for each instantiation of the attributes in $\mathrm{par}(A)$. We transmit first the frequency distribution of the values of the attribute A given the j-th instantiation of the attributes in $\mathrm{par}(A)$. Since there are $N_{\cdot j}$ cases in which the attributes in $\mathrm{par}(A)$ are instantiated with the j-th value combination and since there are r values for the attribute A, there are $\frac{(N_{\cdot j}+r-1)!}{N_{\cdot j}!\,(r-1)!}$ possible frequency distributions. We assume again that all of these are printed in a code book, one per page, and transmit the page number. Finally we transmit the exact assignment of the values of the attribute A to the cases. Since we already know the frequency of the different values, there are $\frac{N_{\cdot j}!}{\prod_{i=1}^{r} N_{ij}!}$ possible assignments. Once again we assume these to be printed in a code book, one per page, and transmit the page number.

It is easy to verify that it is

$$L(A|\,\mathrm{par}(A)) = -\log_2 \mathrm{K2}(A|\,\mathrm{par}(A))$$

if we neglect the term $\log_2 S$ (see above). Hence minimizing the network score w.r.t. $L(A|\,\mathrm{par}(A))$ is equivalent to maximising it w.r.t. $\mathrm{K2}(A|\,\mathrm{par}(A))$.

The above considerations suggests a third way to introduce a parameter for controlling the tendency towards simpler network structures. In the MDL view

the tendency results from the need to transmit the coding scheme, the costs of which can be seen as a penalty for making the network structure more complex: If the dependences of the attributes do not compensate the costs for transmitting a more complex coding scheme, fewer parent attributes are selected. Hence the tendency is mainly due to the term describing the costs for transmitting the coding scheme and we may control the tendency by weighting this term. In order to achieve matching ranges of values for the parameters and thus to simplify the presentation of the experimental results (see below), we write the weighting factor as $\frac{1}{\alpha_3+1}$. Thus we get the following family of scoring functions:

$$L_{\alpha_3}(A|\operatorname{par}(A)) = \frac{1}{\alpha_3+1}\sum_{j=1}^{q}\log_2\frac{(N_{\cdot j}+r-1)!}{N_{\cdot j}!\,(r-1)!} + \sum_{j=1}^{q}\log_2\frac{N_{\cdot j}!}{\prod_{i=1}^{r}N_{ij}!}.$$

Obviously, for $\alpha_3 = 0$ we have a measure that is equivalent to the K2 metric. For $\alpha_3 < 0$ we get a measure with a stronger tendency to select simpler network structures, for $\alpha_3 > 0$ we get a measure with a weaker tendency.

4 Experimental Results

We implemented all of the abovementioned families of scoring functions as part of INES (Induction of NEtwork Structures), a prototype program for learning probabilistic networks from a database of sample, which was written by the first author. With this program we conducted several experiments based on the well-known ALARM network [2] and the BOBLO network [17]. For all experiments we used greedy parent selection w.r.t. a topological order (the search method of the K2 algorithm). Of course, other search methods may also be used, but we do not expect the results to differ significantly.

The experiments were carried out as follows: For each network we chose three database sizes, namely 1000, 2000, and 5000 tuples for the ALARM network and 500, 1000, and 2000 tuples for the BOBLO network. For each of these sizes we randomly generated ten pairs of databases from the networks. The first database of each pair was used to induce a network, the second to test it (see below). For each database size we varied the parameters introduced in the preceding section from -0.95 to 1 (for α_1 and α_3) and from 0 to 1 (for α_2) in steps of 0.05.

The induced networks were evaluated in two different ways: In the first place they were compared to the original networks by counting the number of missing edges and the number of additional edges. Furthermore they were tested against the second database of each pair (see above) by computing the log-likelihood (natural logarithm) of this database given the induced networks. For this the conditional probabilities of the induced networks were estimated from the first database of each pair (i.e., the one the network structure was induced from) with Laplace corrected maximum likelihood estimation, i.e., using

$$\forall i,j : \hat{p}_{i|j} = \frac{N_{ij}+1}{N_{\cdot j}+r},$$

in order to avoid problems with impossible tuples.

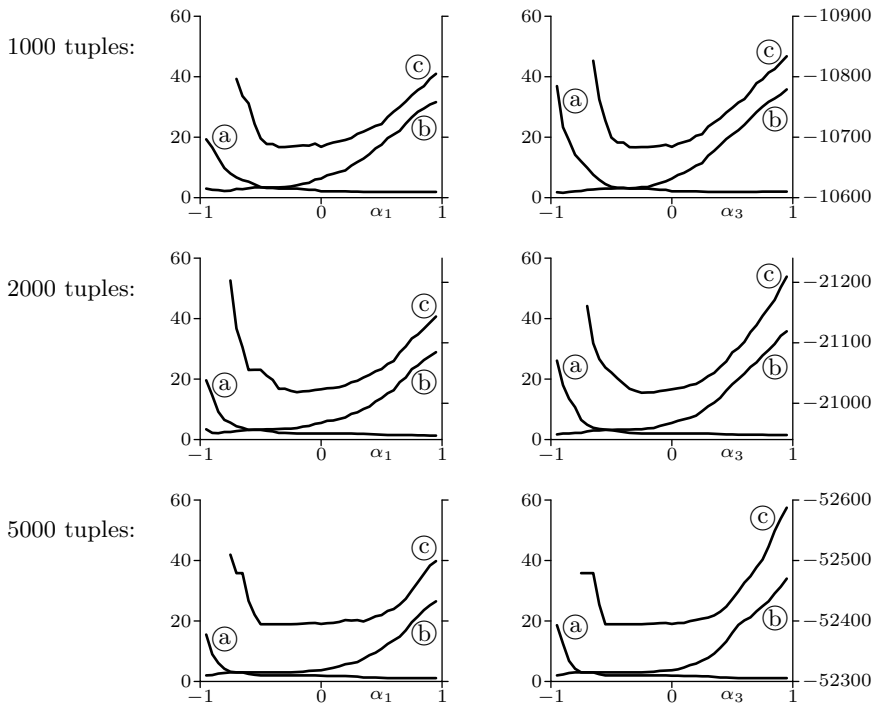

Fig. 1. Results for the ALARM network.

The results w.r.t. the parameters α_1 and α_3 are shown in figures 1 and 2. The results for α_2, which are less instructive, since this parameter should be positive, are very similar to the right halves of the diagrams for α_1 and α_3. Each diagram contains three curves, which represent averages over the ten pairs of databases:

a: the average number of missing edges,
b: the average number of additional edges,
c: the average log-likelihood of the test databases.

The scale for the number of missing/additional tuples is on the left, the scale for the log-likelihood of the test databases on the right of the diagrams.

All diagrams demonstrate that the tendency of the K2 metric (which corresponds to $\alpha_k = 0$, $k = 1, 2, 3$) is very close to optimal. However, the diagrams also indicate that a slightly stronger tendency towards simpler network structures ($\alpha_k < 0$) may lead to even better results. With a slightly stronger tendency some of the few unnecessary additional edges selected with the K2 metric can be suppressed without significantly affecting the log-likelihood of test data (actually the log-likelihood value is usually also slightly better with a stronger tendency, although this is far from being statistically significant).

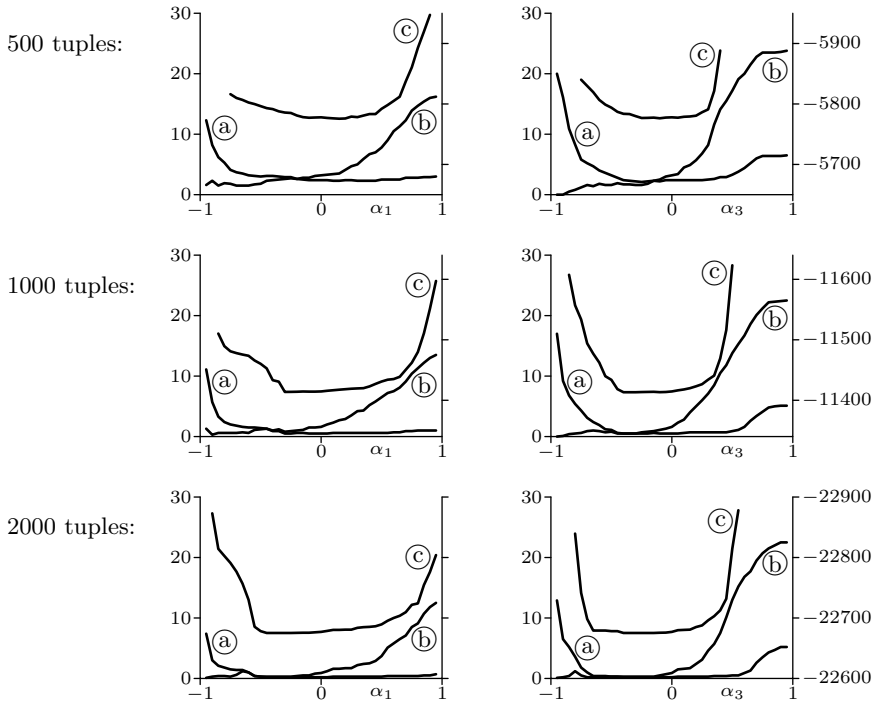

Fig. 2. Results for the BOBLO network.

It should be noted, though, that in some applications a weaker tendency towards simpler network structures is preferable. For example, in a cooperation with DaimlerChrysler, in which we work on fault diagnosis, we faced the problem that in tests against expert knowledge sometimes dependences of faults on the vehicle equipment, which were known to the domain experts, could not be found with the K2 metric. Usually this was the case if the dependence was restricted to one instantiation of the parent attributes. By adapting the parameters introduced above, however, these dependences were easily found. We regret that details of these results are confidential, so that we cannot present them here.

5 Conclusions

In this paper we introduced three modifications of the K2 metric, each of which adds a parameter to control the tendency towards simpler network structures. The resulting families of scoring functions provided us with means to explore empirically the properties of the K2 metric. Our experimental results indicate that the tendency strength of the K2 metric is a very good choice, but that a slightly stronger tendency towards simpler network structures may lead to even better results, although the improvement is only marginal.

References

1. S.K. Andersen, K.G. Olesen, F.V. Jensen, and F. Jensen. HUGIN — A Shell for Building Bayesian Belief Universes for Expert Systems. *Proc. 11th Int. J. Conf. on Artificial Intelligence (IJCAI'89, Detroit, MI, USA)*, 1080–1085. Morgan Kaufmann, San Mateo, CA, USA 1989
2. I.A. Beinlich, H.J. Suermondt, R.M. Chavez, and D.F. Cooper. The ALARM Monitoring System: A Case Study with Two Probabilistic Inference Techniques for Belief Networks. *Proc. Conf. on AI in Medical Care*, London, United Kingdom 1989
3. C. Borgelt and R. Kruse. Evaluation Measures for Learning Probabilistic and Possibilistic Networks. *Proc. 6th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'97, Barcelona, Spain)*, Vol. 2:1034–1038. IEEE Press, Piscataway, NJ, USA 1997
4. L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*, Wadsworth International Group, Belmont, CA, USA 1984
5. W. Buntine. Theory Refinement on Bayesian Networks. *Proc. 7th Conf. on Uncertainty in Artificial Intelligence (UAI'91, Los Angeles, CA, USA)*, 52–60. Morgan Kaufmann, San Mateo, CA, USA 1991
6. C.K. Chow and C.N. Liu. Approximating Discrete Probability Distributions with Dependence Trees. *IEEE Trans. on Information Theory* 14(3):462–467, IEEE Press, Piscataway, NJ, USA 1968
7. G.F. Cooper and E. Herskovits. A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning* 9:309–347. Kluwer, Amsterdam, Netherlands 1992
8. P.G.L. Dirichlet. Sur un nouvelle methode pour la determination des integrales multiples. *Comp. Rend. Acad. Science* 8:156–160. France 1839
9. D. Heckerman. *Probabilistic Similarity Networks*. MIT Press, Cambridge, MA, USA 1991
10. D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning* 20:197–243. Kluwer, Amsterdam, Netherlands 1995
11. I. Kononenko. On Biases in Estimating Multi-Valued Attributes. *Proc. 1st Int. Conf. on Knowledge Discovery and Data Mining (KDD'95, Montreal, Canada)*, 1034–1040. AAAI Press, Menlo Park, CA, USA 1995
12. R. Kruse, E. Schwecke, and J. Heinsohn. *Uncertainty and Vagueness in Knowledge-based Systems: Numerical Methods*. Springer, Berlin, Germany 1991
13. S. Kullback and R.A. Leibler. On Information and Sufficiency. *Ann. Math. Statistics* 22:79–86. Institute of Mathematical Statistics, Hayward, CA, USA 1951
14. S.L. Lauritzen and D.J. Spiegelhalter. Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. *J. Royal Statistical Society, Series B*, 2(50):157–224. Blackwell, Oxford, United Kingdom 1988
15. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (2nd edition)*. Morgan Kaufman, San Mateo, CA, USA 1992
16. J.R. Quinlan. *C4.5: Programs for Machine Learning*, Morgan Kaufman, San Mateo, CA, USA 1993
17. L.K. Rasmussen. *Blood Group Determination of Danish Jersey Cattle in the F-blood Group System*. Dina Foulum, Tjele, Denmark 1992
18. J. Rissanen. Stochastic Complexity. *Journal of the Royal Statistical Society (Series B)*, 49:223-239. Blackwell, Oxford, United Kingdom 1987
19. L. Wehenkel. On Uncertainty Measures Used for Decision Tree Induction. *Proc. 7th Int. Conf. on Inf. Proc. and Management of Uncertainty in Knowledge-based Systems (IPMU'96, Granada, Spain)*, 413–417. Universidad de Granada, Spain 1996