Journal of Machine Learning Research 7 (2006) 2149-2187 Submitted 3/06; Revised 7/06; Published 10/06

# A Scoring Function for Learning Bayesian Networks based on Mutual Information and Conditional Independence Tests

**Luis M. de Campos** LCI@DECSAI.UGR.ES
*Departamento de Ciencias de la Computación e Inteligencia Artificial*
*E.T.S.I. Informática y de Telecomunicaciones, Universidad de Granada*
*18071-Granada, Spain*

**Editor:** Nir Friedman

## Abstract

We propose a new scoring function for learning Bayesian networks from data using score+search algorithms. This is based on the concept of mutual information and exploits some well-known properties of this measure in a novel way. Essentially, a statistical independence test based on the chi-square distribution, associated with the mutual information measure, together with a property of additive decomposition of this measure, are combined in order to measure the degree of interaction between each variable and its parent variables in the network. The result is a non-Bayesian scoring function called MIT (mutual information tests) which belongs to the family of scores based on information theory. The MIT score also represents a penalization of the Kullback-Leibler divergence between the joint probability distributions associated with a candidate network and with the available data set. Detailed results of a complete experimental evaluation of the proposed scoring function and its comparison with the well-known K2, BDeu and BIC/MDL scores are also presented.

**Keywords:** Bayesian networks, scoring functions, learning, mutual information, conditional independence tests

## 1. Introduction

Nowadays, Bayesian networks (Jensen, 1996; Pearl, 1988) constitute a widely accepted formalism for representing knowledge with uncertainty and efficient reasoning. A Bayesian network comprises a qualitative and a quantitative component. While the qualitative part represents structural information about a problem domain, in the form of causality, relevance or (in)dependence relationships between variables, the quantitative part (which allows us to introduce uncertainty into the model) represents probability distributions that quantify these relationships. Once a complete Bayesian network has been built, it is an efficient tool for performing inferences. However, there still remains the previous problem of building such a network, that is, to provide the graph structure and the numerical parameters necessary for characterizing it. As it may be difficult and time-consuming to build Bayesian networks using the method of eliciting opinions from domain experts, and given the increasing availability of data in many domains, directly learning Bayesian networks from data is an interesting alternative.

There are many learning algorithms for automatically building Bayesian networks from data. Although some of these are based on testing conditional independences, in this paper we are more interested in those algorithms based on the so-called *score+search* paradigm. These see the learning task as a combinatorial optimization problem, where a search method operates on a search space associated with Bayesian networks, the search being guided by a scoring function that evaluates the degree of fitness between each element in this space and the available data.

The aim of this work is to define and study a new scoring function to be used by this class of Bayesian network learning algorithms as a competitive alternative to existing scoring functions (Bouckaert, 1993, 1995; Buntine, 1991; Chow and Liu, 1968; Cooper and Herskovits, 1992; Friedman and Goldszmidt, 1996; Heckerman et al., 1995; Herskovits and Cooper, 1990; Lam and Bacchus, 1994; Suzuki, 1993). We also want to empirically evaluate the merits of the new score by means of a comparative experimental study.

The proposed scoring function is based on the concept of mutual information. This measure has several interesting properties, the most important for our purposes being the possibility of building a statistical test of independence based on the chi-square distribution. Mutual information has already been used either directly or indirectly within Bayesian network learning algorithms based on score and search (Bouckaert, 1993; Chow and Liu, 1968; Lam and Bacchus, 1994). The associated statistical test has also been used by several learning algorithms based on conditional independence tests (Acid and de Campos, 2001; Cheng et al., 2002; de Campos and Huete, 2000; Spirtes et al., 1993). However, what is new is the simultaneous quantification of the results of a set of independence tests based on mutual information. Basically, we use mutual information in order to measure the degree of interaction between each variable and its parent variables in the network, but penalizing this value using a term related to the chi-square distribution. This penalization term takes into account not only the network complexity but also its reliability. The result will undoubtedly be a scoring function, but any score+search-based algorithm using it will have some similarities with the learning methods based on independence tests (although we believe that our scoring function makes better use of the information provided by the tests than these methods). To a certain extent what we are proposing is a hybrid algorithm (either an algorithm based on *scoring independences and search* or an algorithm based on *quantitative conditional independence tests*).

Sections 2 and 3 of this paper provide some background about learning Bayesian networks and types of scoring functions, respectively. Section 4 covers the development of the new scoring function, which we shall call MIT (mutual information tests). Section 5 carries out an empirical comparative study of MIT against several state-of-the-art scoring functions (K2, BDeu and BIC/MDL). We first define the performance measures to be used and we then describe the corresponding experimental designs and the obtained results. Section 6 contains our conclusions and some proposals for future research. Finally, Appendix A includes proof of all the theorems set out in the paper.

## 2. Learning Bayesian Networks

Let us consider a finite set $\mathbf{U_n} = \{X_1, X_2, \ldots, X_n\}$ of discrete random variables.[1] A generic variable of the set $\mathbf{U_n}$ will be denoted as either $X_i$ or $X$. The domain of each variable $X_i$ is a finite set $V_i = \{x_{i1}, \ldots, x_{ir_i}\}$. A generic element of $V_i$ will be denoted as $x_i$. In general, we shall use uppercase letters to denote variables, lowercase letters to denote states of the variables, and bold-faced letters (either uppercase or lowercase) to denote sets (of either variables or states of the variables, respectively).

A Bayesian network (BN) is a graphical representation of a joint probability distribution (Pearl, 1988) that includes two components:

[1] Although there are also Bayesian networks with continuous variables, here we are only interested in the case where all the variables are discrete.

- First, a *directed acyclic graph* (DAG) $G = (\mathbf{U_n}, E_G)$, where $\mathbf{U_n}$, the set of nodes, represents the system variables,[2] and $E_G$, the set of arcs, represents direct dependency relationships between variables; the absence of arcs linking pairs of variables in turn represents the existence of conditional independence relationships between these variables. A conditional independence relationship between two variables $X_i$ and $X_j$, given a subset of variables $\mathbf{Z}$, denoted as $I(X_i, X_j|\mathbf{Z})$, means that given the values of the variables in $\mathbf{Z}$, our degree of belief about the possible values of $X_i$ is not modified once we know the value of variable $X_j$: $p(x_i|x_j,\mathbf{z}) = p(x_i|\mathbf{z})$. Each variable $X_i \in \mathbf{U_n}$ has an associated *parent set* in the graph $G$, $Pa_G(X_i) = \{X_j \in \mathbf{U_n} \mid X_j \to X_i \in E_G\}$. If $X_i$ has no parent (it is a root node), then $Pa_G(X_i) = \emptyset$.

- The second component is a set of numerical parameters, which usually represent conditional probability distributions: for each variable $X_i$ in $\mathbf{U_n}$, we store a family of conditional distributions $p(X_i|pa_G(X_i))$, one for each possible *configuration*,[3] $pa_G(X_i)$, of the parent set of $X_i$ in the graph. If $X_i$ has no parent, then $p(X_i|pa_G(X_i))$ equals $p(X_i)$. From these conditional distributions, we can obtain the joint distribution over $\mathbf{U_n}$ using:

$$p(x_1, x_2, \ldots, x_n) = \prod_{X_i \in \mathbf{U_n}} p(x_i|pa_G(X_i))$$

The problem of learning Bayesian networks from data consists in finding the BN that (according to certain criterion) best fits the available data. This problem has been studied in depth over the last ten years and consequently, there are currently a considerable number of learning algorithms. As Bayesian networks have two different components (the graphical and the numerical model), the algorithms for learning BNs must deal with two different but highly related tasks: learning the structure (the DAG) and learning the parameters (the conditional probabilities). These two tasks cannot be carried out completely independently: on the one hand, in order to estimate the conditional probabilities, we must know the graphical structure; on the other, in order to determine whether the graph we are trying to find contains certain arcs, we need to estimate certain statistics from the data which, depending on the kind of learning algorithm being used, will be employed either to carry out some conditional independence tests or to measure the intensity of the relationships between the nodes involved in these arcs.

In this paper, we are only interested in algorithms for learning the structure of Bayesian networks. As we mentioned previously, most of these algorithms can be grouped into two different categories: methods based on *conditional independence tests* (also called *constraint-based* methods) and methods based on *scoring functions and search*, although there are also algorithms that use a combination of independence-based and scoring-based methods with different hybridization strategies (Acid and de Campos, 2000, 2001; Dash and Druzdzel, 1999; de Campos et al., 2003; Singh and Valtorta, 1995; Spirtes and Meek, 1995).

The algorithms based on independence tests (Cheng et al., 2002; de Campos, 1998; de Campos and Huete, 2000; Meek, 1995; Pearl and Verma, 1991; Spirtes et al., 1993; Verma and Pearl, 1990; Wermuth and Lauritzen, 1983) perform a qualitative study of the dependence and independence relationships between the variables in the domain (obtained from the data by means of conditional independence tests), and attempt to find a network that represents these relationships as far as possible. Two fundamental issues for these algorithms are the number and the complexity of

[2] In the same way, we shall represent a variable and its associated node in the graph.

[3] A configuration of a set of variables $\mathbf{Z}$ is an assignment of values to each of the variables in $\mathbf{Z}$.

the independence tests, and this can also cause unreliable results. Nevertheless, constraint-based algorithms generally come with rigorous theoretical founding and have developed a body of work that details sound and complete methods to make use of independence relations in the data while correctly accounting for structure.

The algorithms based on a scoring function attempt to find a graph that maximizes the selected score, which is usually defined as a measure of fitness between the graph and the data. All of them use the scoring function in combination with a search method in order to measure the goodness of each explored structure from the space of feasible solutions. Different learning algorithms are obtained depending on the search procedure used, as well as on the definitions of the scoring function and the search space.

The scoring functions are based on different principles, such as entropy and information (Chow and Liu, 1968; Herskovits and Cooper, 1990), the minimum description length (Bouckaert, 1993, 1995; Friedman and Goldszmidt, 1996; Lam and Bacchus, 1994; Suzuki, 1993), or Bayesian approaches (Buntine, 1991; Cooper and Herskovits, 1992; Heckerman et al., 1995; Kayaalp and Cooper, 2002). The most usual scoring functions will be described later in more detail.

As far as the search is concerned, although the most frequently used are local search methods (Buntine, 1991; Chickering et al., 1995; Cooper and Herskovits, 1992; de Campos et al., 2003; Heckerman et al., 1995) due to the exponentially large size of the search space, there is a growing interest in other heuristic search methods such as simulated annealing (Chickering et al., 1995), tabu search (Acid and de Campos, 2003; Bouckaert, 1995), branch and bound (Tian, 2000), genetic algorithms and evolutionary programming (Larrañaga et al., 1996; Myers et al., 1999; Wong et al., 1999), Markov chain Monte Carlo (Kocka and Castelo, 2001; Myers et al., 1999), variable neighborhood search (de Campos and Puerta, 2001a), ant colony optimization (de Campos et al., 2002), greedy randomized adaptive search procedures (GRASP) (de Campos et al., 2002), and estimation of distribution algorithms (Blanco et al., 2003).

Most learning algorithms employ different search methods but the same search space: the DAG space. Possible alternatives are the space of the orderings of the variables (de Campos et al., 2002; de Campos and Huete, 2002; de Campos and Puerta, 2001b; Friedman and Koller, 2003; Larrañaga et al., 1996), with a secondary search in the DAG space compatible with a given ordering; the space of *essential graphs* (Pearl and Verma, 1990) (also called *patterns* or *completed PDAGs*), which are partially directed acyclic graphs[4] or PDAGs that canonically represent equivalence classes of DAGs (Andersson et al., 1997; Chickering, 2002; Dash and Druzdzel, 1999; Madigan et al., 1996; Spirtes and Meek, 1995); and the space of *RPDAGs* (restricted PDAGs), which also represent equivalence classes of DAGs (Acid and de Campos, 2003; Acid et al., 2005).

## 3. Scoring Functions for Learning Bayesian Networks

Focusing on the methods for learning Bayesian networks based on the score+search paradigm, the problem can be formally expressed as follows: given a *complete*[5] training data set $D = \{\mathbf{u}^1, \ldots, \mathbf{u}^N\}$ of instances of $\mathbf{U_n}$, find a DAG $G^*$ such that

$$G^* = \arg\max_{G \in \mathcal{G}_n} g(G : D),$$

[4] Containing both directed (arcs) and undirected (links) edges.

[5] We consider neither missing values nor latent variables.

where $g(G:D)$ is the scoring function measuring the degree of fitness of any candidate DAG $G$ to the data set, and $\mathcal{G}_n$ is the family of all the DAGs defined on $\mathbf{U_n}$.

The learning algorithms that search in the DAG space with local search-based methods can be more efficient if the scoring function being used has the property of *decomposability*: a scoring function $g$ is *decomposable* if the value assigned to each structure can be expressed as a sum (in the logarithmic space) of local values that depend only on each node and its parents:

$$g(G:D) = \sum_{X_i \in \mathbf{U_n}} g(X_i, Pa_G(X_i) : D)$$

$$g(X_i, Pa_G(X_i) : D) = g(X_i, Pa_G(X_i) : N^D_{X_i, Pa_G(X_i)}),$$

where $N^D_{X_i, Pa_G(X_i)}$ are the sufficient statistics of the set of variables $\{X_i\} \cup Pa_G(X_i)$ in $D$, that is, the number of instances in $D$ corresponding to each possible configuration of $\{X_i\} \cup Pa_G(X_i)$.

For example, a search procedure that only changes one arc at each move can efficiently evaluate the improvement obtained by this change. It can reuse most of the previous computations and only the statistics for the variables whose parent sets have been modified must be recomputed. In this way, the insertion or deletion of an arc $X_j \rightarrow X_i$ in a DAG $G$ can be evaluated by computing only one new local score, $g(X_i, Pa_G(X_i) \cup \{X_j\} : D)$ or $g(X_i, Pa_G(X_i) \setminus \{X_j\} : D)$, respectively; the reversal of an arc $X_j \rightarrow X_i$ requires the evaluation of two new local scores, $g(X_i, Pa_G(X_i) \setminus \{X_j\} : D)$ and $g(X_j, Pa_G(X_j) \cup \{X_i\} : D)$.

Another property which is particularly interesting if the learning algorithm searches in a space of equivalence classes of DAGs is called the *score equivalence*: a scoring function $g$ is *score-equivalent* if it assigns the same value to all DAGs that are represented by the same essential graph. In this way, the result of evaluating an equivalence class will be the same regardless of which DAG from this class is selected.

There are different ways to measure the degree of fitness of a DAG with respect to a data set. Most can be grouped into two categories: Bayesian and information measures. We shall use the following notation: the number of states of the variable $X_i$ is $r_i$; the number of possible configurations of the parent set $Pa_G(X_i)$ of $X_i$ is $q_i$; obviously, $q_i = \prod_{X_j \in Pa_G(X_i)} r_j$; $w_{ij}$, $j = 1, \ldots q_i$, represents a configuration of $Pa_G(X_i)$; $N_{ijk}$ is the number of instances in the data set $D$ where the variable $X_i$ takes the value $x_{ik}$ and the set of variables $Pa_G(X_i)$ take the value $w_{ij}$; $N_{ij}$ is the number of instances in the data set where the variables in $Pa_G(X_i)$ take their $j$-th configuration $w_{ij}$; obviously $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$; similarly, $N_{ik}$ is the number of instances in $D$ where the variable $X_i$ takes its $k$-th value $x_{ik}$, and therefore $N_{ik} = \sum_{j=1}^{q_i} N_{ijk}$; the total number of instances in $D$ is $N$.

### 3.1 Bayesian Scoring Functions

Starting from a prior probability distribution on the possible networks, the general idea is to compute the posterior probability distribution conditioned to the available data $D$, $p(G|D)$. The best network is the one that maximizes the posterior probability. It is not in fact necessary to compute $p(G|D)$ and for comparative purposes, computing $p(G,D)$ is sufficient since the term $p(D)$ is the same for all the possible networks. As it is easier to work in the logarithmic space, in practice, the scoring functions use the value $\log(p(G,D))$ instead of $p(G,D)$.

One of the first Bayesian scoring functions, called K2, was proposed by Cooper and Herskovits (1992). It relies on several assumptions (multinomiality, lack of missing values, parameter independence, parameter modularity, uniformity of the prior distribution of the parameters given the network structure), and can be expressed as follows:

$$g_{K2}(G:D) = \log(p(G)) + \sum_{i=1}^{n}\left[\sum_{j=1}^{q_i}\left[\log\left(\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}\right) + \sum_{k=1}^{r_i}\log\left(N_{ijk}!\right)\right]\right], \qquad (1)$$

where $p(G)$ represents the prior probability of the DAG $G$. Afterwards, the so-called BD (Bayesian Dirichlet) score was proposed by Heckerman et al. (1995) as a generalization of K2:

$$g_{BD}(G:D) = \log(p(G)) + \sum_{i=1}^{n}\left[\sum_{j=1}^{q_i}\left[\log\left(\frac{\Gamma(\eta_{ij})}{\Gamma(N_{ij}+\eta_{ij})}\right) + \sum_{k=1}^{r_i}\log\left(\frac{\Gamma(N_{ijk}+\eta_{ijk})}{\Gamma(\eta_{ijk})}\right)\right]\right], \qquad (2)$$

where the values $\eta_{ijk}$ are the hyperparameters for the Dirichlet prior distributions of the parameters given the network structure, and $\eta_{ij} = \sum_{k=1}^{r_i}\eta_{ijk}$. $\Gamma(.)$ is the function *Gamma*, $\Gamma(c) = \int_0^\infty e^{-u}u^{c-1}du$. It should be noted that if $c$ is an integer, $\Gamma(c) = (c-1)!$. If the values of all the hyperparameters are $\eta_{ijk} = 1$, we obtain the K2 score as a particular case of BD.

In practical terms, the specification of the hyperparameters $\eta_{ijk}$ is quite difficult (except if we use non-informative assignments, as the ones employed by K2). However, by considering the additional assumption of likelihood equivalence (Heckerman et al., 1995), it is possible to specify the hyperparameters relatively easily. While the result is a scoring function called BDe (and its expression is identical to the BD one in Equation 2), the hyperparameters can now be computed in the following way:

$$\eta_{ijk} = \eta \times p(x_{ik}, w_{ij}|G_0),$$

where $p(.|G_0)$ represents a probability distribution associated with a *prior Bayesian network* $G_0$ and $\eta$ is a parameter representing the equivalent sample size.

A particular case of BDe which is especially interesting appears when $p(x_{ik}, w_{ij}|G_0) = \frac{1}{r_i q_i}$, that is, the prior network assigns a uniform probability to each configuration of $\{X_i\} \cup Pa_G(X_i)$. The resulting score is called BDeu, which was originally proposed by Buntine (1991). This score only depends on one parameter, the equivalent sample size $\eta$, and is expressed as follows:

$$g_{BDeu}(G:D) = \log(p(G)) + \sum_{i=1}^{n}\left[\sum_{j=1}^{q_i}\left[\log\left(\frac{\Gamma(\frac{\eta}{q_i})}{\Gamma(N_{ij}+\frac{\eta}{q_i})}\right) + \sum_{k=1}^{r_i}\log\left(\frac{\Gamma(N_{ijk}+\frac{\eta}{r_i q_i})}{\Gamma(\frac{\eta}{r_i q_i})}\right)\right]\right]. \qquad (3)$$

Regarding the term $\log(p(G))$ which appears in all the previous expressions, it is quite common to assume a uniform distribution (except if we really have information about the greater desirability of certain structures) so that it becomes a constant and can be removed.

### 3.2 Scoring Functions based on Information Theory

These scoring functions represent another option for measuring the degree of fitness of a DAG to a data set and are based on codification and information theory concepts. Coding attempts to reduce as much as possible the number of elements which are necessary to represent a message (depending on its probability). Frequent messages will therefore have shorter codes whereas larger codes will be assigned to the less frequent messages. The minimum description length principle (MDL) selects the coding that requires minimum length to represent the messages. Another more general formulation of the same idea establishes that in order to represent a data set with one model from a specific type, the best model is the one that minimizes the sum of the description length of the model and the description length of the data given the model. Complex models usually require greater description lengths but reduce the description length of the data given the model (they are more accurate). On the other hand, simple models require shorter description lengths but the description length of the data given the model increases. The minimum description length principle establishes an appropriate trade-off between complexity and precision.

In our case, the data set to be represented is $D$ and the selected class of models are Bayesian networks. Therefore, the description length includes the length required to represent the network plus the length necessary to represent the data given the network (Bouckaert, 1993, 1995; Friedman and Goldszmidt, 1996; Lam and Bacchus, 1994; Suzuki, 1993). In order to represent the network, we must store its probability values, and this requires a length which is proportional to the number of free parameters of the factorized joint probability distribution.[6] This number, called network complexity and denoted as $C(G)$, is:

$$C(G) = \sum_{i=1}^{n}(r_i - 1)q_i.$$

The usual proportionality factor is $\frac{1}{2}\log(N)$ (Rissanen, 1986). Therefore, the description length of the network is:

$$\frac{1}{2}C(G)\log(N).$$

Regarding the description of the data given the model, by using Huffmann codes its length turns out to be the negative of the log-likelihood, that is, the logarithm of the likelihood function of the data with respect to the network. This value is minimum for a fixed network structure when the network parameters are estimated from the data set itself by using maximum likelihood. The log-likelihood can be expressed in the following way (Bouckaert, 1995):

$$LL_D(G) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk}\log\left(\frac{N_{ijk}}{N_{ij}}\right). \tag{4}$$

Therefore, the MDL scoring function (by changing the signs to deal with a maximization problem) is:

$$g_{MDL}(G:D) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk}\log\left(\frac{N_{ijk}}{N_{ij}}\right) - \frac{1}{2}C(G)\log(N). \tag{5}$$

Another way of measuring the quality of a Bayesian network is to use measures based on information theory and some of these are closely related with the previous one. The basic idea is to select the network structure that best fits the data, penalized by the number of parameters which are necessary to specify the joint distribution. This leads to a generalization of the scoring function in Equation 5:

$$g(G:D) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk}\log\left(\frac{N_{ijk}}{N_{ij}}\right) - C(G)f(N), \tag{6}$$

where $f(N)$ is a non-negative penalization function. If $f(N) = 1$, the score is based on the Akaike information criterion (AIC) (Akaike, 1974). If $f(N) = \frac{1}{2}\log(N)$, then the score, called BIC, is

[6] There are other versions (Lam and Bacchus, 1994) that also include the description length of the graph itself, which is proportional to the sum of the number of parents for each node, $\sum_{i=1}^{n}|Pa_G(X_i)|$. However, the most usual formulation does not consider it.

based on the Schwarz information criterion (Schwarz, 1978), which coincides with the MDL score. If $f(N)=0$, we have the maximum likelihood score, although this is not very useful as the best network using this criterion is always a complete network which includes all the possible arcs.

It is interesting to note that another way of expressing the log-likelihood in Equation 4 is:

$$LL_D(G) = -N\sum_{i=1}^{n} H_D(X_i|Pa_G(X_i)), \tag{7}$$

where $H_D(X_i|Pa_G(X_i))$ represents the conditional entropy of the variable $X_i$ given its parent set $Pa_G(X_i)$, for the probability distribution $p_D$:

$$H_D(X_i|Pa_G(X_i)) = \sum_{j=1}^{q_i} p_D(w_{ij})\left(-\sum_{k=1}^{r_i} p_D(x_{ik}|w_{ij})\log(p_D(x_{ik}|w_{ij}))\right),$$

and $p_D$ is the joint probability distribution associated with the data set $D$, obtained from the data by maximum likelihood. The log-likelihood $LL_D(G)$ can also be expressed as follows (Bouckaert, 1995):

$$LL_D(G) = -NH_D(G),$$

where $H_D(G)$ represents the entropy of the joint probability distribution associated with the graph $G$ when the network parameters are estimated from $D$ by maximum likelihood:

$$H_D(G) = -\sum_{x_1,\ldots,x_n}\left(\left(\prod_{i=1}^{n} p_D(x_i|pa_G(X_i))\right)\log\left(\prod_{i=1}^{n} p_D(x_i|pa_G(X_i))\right)\right).$$

Therefore, another interpretation of the scoring functions based on information is that they attempt to minimize the conditional entropy of each variable given its parents, and so they search for the parent set of each variable that gives as much information as possible about this variable (or which most restricts the distribution). It is necessary to add a penalization term since the minimum conditional entropy is always obtained after adding all the possible variables to the parent set.

An alternative way to avoid this overfitting without using a penalization function was proposed by Herskovits and Cooper (1990) who used the maximum likelihood score, but the process of inserting arcs into the network was stopped by means of a statistical test, which determined whether the difference in entropy between the current network and the one obtained by including an additional arc was statistically significant.

With respect to the characteristics of the different scoring functions, all are decomposable and with the exception of K2 and BD, they are also score-equivalent (Chickering, 1995).

## 4. A New Scoring Function based on Mutual Information and Independence Tests

In order to explain the ideas behind the proposed scoring function more clearly, we shall first introduce several preliminary considerations. These will lead to a first version of the scoring function, which will be later refined in order to obtain the final version.

### 4.1 Preliminary Considerations

Our goal is to design a scoring function in such a way that the value $g(G:D)$ represents a measure of the distance between the joint probability distribution associated with the DAG $G$, $p_G$, and the joint probability distribution associated with the data, $p_D$. We should mention that $p_G$ must be understood to be the joint probability distribution that factorizes according to $G$ and whose local conditional probability distributions are estimated from $D$ by means of maximum likelihood, that is,

$$p_G(x_1,\ldots,x_n) = \prod_{i=1}^{n} p_D(x_i|pa_G(X_i)).$$

A reasonable choice for the distance measure is the *Kullback-Leibler divergence* (Kullback, 1968):

$$KL(p_D,p_G) = \sum_{x_1,\ldots,x_n} p_D(x_1,\ldots,x_n)\log\left(\frac{p_D(x_1,\ldots,x_n)}{p_G(x_1,\ldots,x_n)}\right).$$

This distance can also be expressed in another more convenient way:

$$\begin{aligned} KL(p_D,p_G) \quad &= \quad -H_D(\{X_1,\ldots,X_n\}) + \sum_{\substack{i=1\\ Pa_G(X_i)=\emptyset}}^{n} H_D(X_i) \\ &\quad + \sum_{\substack{i=1\\ Pa_G(X_i)\neq\emptyset}}^{n} \Big(H_D(\{X_i\}\cup Pa_G(X_i)) - H_D(Pa_G(X_i))\Big), \end{aligned} \tag{8}$$

where $H_D(\mathbf{X})$ represents the entropy of the set of variables $\mathbf{X}$ with respect to the distribution $p_D$.

We shall now consider the concept of *mutual information*. Given a probability distribution $p$ defined over two sets of variables $\mathbf{X}$ and $\mathbf{Y}$, the mutual information between $\mathbf{X}$ and $\mathbf{Y}$ is:

$$MI(\mathbf{X},\mathbf{Y}) = \sum_{\mathbf{x},\mathbf{y}} p(\mathbf{x},\mathbf{y})\log\left(\frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x})p(\mathbf{y})}\right),$$

which can also be expressed in terms of entropy as:

$$MI(\mathbf{X},\mathbf{Y}) = H(\mathbf{X}) + H(\mathbf{Y}) - H(\mathbf{X}\cup\mathbf{Y}). \tag{9}$$

Mutual information (which is simply the Kullback-Leibler divergence between the joint distribution for $\mathbf{X}$ and $\mathbf{Y}$ and the product of the corresponding marginals) can be considered as a way of measuring the dependence degree between the sets of variables $\mathbf{X}$ and $\mathbf{Y}$, which is null when the two sets of variables are independent and maximum when they are functionally dependent. By using Equation 9, we can rewrite Equation 8 as follows (Lam and Bacchus, 1994):

$$KL(p_D,p_G) = -H_D(\{X_1,\ldots,X_n\}) + \sum_{i=1}^{n} H_D(X_i) - \sum_{\substack{i=1\\ Pa_G(X_i)\neq\emptyset}}^{n} MI_D(X_i, Pa_G(X_i)). \tag{10}$$

As the two first terms in Equation 10 do not depend on the DAG $G$ being considered, we obtain:

$$\arg\min_{G\in\mathcal{G}_n} KL(p_D,p_G) = \arg\max_{G\in\mathcal{G}_n} \sum_{\substack{i=1\\ Pa_G(X_i)\neq\emptyset}}^{n} MI_D(X_i, Pa_G(X_i)), \tag{11}$$

and therefore minimizing the Kullback-Leibler divergence is equivalent to maximizing the sum of the measures of mutual information between each variable and its parent variables in the graph.

We have still not achieved anything useful, however, since mutual information has the property that $MI(\mathbf{X},\mathbf{Y}\cup\mathbf{W}) \geq MI(\mathbf{X},\mathbf{Y})$, in other words, mutual information always increases by including additional variables. Therefore, the complete network will always have minimum Kullback-Leibler divergence with respect to the data. In fact, by taking into account Equation 7 and the relation between mutual information and conditional entropy, namely $MI(\mathbf{X},\mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y})$, we can write:

$$\sum_{\substack{i=1\\ Pa_G(X_i)\neq\emptyset}}^{n} MI_D(X_i, Pa_G(X_i)) = \frac{LL_D(G)}{N} + \sum_{i=1}^{n} H_D(X_i). \tag{12}$$

Therefore, minimizing the Kullback-Leibler divergence is also equivalent to maximizing log-likelihood. The following expression is equivalent to the previous one:

$$\sum_{\substack{i=1\\ Pa_G(X_i)\neq\emptyset}}^{n} MI_D(X_i, Pa_G(X_i)) = \frac{1}{N}\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk}\log\left(\frac{N N_{ijk}}{N_{ik}N_{ij}}\right).$$

However, there are certain advantages to using mutual information instead of log-likelihood as we shall see later. First, let us consider the concept of *conditional mutual information* between $\mathbf{X}$ and $\mathbf{Y}$ given a set of variables $\mathbf{Z}$, defined as:

$$MI(\mathbf{X},\mathbf{Y}|\mathbf{Z}) = \sum_{\mathbf{z}}\left(p(\mathbf{z})\sum_{\mathbf{x},\mathbf{y}} p(\mathbf{x},\mathbf{y}|\mathbf{z})\log\left(\frac{p(\mathbf{x},\mathbf{y}|\mathbf{z})}{p(\mathbf{x}|\mathbf{z})p(\mathbf{y}|\mathbf{z})}\right)\right),$$

which can be expressed by $MI(\mathbf{X},\mathbf{Y}|\mathbf{Z}) = H(\mathbf{X}|\mathbf{Z}) - H(\mathbf{X}|\mathbf{Y}\cup\mathbf{Z})$, and also by:

$$MI(\mathbf{X},\mathbf{Y}|\mathbf{Z}) = H(\mathbf{X}\cup\mathbf{Z}) + H(\mathbf{Y}\cup\mathbf{Z}) - H(\mathbf{Z}) - H(\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z}).$$

The following property[7] of conditional mutual information is important for our purposes:

$$MI(\mathbf{X},\mathbf{Y}\cup\mathbf{W}|\mathbf{Z}) = MI(\mathbf{X},\mathbf{Y}|\mathbf{Z}) + MI(\mathbf{X},\mathbf{W}|\mathbf{Z}\cup\mathbf{Y}). \tag{13}$$

Another fundamental property of mutual information is:

**Theorem 1 (Kullback, 1968)** *Given a data set D with N elements, if the hypothesis that* $\mathbf{X}$ *and* $\mathbf{Y}$ *are conditionally independent given* $\mathbf{Z}$ *is true, then the statistics* $2N\,MI_D(\mathbf{X},\mathbf{Y}|\mathbf{Z})$ *approximates to a distribution* $\chi^2(l)$ *(Chi-square) with* $l = (r_{\mathbf{X}}-1)(r_{\mathbf{Y}}-1)r_{\mathbf{Z}}$ *degrees of freedom, where* $r_{\mathbf{X}}$, $r_{\mathbf{Y}}$ *and* $r_{\mathbf{Z}}$ *represent the number of configurations for the sets of variables* $\mathbf{X}$, $\mathbf{Y}$ *and* $\mathbf{Z}$, *respectively. If* $\mathbf{Z} = \emptyset$, *the statistics* $2N\,MI_D(\mathbf{X},\mathbf{Y})$ *approximates to a distribution* $\chi^2(l)$ *with* $l = (r_{\mathbf{X}}-1)(r_{\mathbf{Y}}-1)$ *degrees of freedom.*

### 4.2 Developing a New Scoring Function

The basic idea underlying the new scoring function that we shall propose is very simple: to use the mutual information $MI_D(X_i, Pa_G(X_i))$ in order to measure the degree of interaction between each variable $X_i$ and its parents $Pa_G(X_i)$, as in Equation 11, but penalizing this value using a term related

[7] It should be noted that this property is a numeric version of the properties of *decomposition*, *weak union* and *contraction* of the probabilistic independence relationships and other dependence models (Pearl, 1988). These three properties, together with *symmetry*, characterize the dependence models called *semi-graphoids*.

to the $\chi^2$ distribution. This term attempts to re-scale the mutual information values in order to prevent these values from systematically increasing as the number of variables in $Pa_G(X_i)$ does.

In our opinion, one problem with the scoring functions based on information (Equation 6) is that they penalize log-likelihood globally, with a combination of the network complexity and a function that depends only on the number of instances. Since we believe that as the log-likelihood can be decomposed as a sum of components (each being associated with a variable and its parents), then each of these components should be penalized differently, depending not only on its complexity but also on its *reliability*. For example, a DAG where a variable $X_i$ has many parents is always penalized in the same way, without taking into account to what extent this topology is actually necessary to adequately and reliably represent the distribution for $X_i$. The scoring function that we shall propose naturally incorporates this kind of penalization, and is based on solid statistical grounds.

Given a DAG $G$, let us consider the mutual information between a variable $X_i$ and its parents, $MI_D(X_i, Pa_G(X_i))$. Let $s_i$ be the number of parent variables[8] of $X_i$, $s_i = |Pa_G(X_i)|$. Let us assume that $Pa_G(X_i) = \{X_{i1}, \ldots, X_{is_i}\}$. By iteratively applying Equation 13, we can express $MI_D(X_i, Pa_G(X_i))$ as:

$$\begin{aligned}
MI_D(X_i, Pa_G(X_i)) &= MI_D(X_i, \{X_{i1}, \ldots, X_{is_i}\}) \\
&= MI_D(X_i, \{X_{i1}, \ldots, X_{i(s_i-1)}\}) + MI_D(X_i, X_{is_i} | \{X_{i1}, \ldots, X_{i(s_i-1)}\}) \\
&= MI_D(X_i, \{X_{i1}, \ldots, X_{i(s_i-2)}\}) + MI_D(X_i, X_{i(s_i-1)} | \{X_{i1}, \ldots, X_{i(s_i-2)}\}) + \\
&\quad MI_D(X_i, X_{is_i} | \{X_{i1}, \ldots, X_{i(s_i-1)}\}) = \ldots\ldots \\
&= MI_D(X_i, X_{i1}) + \sum_{j=2}^{s_i} MI_D(X_i, X_{ij} | \{X_{i1}, \ldots, X_{i(j-1)}\}). \qquad (14)
\end{aligned}$$

The elements in this decomposition of the mutual information will be interpreted as follows: starting with an empty set of parents of $X_i$, we have first included the arc $X_{i1} \rightarrow X_i$, and the degree of dependence between these variables is $MI_D(X_i, X_{i1})$. We then insert the arc $X_{i2} \rightarrow X_i$ and as $X_{i1}$ is already a parent of $X_i$, the dependence degree between $X_{i2}$ and $X_i$ is $MI_D(X_i, X_{i2} | X_{i1})$. We continue inserting arcs in this way until the last one $X_{is_i} \rightarrow X_i$ (with a dependence degree between $X_{is_i}$ and $X_i$ equal to $MI_D(X_i, X_{is_i} | \{X_{i1}, \ldots, X_{i(s_i-1)}\})$) has been included. If we do not insert any additional arcs, this is because each remaining variable $X_h$ does not contribute any additional information[9] with respect to $X_i$, this information being measured as $MI_D(X_i, X_h | \{X_{i1}, \ldots, X_{is_i}\})$. The key question is how to determine whether the values of mutual information represent an appreciable (i.e., statistically significant) amount of information. At this point, we can use the result in Theorem 1.

We know that $2N MI_D(X_i, X_{ij} | \{X_{i1}, \ldots, X_{i(j-1)}\})$ approximates to a distribution $\chi^2(l_{ij})$, with the appropriate degrees of freedom $l_{ij}$. Let us fix a *confidence level* $\alpha$ and determine the value $\chi_{\alpha, l_{ij}}$ such that $p(\chi^2(l_{ij}) \leq \chi_{\alpha, l_{ij}}) = \alpha$. This does in fact represent a statistical test of conditional independence: if $2N MI_D(X_i, X_{ij} | \{X_{i1}, \ldots, X_{i(j-1)}\}) \leq \chi_{\alpha, l_{ij}}$, then we accept the hypothesis of independence between $X_i$ and $X_{ij}$ given $\{X_{i1}, \ldots, X_{i(j-1)}\}$ (with probability $\alpha$); otherwise we reject it.

The use of this kind of independence test within BN learning algorithms is quite frequent (Acid and de Campos, 2001; de Campos and Huete, 2000; Spirtes et al., 1993). It has also been used by algorithms based on score+search to stop the search process (Acid and de Campos, 2000; Herskovits and Cooper, 1990). The problem with an independence test is that it only asserts whether the

[8] $s_i$ should not be confused with $q_i$, which represents the number of configurations of these variables.

[9] There may obviously be some variables that cannot be included as parents of $X_i$ since they would create directed cycles in the graph.

variables are independent or not, rather than quantifying the extent to which they are. For example, if an algorithm is trying to decide which of the two variables $X_j$ and $X_k$ to exclude from the parent set of another variable $X_i$, if both variables turn out to be dependent on $X_i$ (given its current parent set), the test is not able to discriminate between them, although it may be possible for one variable to be more closely dependent on $X_i$ than the other.

Our proposal is to quantify the result of the independence test to build the scoring function. The difference $2NMI_D(X_i,X_{ij}|\{X_{i1},\ldots,X_{i(j-1)}\}) - \chi_{\alpha,l_{ij}}$ gives us a measure of the degree of interest for adding the variable $X_{ij}$ to the current parent set of $X_i$: if the difference is negative (the test would say that $X_i$ and $X_{ij}$ are independent), the score will decrease, and the more clearly independent the variables are, the more it will decrease; when the difference is positive (the test would assert that these two variables are dependent), the score will increase, and the more dependent $X_i$ and $X_{ij}$ are, the more it will increase.

Therefore, a measure of the global quality of the set $Pa_G(X_i)$ as the parent set of variable $X_i$ is:

$$\begin{aligned} g(X_i, Pa_G(X_i) : D) \quad &= \quad \sum_{j=2}^{s_i} \Big(2NMI_D(X_i,X_{ij}|\{X_{i1},\ldots,X_{i(j-1)}\}) - \chi_{\alpha,l_{ij}}\Big) \\ &\quad\; +2NMI_D(X_i,X_{i1}) - \chi_{\alpha,l_{i1}}, \end{aligned} \tag{15}$$

where $\chi_{\alpha,l_{ij}}$ is the value such that $p(\chi^2(l_{ij}) \le \chi_{\alpha,l_{ij}}) = \alpha$, and the number of degrees of freedom is:

$$l_{ij} = \begin{cases} (r_i-1)(r_{ij}-1)\prod_{k=1}^{j-1} r_{ik} & j = 2,\ldots,s_i \\ (r_i-1)(r_{i1}-1) & j = 1\,. \end{cases} \tag{16}$$

The expression in Equation 15 is then a global quantification of a series of $s_i$ simultaneous conditional independence tests, and by virtue of the decomposition of mutual information in Equation 14, it is equivalent to:

$$g(X_i, Pa_G(X_i) : D) = 2NMI_D(X_i, Pa_G(X_i)) - \sum_{j=1}^{s_i} \chi_{\alpha,l_{ij}}. \tag{17}$$

The scoring function would therefore be defined according to Equation 11 as:

$$g(G : D) = \sum_{\substack{i=1 \\ Pa_G(X_i)\neq\emptyset}}^{n} \Big(2NMI_D(X_i, Pa_G(X_i)) - \sum_{j=1}^{s_i} \chi_{\alpha,l_{ij}}\Big). \tag{18}$$

It should be noted that although the value of mutual information will increase after new variables are added to the parent set, the penalization component (which contains one term for each parent variable) will also increase. In this way, we are able to appropriately re-scale the mutual information measure.

The value of $\alpha$, which represents the confidence level associated with the statistical test, is a free parameter that may be fixed to any standard value (for example 0.90, 0.95 or 0.99). However, since we are in fact performing several simultaneous tests (as many as the number of variables in $Pa_G(X_i)$), and also taking into account the Bonferroni inequality,[10] in order for the *global* confidence level to be acceptable (that is to say, a reasonably high value of $p(\cap_{j=1}^{s_i}(\chi^2(l_{ij}) \le \chi_{\alpha,l_{ij}})))$, it will be necessary for $\alpha$ to be greater than the standard values used when performing a single test.

[10] $p(\cap_{i=1}^{n} A_i) \ge 1 - \sum_{i=1}^{n}\Big(1 - p(A_i)\Big)$, where $A_i$ represent any events.

In order to accurately compute the values $\chi_{\alpha,l}$, we can use a standard method which is based on the algorithm proposed by Hill and Pike (1965, 1985) to compute the chi-squared integral (i.e., the probability $p(\chi^2(l) > x)$) in combination with a simple bisection search. Alternatively, if speed is more important than great accuracy, as the $\chi^2(l)$ distribution can be approximated by several transformations of the standardized normal distribution $N(0,1)$ for large degrees of freedom (Evans et al., 1993), we can use tabulated exact values for $l \leq 100$ and the Wilson-Hilferty approximation (which is quite accurate) for $l > 100$:

$$\chi^2(l) \approx l\left[1 - \frac{2}{9l} + \sqrt{\frac{2}{9l}}N(0,1)\right]^3.$$

### 4.3 The MIT Score

Throughout the previous discussion, we have omitted one very important detail: the decomposition of mutual information that we have used (Equation 14) is not unique and we can decompose $MI_D(X_i, Pa_G(X_i))$ in many other ways - as many as the number of possible orderings of the variables in $Pa_G(X_i)$, that is, $s_i!$. Each corresponds to a different way of including the variables in the parent set of $X_i$ one at a time. The ordering does not affect the value $MI_D(X_i, Pa_G(X_i))$, but it can affect the penalization component (this will be the case whenever the number of states $r_{ik}$ of all the variables is not the same). By way of example, let us assume that $Pa_G(X_i) = \{X_1, X_2, X_3\}$. The six possible decompositions of $MI_D(X_i, \{X_1, X_2, X_3\})$ are:

$$\begin{array}{c}
MI_D(X_i,X_1) + MI_D(X_i,X_2|X_1) + MI_D(X_i,X_3|\{X_1,X_2\}) \\
MI_D(X_i,X_1) + MI_D(X_i,X_3|X_1) + MI_D(X_i,X_2|\{X_1,X_3\}) \\
MI_D(X_i,X_2) + MI_D(X_i,X_1|X_2) + MI_D(X_i,X_3|\{X_1,X_2\}) \\
MI_D(X_i,X_2) + MI_D(X_i,X_3|X_2) + MI_D(X_i,X_1|\{X_2,X_3\}) \\
MI_D(X_i,X_3) + MI_D(X_i,X_1|X_3) + MI_D(X_i,X_2|\{X_1,X_3\}) \\
MI_D(X_i,X_3) + MI_D(X_i,X_2|X_3) + MI_D(X_i,X_1|\{X_2,X_3\}).
\end{array}$$

Let us suppose that the number of states of the variables $X_i$, $X_1$, $X_2$ and $X_3$ is $r_i = 3$, $r_1 = 2$, $r_2 = 3$ and $r_3 = 4$. The penalization component in Equation 17 for each of the six previous decompositions is therefore:

$$\begin{array}{r}
\chi_{\alpha,2} + \chi_{\alpha,8} + \chi_{\alpha,36} = 107.93 \\
\chi_{\alpha,2} + \chi_{\alpha,12} + \chi_{\alpha,32} = 109.21 \\
\chi_{\alpha,4} + \chi_{\alpha,6} + \chi_{\alpha,36} = 108.91 \\
\chi_{\alpha,4} + \chi_{\alpha,18} + \chi_{\alpha,24} = 111.96 \\
\chi_{\alpha,6} + \chi_{\alpha,8} + \chi_{\alpha,32} = 111.07 \\
\chi_{\alpha,6} + \chi_{\alpha,16} + \chi_{\alpha,24} = 112.89.
\end{array}$$

The numerical values in these expressions are computed for the parameter $\alpha = 0.999$. It should be noted that the total number $\sum_{j=1}^{s_i} l_{ij}$ of degrees of freedom is always the same, 46 in this case, which would correspond to the degrees of freedom of a marginal independence test between $X_i$ and $Pa_G(X_i)$; such a test would use $(r_i - 1)(\prod_{j=1}^{s_i} r_{ij} - 1)$ degrees of freedom[11] (the value of $\chi_{\alpha,46}$

[11] Observe that $\sum_{j=1}^{s_i} l_{ij} = \sum_{j=1}^{s_i}\left((r_i - 1)(r_{ij} - 1)\prod_{k=1}^{j-1} r_{ik}\right) = (r_i - 1)(\prod_{j=1}^{s_i} r_{ij} - 1)$.

in the example is 81.40). In any case, the values are different since the chi-square distribution is not additive with respect to the number of degrees of freedom.[12] Therefore, depending on the selected ordering, the score in Equation 17 will be different. This is undesirable since the same DAG (depending on the path that the search process follows to reach it) would be evaluated differently. In order to solve this problem, we believe that the best we can do is to use the most conservative option, that is, to use the greatest of all these values so as to evaluate each parent set in the worst possible way.

In order to formalize this idea, let $\sigma_i = (\sigma_i(1), \ldots, \sigma_i(s_i))$ denote any permutation of the index set $(1, \ldots, s_i)$ of the variables in $Pa_G(X_i) = \{X_{i1}, \ldots, X_{is_i}\}$, and let us define:

$$l_{i\sigma_i(j)} = \begin{cases} (r_i - 1)(r_{i\sigma_i(j)} - 1)\prod_{k=1}^{j-1} r_{i\sigma_i(k)} & j = 2 \ldots, s_i \\ (r_i - 1)(r_{i\sigma_i(1)} - 1) & j = 1\ . \end{cases} \tag{19}$$

Then, instead of using Equation 17, the global quality measure of the set $Pa_G(X_i)$ that we propose is:

$$g(X_i, Pa_G(X_i) : D) = 2N\, MI_D(X_i, Pa_G(X_i)) - \max_{\sigma_i} \sum_{j=1}^{s_i} \chi_{\alpha, l_{i\sigma_i(j)}}.$$

The final expression of the proposed scoring function, which we shall call MIT (from mutual information tests), is:

$$g_{MIT}(G : D) = \sum_{\substack{i=1 \\ Pa_G(X_i) \neq \emptyset}}^{n} \left( 2N\, MI_D(X_i, Pa_G(X_i)) - \max_{\sigma_i} \sum_{j=1}^{s_i} \chi_{\alpha, l_{i\sigma_i(j)}} \right). \tag{20}$$

Computing each penalization component $\max_{\sigma_i} \sum_{j=1}^{s_i} \chi_{\alpha, l_{i\sigma_i(j)}}$ in the previous expression might seem to be a very time-consuming task since it would be necessary to evaluate all the $s_i!$ possible permutations of the variables in the set $Pa_G(X_i)$ in order to calculate the maximum. Fortunately, this will not be necessary as this maximum can be obtained in a much simpler way:

**Theorem 2** *For the values $l_{i\sigma_i(j)}$ defined in Equation 19,*

$$\max_{\sigma_i} \sum_{j=1}^{s_i} \chi_{\alpha, l_{i\sigma_i(j)}} = \sum_{j=1}^{s_i} \chi_{\alpha, l_{i\sigma_i^*(j)}},$$

*where $\sigma_i^*$ is any permutation of $Pa_G(X_i)$ satisfying $r_{i\sigma_i^*(1)} \geq r_{i\sigma_i^*(2)} \geq \ldots \geq r_{i\sigma_i^*(s_i)}$, whenever the function $f_{i,\alpha} : \mathcal{N}^{s_i} \longrightarrow \mathcal{R}$, defined as $f_{i,\alpha}(l_1, \ldots, l_{s_i}) = \sum_{j=1}^{s_i} \chi_{\alpha, l_j}$, is a Shur-concave function.*

This result says that the permutation that produces the maximum penalization value is the one where the first variable has the greatest number of states, the second variable has the second largest number of states, and so on. In the previously considered example, this permutation is $\{X_3, X_2, X_1\}$, and this reaches a maximum value equal to 112.89.

**Conjecture 3** *The function $f_{i,\alpha}$ defined in Theorem 2 is Shur-concave,* whenever $\alpha \geq 0.59$.

[12] With the exception of a sum of *independent* chi-square distributions, which obviously is not the case.

The combination of theoretical and empirical arguments that support this conjecture is included in the Appendix. The restriction concerning $\alpha$ does not represent any practical problem since we shall always use values of $\alpha$ which are much greater than 0.59.

Another way of measuring the quality of a set of variables $\mathbf{Z}$ as the parent set of $X_i$, which as it turns out is equivalent to the previous one, is as follows: we can consider that $\mathbf{Z}$ will be a good parent set if it continues to be a good parent set when one of its variables is removed, $\mathbf{Z}\setminus\{Y\}$, and also the variable $Y$ that we have removed should not have been removed, that is, $Y$ is not independent of $X_i$ given $\mathbf{Z}\setminus\{Y\}$. As we can do this for each variable in $\mathbf{Z}$, the final value should be the smallest one (we are again using a conservative or pessimistic view). This leads to a recursive definition of $g(X_i, Pa_G(X_i):D)$. The way of measuring the degree of undesirability of removing the variable $Y$ from $\mathbf{Z}$ is to use the difference between the mutual information statistic $2N\,MI_D(X_i,Y|\mathbf{Z}\setminus\{Y\})$ and the chi-square value $\chi_{\alpha,l}$ with the appropriate degrees of freedom. In this way, if $Y$ is truly independent on $X_i$ given $\mathbf{Z}\setminus\{Y\}$, then this difference will be negative and in this case we would prefer to use $\mathbf{Z}\setminus\{Y\}$ instead of $\mathbf{Z}$ as the parent set of $X_i$. If, on the contrary, the difference is positive, the set $\mathbf{Z}$ will be preferable to $\mathbf{Z}\setminus\{Y\}$.

We can therefore recursively define the score $g_r(X_i,Pa_G(X_i):D)$ in the following way:

$$g_r(X_i,Pa_G(X_i):D) = \min_{X_{ij}\in Pa_G(X_i)} \Big\{ g_r(X_i,Pa_G(X_i)\setminus\{X_{ij}\}:D) + 2N\,MI_D(X_i,X_{ij}|Pa_G(X_i)\setminus\{X_{ij}\}) - \chi_{\alpha,l^r_{ij}} \Big\}, \tag{21}$$

where $\chi_{\alpha,l^r_{ij}}$ is the value such that $p(\chi^2(l^r_{ij}) \le \chi_{\alpha,l^r_{ij}}) = \alpha$ and the number of degrees of freedom is $l^r_{ij} = (r_i-1)(r_{ij}-1)\prod_{k=1, k\neq j}^{s_i} r_{ik}$. The starting point of this recursive definition is obviously $g_r(X_i,\emptyset : D) = 0$. We can prove the following result:

**Theorem 4** *The MIT scoring function defined in Equation 20 can also be expressed as:*

$$g_{MIT}(G:D) = \sum_{\substack{i=1\\ Pa_G(X_i)\neq\emptyset}}^{n} g_r(X_i,Pa_G(X_i):D),$$

*where $g_r(X_i,Pa_G(X_i):D)$ are the local scores defined in Equation 21.*

Let us study some of the properties of the MIT score.

**Theorem 5** *The MIT scoring function defined in Equation 20 is decomposable.*

Unfortunately, MIT is not score-equivalent. Let us consider the following example: for the two DAGs $G_1$ and $G_2$ in Figure 1 and which are equivalent, let us suppose that the number of states of each variable is: $r_1 = 5$, $r_2 = 4$, $r_3 = 3$, $r_4 = 2$. Therefore:

$$\begin{aligned} g(G_1:D) &= 2N(MI_D(X_1,\{X_2,X_3\}) + MI_D(X_2,X_3) + MI_D(X_3,X_4)) \\ &\quad -(\chi_{\alpha,12}+\chi_{\alpha,32}+\chi_{\alpha,6}+\chi_{\alpha,2}) \\ g(G_2:D) &= 2N(MI_D(X_2,\{X_1,X_3\}) + MI_D(X_3,X_1) + MI_D(X_4,X_3)) \\ &\quad -(\chi_{\alpha,12}+\chi_{\alpha,30}+\chi_{\alpha,8}+\chi_{\alpha,2}). \end{aligned}$$

Although it seems that the part corresponding to mutual information is different in both cases, it is in fact not. It is sufficient to take into account Equation 12 and remember that the maximum

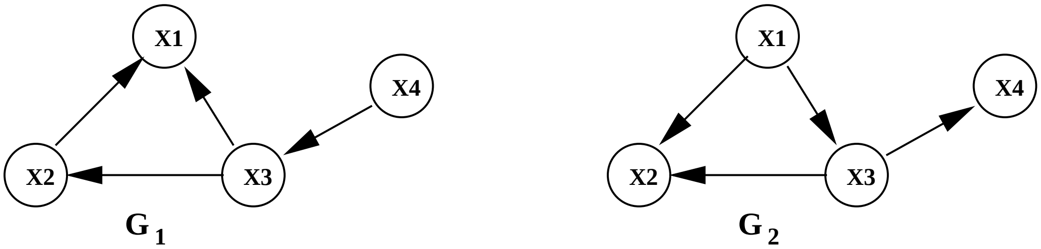

Figure 1: Two equivalent DAGs with different values of the MIT score

likelihood score is score-equivalent. The problem appears with the penalization by means of the sum of chi-square values: if the variables have a different number of states (as in this case), the results are different. More specifically, the penalization component is 131.67 for $G_1$ but 132.55 for $G_2$ (assuming that $\alpha = 0.999$).

The MIT score, however, satisfies a less demanding property than score-equivalence, and this concerns another type of space of equivalent DAGs, namely RPDAGs (Acid and de Campos, 2003). They are PDAGs which represent sets of equivalent DAGs, although they are not a canonical representation of equivalence classes of DAGs (two different RPDAGs may correspond to the same equivalence class). Let us introduce some additional notation and then the concept of RPDAG. The *skeleton* of a DAG is the undirected graph that results from ignoring the directionality of every arc. A *h-h pattern* (*head-to-head pattern*) in a DAG $G$ is an ordered triplet of nodes, $(X_i, X_k, X_j)$, such that $G$ contains the arcs $X_i \rightarrow X_k$ and $X_j \rightarrow X_k$. Given a PDAG $G = (\mathbf{U_n}, E_G)$, for each node $X_i \in \mathbf{U_n}$, $Sib_G(X_i) = \{X_j \in \mathbf{U_n} \mid X_i — X_j \in E_G\}$ is the set of *siblings* or *neighbors* of $X_i$. A PDAG $G$ is an RPDAG if and only if it satisfies the following conditions:

1. $\forall X_i \in \mathbf{U_n}$, if $Pa_G(X_i) \neq \emptyset$ then $Sib_G(X_i) = \emptyset$.
2. $G$ contains neither directed nor completely undirected cycles.
3. $\forall X_i, X_j \in \mathbf{U_n}$, if $X_j \in Pa_G(X_i)$ then either $|Pa_G(X_i)| \geq 2$ or $Pa_G(X_j) \neq \emptyset$.

The difference between essential graphs and RPDAGs appears when there are triangular structures: essential graphs may have completely undirected cycles, but these cycles must be *chordal* (Andersson et al., 1997). In other words, undirected cycles are forbidden in RPDAGs, whereas in essential graphs only undirected non-chordal cycles are forbidden. It can be seen that all the DAGs which are represented by a given RPDAG are equivalent and have the same skeleton and the same h-h patterns, whereas the DAGs associated with an essential graph have the same skeleton and the same v-structures (h-h patterns where the extreme nodes are not adjacent) (Pearl and Verma, 1990). Therefore, the role played by the v-structures in essential graphs is the same as that played by the h-h patterns in RPDAGs. The objective of RPDAGs is to trade the uniqueness of the representation of equivalence classes of DAGs for a more manageable one, because testing whether a given PDAG $G$ is an RPDAG is easier than testing whether $G$ is an essential graph.

**Theorem 6** *The MIT scoring function assigns the same value to all DAGs that are represented by the same RPDAG.*

Although the MIT score should not be used to search in the space of essential graphs, we can therefore use it without any problem to search in both the DAG and the RPDAG space.

To conclude our study of the new score, we have observed an interesting relation between MIT and the scoring functions based on Equation 6. First, it should be noted that the log-likelihood of the simplest possible network, namely the empty network $G_\emptyset$, is, according to Equation 4 (and taking into account that in this case $q_i = 1$ and $N_{ijk} = N_{ik}$):

$$LL_D(G_\emptyset) = \sum_{i=1}^{n}\sum_{k=1}^{r_i} N_{ik}\log\left(\frac{N_{ik}}{N}\right) = -N\sum_{i=1}^{n} H_D(X_i).$$

Then, considering Equation 12, we can express the sum of mutual information measures between each variable and its set of parents in $G$ as follows:

$$\sum_{\substack{i=1\\ Pa_G(X_i)\neq\emptyset}}^{n} MI_D(X_i, Pa_G(X_i)) = \frac{LL_D(G) - LL(G_\emptyset)}{N}.$$

Therefore, the sum of mutual information measures coincides with the difference between the log-likelihood of $G$ and the one of $G_\emptyset$ or, equivalently, with the difference between the description length of the data given $G_\emptyset$ and given $G$. Now, let us consider the difference between $G$ and $G_\emptyset$ in terms of complexity, which is:

$$C(G) - C(G_\emptyset) = \sum_{i=1}^{n}(r_i-1)q_i - \sum_{i=1}^{n}(r_i-1) = \sum_{\substack{i=1\\ Pa_G(X_i)\neq\emptyset}}^{n}(r_i-1)(q_i-1) = \sum_{\substack{i=1\\ Pa_G(X_i)\neq\emptyset}}^{n}\sum_{j=1}^{s_i} l_{ij},$$

with $l_{ij}$ defined as in Equation 16. Therefore, for the information-based scoring function defined in Equation 6, using $f(N) = 1/2$, the difference between the scores of $G$ and $G_\emptyset$ is:

$$\begin{aligned} g(G:D) - g(G_\emptyset:D) &= \Big(LL(G) - C(G)f(N)\Big) - \Big(LL(G_\emptyset) - C(G_\emptyset)f(N)\Big) \\ &= N\sum_{\substack{i=1\\ Pa_G(X_i)\neq\emptyset}}^{n} MI_D(X_i, Pa_G(X_i)) - \frac{1}{2}\sum_{\substack{i=1\\ Pa_G(X_i)\neq\emptyset}}^{n}\sum_{j=1}^{s_i} l_{ij} \\ &= \frac{1}{2}\sum_{\substack{i=1\\ Pa_G(X_i)\neq\emptyset}}^{n}\Big(2N\,MI_D(X_i, Pa_G(X_i)) - \sum_{j=1}^{s_i} l_{ij}\Big). \qquad (22) \end{aligned}$$

The similarity of this expression with those in Equations 18 and 20 is apparent. Therefore, the MIT score of a network $G$ could be interpreted in terms of the difference between the information-based scores of $G$ and $G_\emptyset$, and also as the decrease in description length achieved by using $G$ instead of $G_\emptyset$. By considering that the mean value of a $\chi^2$ distribution with $l$ degrees of freedom is just $l$, we can see that the MIT score appears when we replace in Equation 22 the mean values of the $\chi^2(l_{ij})$ distributions by the corresponding $\alpha$-quantiles.

## 5. Experimental Evaluation

In order to determine the possible merit of the proposed scoring function in practical terms, in this section we shall carry out an experimental evaluation of the MIT score, comparing it with other well-known scoring functions. The selected scoring functions are the most frequently used: K2 (Equation 1), BDeu (Equation 3) and BIC/MDL (Equation 5). For BDeu, we shall use a uniform prior distribution over possible structures and as this score is quite sensitive with respect to the value of the equivalent sample size, we shall use five values of this parameter, more precisely $\eta = 1, 2, 4, 8, 16$. For the single parameter of the MIT score (i.e., the confidence level), we shall use three values: $\alpha = 0.99, 0.999, 0.9999$.

The software necessary to carry out the experiments has been developed on the *Elvira* system (Elvira, 2002), a Java tool for building and using Bayesian networks and influence diagrams.

First, we define the performance criteria that we shall use to compare the different scoring functions.

### 5.1 Performance Criteria

One way of measuring the quality of a scoring function is to study its ability to *reconstruct* (in combination with a learning algorithm based on score+search) the Bayesian network which generated the data. In other words, we begin with a Bayesian network $G_0$ which is completely specified in terms of structure and parameters, and we obtain a data set of a given size by sampling from $G_0$. Then, using the scoring function together with a search method, we obtain a learned network $G$, which must be compared with the original network $G_0$. This capacity for reconstruction can be understood in two different but complementary ways: reconstructing the graphical structure and reconstructing the associated joint probability distribution. In terms of the first of these, the usual evaluation consists in measuring the structural differences between the original and the learned networks. More precisely, the number of added arcs ($A(G)$), deleted arcs ($D(G)$), and inverted arcs ($I(G)$) in the learned network with respect to the original one is computed. In order to eliminate fictitious differences or similarities between the two networks regarding the number of inverted arcs (caused by different but equivalent subDAG structures), before the two networks are compared they will be converted into their corresponding essential graph representation using the algorithm proposed by Chickering (1995). If $G'$ and $G'_0$ represent the essential graphs associated with $G$ and $G_0$, respectively, then the three measures of structural difference can be calculated using the following expressions:

$$A(G) = \frac{1}{2}\sum_{i=1}^{n} |Ad_{G'}(X_i) \setminus Ad_{G'_0}(X_i)|$$

$$D(G) = \frac{1}{2}\sum_{i=1}^{n} |Ad_{G'_0}(X_i) \setminus Ad_{G'}(X_i)|$$

$$I(G) = \sum_{i=1}^{n} \Big( |Pa_{G'_0}(X_i) \cap Sib_{G'}(X_i)| + |Pa_{G'}(X_i) \cap Sib_{G'_0}(X_i)| + |Pa_{G'_0}(X_i) \cap Ch_{G'}(X_i)| \Big).$$

where $Ch_H(X_i) = \{X_j \in \mathbf{U_n} \mid X_i \to X_j \in E_H\}$ and $Ad_H(X_i) = Pa_H(X_i) \cup Ch_H(X_i) \cup Sib_H(X_i)$ are the sets of children and adjacent nodes of $X_i$ in a PDAG $H$. As a way of summarizing these three measures, the *Hamming distance*, which is simply the sum of all the structural differences, $H(G) = A(G) + D(G) + I(G)$, is also usually considered.

In terms of the ability to reconstruct the joint probability distribution, we can evaluate this by means of a distance measure between the distributions associated with the original and the learned networks, $p_{G_0}$ and $p_G$, respectively. We shall use the Kullback-Leibler divergence:

$$KL(G) = KL(p_{G_0}, p_G) = \sum_{x_1,\ldots,x_n} p_{G_0}(x_1,\ldots,x_n) \log\left(\frac{p_{G_0}(x_1,\ldots,x_n)}{p_G(x_1,\ldots,x_n)}\right).$$

The conditional probability distributions that constitute the factorization of $p_G$ will be calculated from the data set using the Laplace estimation (Good, 1965), which avoids the problem of obtaining an infinite value of the Kullback-Leibler divergence, caused by zero probability values in $p_G$.

The calculus of this distance measure for joint distributions with many variables is computationally very expensive. However, by taking advantage of the factorization of the distributions, the complexity may be considerably reduced and the value $KL(G)$ can be expressed as follows:

$$\begin{aligned} KL(G) &= \sum_{i=1}^{n}\sum_{k=1}^{r_i}\sum_{j=1}^{q_i^{G_0}} p_{G_0}(x_{ik}, w_{ij}^{G_0}) \log(p_{G_0}(x_{ik}|w_{ij}^{G_0})) \\ &\quad -\sum_{i=1}^{n}\sum_{k=1}^{r_i}\sum_{j=1}^{q_i^{G}} p_{G_0}(x_{ik}, w_{ij}^{G}) \log(p_{G}(x_{ik}|w_{ij}^{G})), \end{aligned}$$

where $w_{ij}^{G_0}$ and $w_{ij}^{G}$ represent the $j$-th configuration of the parent sets of $X_i$ in $G_0$ and $G$, respectively (each having a total number of possible configurations equal to $q_i^{G_0}$ and $q_i^{G}$, respectively). In this way, the only probability values that must be computed are $p_{G_0}(x_{ik}, w_{ij}^{G_0})$ and $p_{G_0}(x_{ik}, w_{ij}^{G})$, and this can be done relatively efficiently by using a propagation algorithm in the network $G_0$. We have used an exact algorithm based on variable elimination.

One alternative way of measuring the quality of a scoring function which does not require an initial Bayesian network to be used as a starting point is to use the network learned with such a scoring function for a specific task and then to evaluate the level of success achieved. As Bayesian networks have been used in different ways to build classifiers, we can evaluate the quality of a scoring function (at least in comparative terms) by building a classifier using an algorithm for learning Bayesian networks which is specific for classification and equipped with the scoring function, and then measuring its classification capacity.

### 5.2 Experiments for Reconstructing Bayesian Networks

In order to make our comparative study more representative, we shall use different problems or rather different original networks. We shall also use different database sizes. Although this parameter clearly affects the quality of the networks learned with any scoring function (greater sizes lead to better estimations), we want to check which of the scoring functions may be more or less sensitive in the sense that their behavior deteriorates more quickly when smaller sample sizes are used.

In the following sections, we shall first give details of the experimental design before presenting the obtained results.

#### 5.2.1 EXPERIMENTAL DESIGN

We have selected four Bayesian networks corresponding to different problems: Alarm (Figure 2), Boblo (Figure 3), Insurance (Figure 4) and Hailfinder (Figure 5).

The Alarm network displays the relevant variables and relationships for the Alarm Monitoring System (Beinlich et al., 1989), a diagnostic application for patient monitoring. This network contains 37 variables and 46 arcs. Boblo (Rasmussen, 1995) is part of a system for determining the blood group of Jersey cattle. The Boblo network contains 23 variables and 24 arcs. Hailfinder (Abramson et al., 1996) is a normative system that forecasts severe summer hail in northeastern Colorado. The Hailfinder network contains 56 variables and 66 arcs. Insurance (Binder et al., 1997)

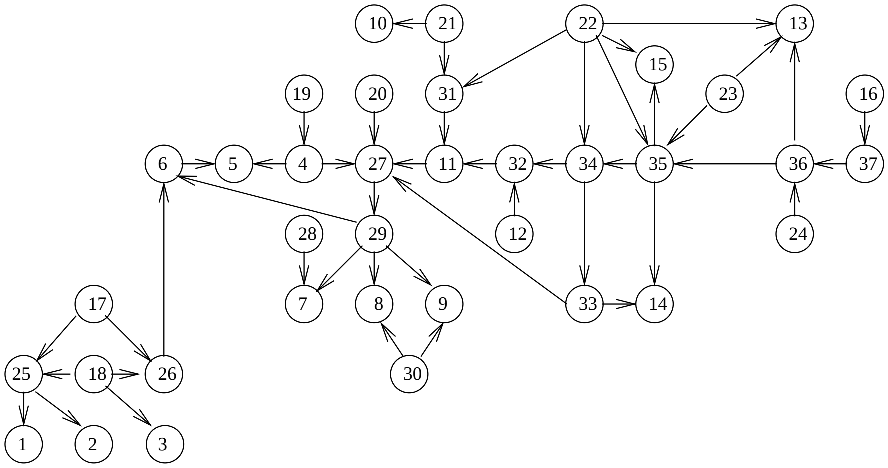

Figure 2: The Alarm network

is a network for evaluating car insurance risks. The Insurance network contains 27 variables and 52 arcs. All these networks have been widely used in specialist literature for comparative purposes.

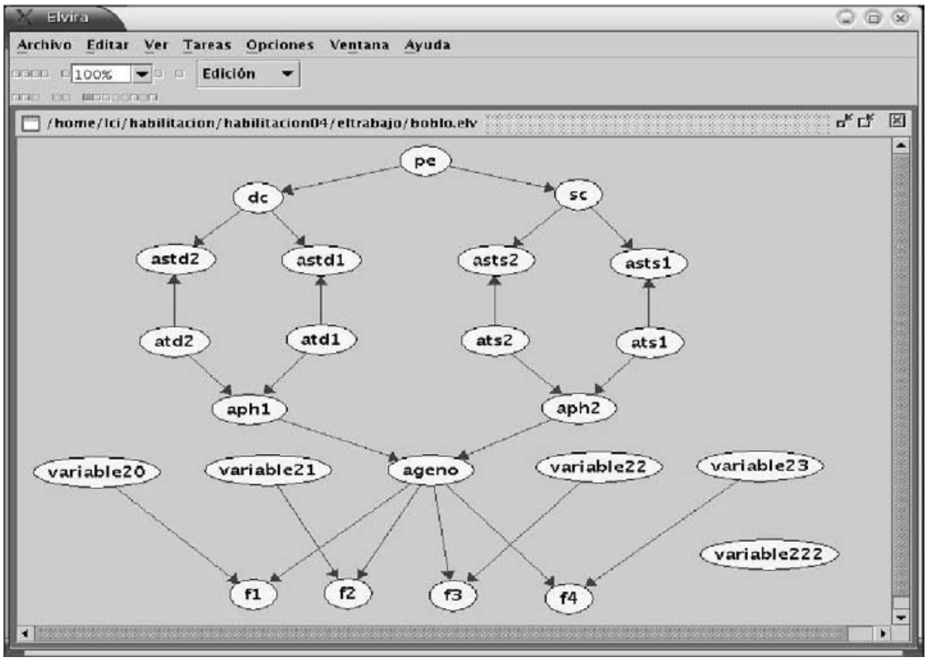

Figure 3: The Boblo network

Each network has been used to generate several databases, each of which contains 10000 instances; more precisely, we have generated five data sets for each problem. The results that we will show are the averages across the five data sets. The sample sizes considered are $N = 10000$, 5000 and 1000 (using the complete data sets and the first 5000 and 1000 instances of each one, respectively).

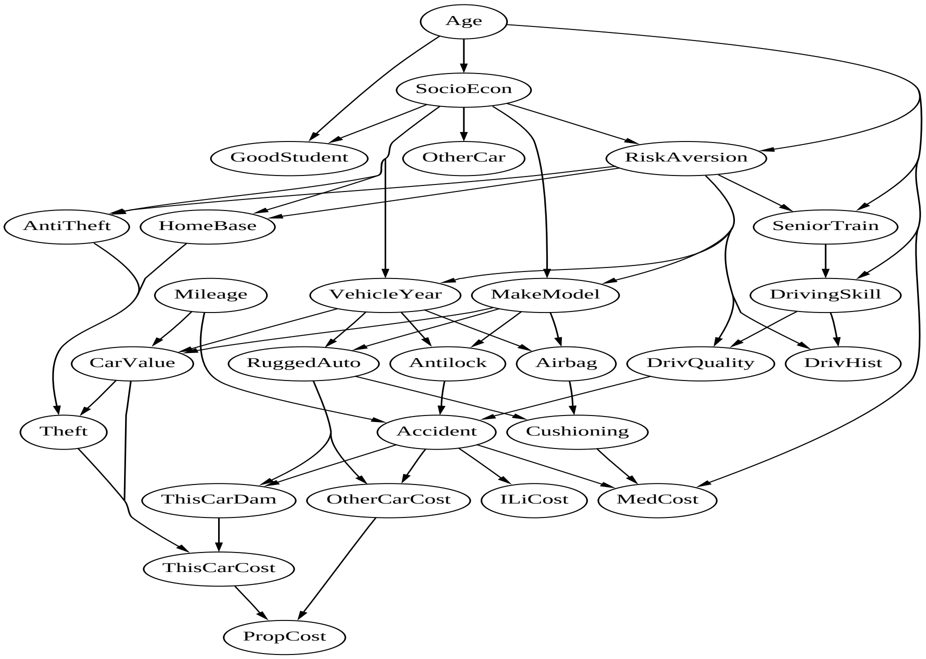

Figure 4: The Insurance network

The search method that we shall use is a local search in the DAG space with the classical operators of arc addition, arc deletion and arc reversal. The starting point of the search is always the empty graph. Although our main objective is to compare the proposed score with others, given that MIT has some similarities with constraint-based methods, it is also interesting to include one of these methods in the comparison. We have selected the well-known PC algorithm (Spirtes et al., 1993). This algorithm also depends on one parameter $\alpha$ representing the confidence level of the independence tests. We shall use three values: $\alpha = 0.90, 0.95, 0.99$.

We therefore have a design $13 \times 4 \times 3$ (10 scoring functions plus 3 versions of a constraint-based algorithm, 4 problems and 3 sample sizes), and for each of these 156 configurations we use 5 different databases, which gives us a total of 780 experiments.

### 5.2.2 RECONSTRUCTION RESULTS

Tables 1, 2, 3 and 4 display the results obtained for the Alarm, Boblo, Hailfinder and Insurance networks, respectively. For each sample size and each method, each table shows the average values of the previously mentioned performance measures (A, D, I, H and KL). The best value for each performance measure is written in bold and the second best in italics. In the last two rows of each table, we also show the KL values for the original network (with parameters re-trained from the corresponding database) and the empty network, which may serve as a kind of scale. Table 5 displays an illustrative summary of the results: it shows the number of times (from the 12 configurations being considered for each method) that each method has obtained the best result (and either the best or the second best result) for each of the five performance measures.

The first thing that can be observed is that these results seem to confirm our intuition about the need to use MIT with a greater confidence level $\alpha$ than those typically used for independence tests,

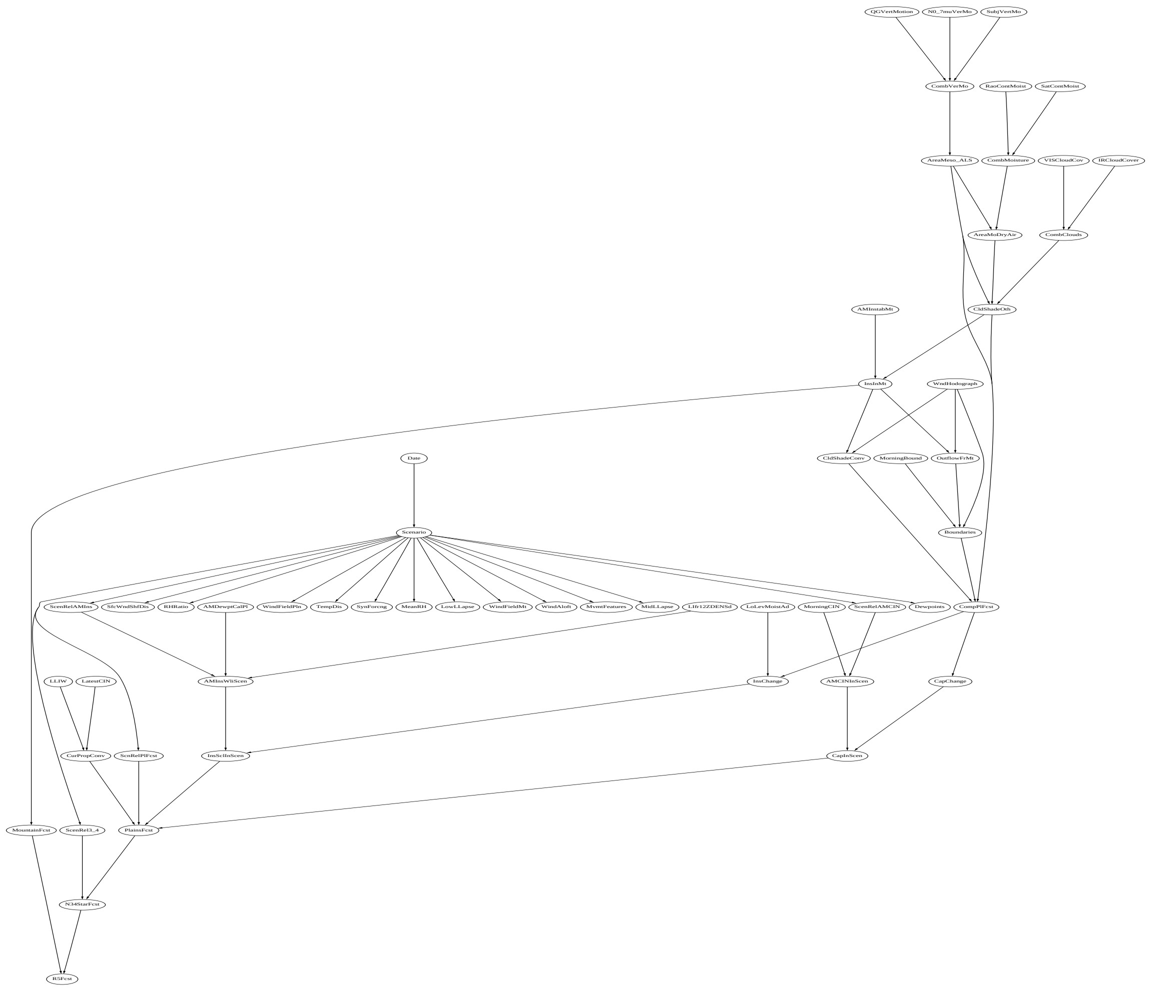

Figure 5: The Hailfinder network

since MIT with the values $\alpha = 0.999, 0.9999$ offers better results than with $\alpha = 0.99$. It is also possible to observe how MIT generally behaves better than the other scores, with respect to all the performance measures, and more specifically, in terms of BIC/MDL (which is the closest scoring function in spirit to the new score), MIT systematically obtains much better results. Although BIC behaves acceptably in terms of the number of added arcs, it does however have a marked propensity to remove a large number of arcs. This suggests that the penalization component used by BIC is not well calibrated. On the other hand, the different versions of BDeu behave rather poorly (except in terms of the number of deleted arcs). K2 only offers good results for the KL divergence. The PC algorithm behaves very good for the number of added and inverted arcs. However, its results in terms of the number of deleted arcs and KL divergence are extremely poor.

Focusing on the two main performance measures (the Hamming distance and the KL divergence), for each pair of methods, Tables 6 and 7 contain the number of times that each method obtains better results than the other. Table 6 refers to the KL divergence and Table 7 to the Hamming distance. In both cases, the MIT versions using high confidence levels (0.9999 and 0.999)

| ALARM | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *N* | 1000 | | | | | 5000 | | | | | 10000 | | | | |
| Score | A | D | I | H | KL | A | D | I | H | KL | A | D | I | H | KL |
| M9999 | 4.2 | 4.6 | 9.6 | *18.4* | 0.32752 | 4.6 | *2.4* | **4.6** | **11.6** | **0.06384** | 7.6 | *2.6* | 9.2 | 19.4 | **0.04372** |
| M999 | 4.2 | 4.0 | 9.4 | **17.6** | 0.31571 | 4.2 | 3.0 | **4.6** | *11.8* | *0.06448* | 9.8 | *2.6* | 10.0 | 22.4 | 0.04563 |
| M99 | 7.8 | 4.0 | 9.4 | 21.2 | *0.31270* | 8.4 | **2.0** | *4.8* | 15.2 | 0.06925 | 12.6 | **2.4** | 10.0 | 25.0 | 0.04743 |
| BIC | *7.2* | 7.4 | 20.0 | 34.6 | 0.49799 | 7.4 | 4.6 | 14.0 | 26.0 | 0.18683 | 9.6 | 3.4 | 18.2 | 31.2 | 0.09983 |
| K2 | 10.0 | 4.2 | 16.0 | 30.2 | **0.27079** | 8.4 | 3.2 | 14.2 | 25.8 | 0.07222 | 8.8 | 3.0 | 14.6 | 26.4 | *0.04375* |
| BD1 | 11.0 | 4.0 | 17.4 | 32.4 | 0.32570 | 9.6 | 3.2 | 13.4 | 26.2 | 0.08782 | 8.2 | 3.0 | 14.2 | 25.4 | 0.04855 |
| BD2 | 14.6 | 4.2 | 20.6 | 39.4 | 0.33198 | 11.0 | 2.8 | 15.0 | 28.8 | 0.09294 | 7.4 | *2.6* | 16.0 | 26.0 | 0.04387 |
| BD4 | 18.0 | **3.4** | 15.4 | 36.8 | 0.32044 | 11.6 | *2.4* | 17.6 | 31.6 | 0.06652 | 14.0 | 3.2 | 19.4 | 36.6 | 0.04797 |
| BD8 | 27.8 | 3.8 | 17.8 | 49.4 | 0.34363 | 16.8 | 2.6 | 16.0 | 35.4 | 0.07469 | 13.4 | **2.4** | 15.0 | 30.8 | 0.04491 |
| BD16 | 48.8 | *3.6* | 19.4 | 71.8 | 0.42465 | 31.8 | 3.0 | 15.2 | 50.0 | 0.09508 | 24.4 | 2.8 | 14.2 | 41.4 | 0.04582 |
| PC90 | 2.8 | 17.0 | **8.4** | 28.2 | 2.63819 | 0.6 | 9.0 | 5.4 | 15.0 | 1.21272 | *0.4* | 8.0 | **4.6** | **13.0** | 1.06377 |
| PC95 | *2.2* | 17.6 | **8.4** | 28.2 | 2.69645 | *0.4* | 9.2 | 5.4 | 15.0 | 1.29207 | **0.2** | 7.6 | *5.8* | *13.6* | 0.95810 |
| PC99 | **1.8** | 18.8 | *8.8* | 29.4 | 2.82810 | **0.2** | 10.6 | 6.0 | 16.8 | 1.63841 | *0.4* | 7.8 | 6.2 | 14.4 | 1.00228 |
| true | | | | | 0.21351 | | | | | 0.04759 | | | | | 0.02421 |
| empty | | | | | 10.2445 | | | | | 10.0677 | | | | | 10.0631 |

Table 1: Results for the Alarm network

| BOBLO | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *N* | 1000 | | | | | 5000 | | | | | 10000 | | | | |
| Score | A | D | I | H | KL | A | D | I | H | KL | A | D | I | H | KL |
| M9999 | *0.4* | 5.0 | *0.8* | *6.2* | 0.15105 | **0.0** | 2.2 | **0.0** | 2.2 | 0.03359 | 0.8 | *0.2* | **1.6** | **2.6** | *0.01396* |
| M999 | *0.4* | 4.4 | **0.4** | **5.2** | *0.14458* | *0.2* | 1.8 | **0.0** | *2.0* | 0.03266 | 0.8 | *0.2* | **1.6** | **2.6** | *0.01396* |
| M99 | 1.0 | 4.0 | 1.2 | *6.2* | 0.14812 | *0.2* | *1.6* | **0.0** | **1.8** | **0.03208** | 1.2 | **0.0** | **1.6** | *2.8* | **0.01353** |
| BIC | 2.0 | 6.4 | 4.6 | 13.0 | 0.16222 | 3.0 | 3.8 | *4.6* | 11.4 | 0.03651 | 2.8 | 2.4 | *3.0* | 8.2 | 0.01993 |
| K2 | 10.6 | 4.0 | 8.8 | 23.4 | **0.13805** | 11.0 | 2.6 | 7.6 | 21.2 | 0.03563 | 7.8 | 1.2 | 6.8 | 15.8 | 0.01748 |
| BD1 | 28.6 | *3.2* | 2.8 | 34.6 | 0.15329 | 13.4 | *1.6* | *4.6* | 19.6 | *0.03211* | 7.2 | 2.0 | 4.4 | 13.6 | 0.01481 |
| BD2 | 30.8 | **2.6** | 4.0 | 37.4 | 0.15452 | 21.2 | 2.2 | 7.2 | 30.6 | 0.03928 | 16.8 | 1.6 | 7.4 | 25.8 | 0.01705 |
| BD4 | 37.4 | **2.6** | 2.8 | 42.8 | 0.16213 | 28.0 | 1.8 | 4.8 | 34.6 | 0.03983 | 26.2 | 1.4 | 6.4 | 34.0 | 0.02065 |
| BD8 | 50.8 | 3.6 | 3.4 | 57.8 | 0.17616 | 41.2 | **1.4** | 5.2 | 47.8 | 0.04539 | 38.2 | 1.0 | 9.2 | 48.4 | 0.02317 |
| BD16 | 64.2 | **2.6** | 6.6 | 73.4 | 0.18015 | 54.0 | 2.0 | 6.0 | 62.0 | 0.05415 | 49.6 | 1.4 | 3.2 | 54.2 | 0.02830 |
| PC90 | **0.0** | 13.0 | 5.4 | 18.4 | 2.02929 | 0.8 | 10.0 | 6.2 | 17.0 | 1.44017 | 1.4 | 10.2 | 6.2 | 17.8 | 1.43512 |
| PC95 | **0.0** | 14.4 | 5.0 | 19.4 | 2.22612 | *0.2* | 10.0 | 6.0 | 16.2 | 1.43634 | *0.2* | 9.6 | 6.4 | 16.2 | 1.42543 |
| PC99 | **0.0** | 15.0 | 4.6 | 19.6 | 2.33032 | **0.0** | 10.8 | 5.6 | 16.4 | 1.50436 | **0.0** | 9.8 | 6.2 | 16.0 | 1.42574 |
| true | | | | | 0.13107 | | | | | 0.02712 | | | | | 0.01355 |
| empty | | | | | 7.44795 | | | | | 7.42898 | | | | | 7.42653 |

Table 2: Results for the Boblo network

compare favorably with the other scores. They systematically produce networks with much fewer structural differences with respect to the original networks and, at the same time, they almost always estimate the true joint probability distributions more closely. In terms of the Hamming distance, BIC is somewhat better than K2 and much better than BDeu, which systematically obtains worse results as the equivalent sample size increases. However, regarding the Kullback-Leibler divergence, K2 is much better than BIC and most of the versions of BDeu. The constraint-based algorithm is not able to find a good approximation of the joint probability distribution, probably because of the high number of deleted arcs together with the low number of added arcs.[13] In terms of the Hamming distance, PC performs better than all the Bayesian scores, although MIT and, to a lesser extent, BIC, outperform it.

[13] Extra arcs could be useful to compensate for the missing arcs.

| HAILFINDER | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *N* | 1000 | | | | | 5000 | | | | | 10000 | | | | |
| Score | A | D | I | H | KL | A | D | I | H | KL | A | D | I | H | KL |
| M9999 | *7.2* | 12.2 | *8.2* | **27.6** | **1.08438** | **8.0** | *5.8* | **4.2** | **18.0** | **0.26576** | **6.2** | *5.6* | **1.2** | **13.0** | **0.14678** |
| M999 | 8.6 | *11.0* | 8.6 | *28.2* | 1.13183 | 9.6 | **5.6** | *4.6* | *19.8* | 0.29131 | *7.6* | **5.4** | *1.6* | *14.6* | 0.16634 |
| M99 | 19.6 | **10.0** | **6.8** | 36.4 | 1.45014 | 21.2 | *5.8* | 8.8 | 35.8 | 0.47866 | 18.2 | 5.8 | 9.8 | 33.8 | 0.28220 |
| BIC | **6.4** | 16.2 | 15.0 | 37.6 | 1.36774 | 9.6 | 13.8 | 14.4 | 37.8 | 0.38606 | 10.0 | 10.2 | 17.2 | 37.4 | 0.21192 |
| K2 | 10.4 | 13.2 | 18.2 | 41.8 | *1.09179* | *9.0* | 8.6 | 22.0 | 39.6 | *0.27891* | 10.2 | 7.6 | 22.2 | 40.0 | *0.15910* |
| BD1 | 16.0 | 18.4 | 16.2 | 50.6 | 1.43422 | 17.0 | 13.0 | 21.4 | 51.4 | 0.40585 | 19.2 | 10.8 | 26.4 | 56.4 | 0.23520 |
| BD2 | 16.2 | 17.0 | 20.4 | 53.6 | 1.35804 | 19.2 | 12.6 | 20.6 | 52.4 | 0.35806 | 16.2 | 9.8 | 18.8 | 44.8 | 0.19763 |
| BD4 | 16.6 | 17.2 | 13.8 | 47.6 | 1.30878 | 18.4 | 13.2 | 18.0 | 49.6 | 0.36146 | 19.0 | 8.8 | 17.0 | 44.8 | 0.18702 |
| BD8 | 15.8 | 15.8 | 16.8 | 48.4 | 1.25347 | 20.2 | 12.0 | 20.4 | 52.6 | 0.33352 | 21.4 | 9.2 | 25.6 | 56.2 | 0.18622 |
| BD16 | 23.0 | 15.0 | 15.2 | 53.2 | 1.30559 | 22.8 | 10.4 | 15.0 | 48.2 | 0.33260 | 23.0 | 8.2 | 15.2 | 46.4 | 0.19391 |
| PC90 | 10.2 | 36.6 | 8.8 | 55.6 | 9.19075 | 14.8 | 33.4 | 7.0 | 55.2 | 8.38057 | 16.6 | 33.2 | 8.4 | 58.2 | 8.25173 |
| PC95 | 10.2 | 36.6 | 9.0 | 55.8 | 9.19961 | 13.8 | 33.2 | 6.8 | 53.8 | 8.38573 | 15.6 | 32.8 | 8.0 | 56.4 | 8.23382 |
| PC99 | 11.6 | 36.8 | 9.4 | 57.8 | 9.15348 | 13.8 | 33.4 | 6.6 | 53.8 | 8.32864 | 14.8 | 32.4 | 7.2 | 54.4 | 8.21041 |
| true | | | | | 1.18225 | | | | | 0.28146 | | | | | 0.14798 |
| empty | | | | | 20.6712 | | | | | 20.6048 | | | | | 20.5969 |

Table 3: Results for the Hailfinder network

| INSURANCE | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *N* | 1000 | | | | | 5000 | | | | | 10000 | | | | |
| Score | A | D | I | H | KL | A | D | I | H | KL | A | D | I | H | KL |
| M9999 | 3.4 | 14.8 | 13.4 | 31.6 | 0.50383 | 4.8 | 10.2 | 12.8 | 27.8 | 0.14468 | 3.8 | 7.2 | 6.4 | **17.4** | **0.06440** |
| M999 | 3.6 | *14.0* | 13.0 | *30.6* | 0.50499 | 5.0 | *9.4* | 12.2 | **26.6** | **0.14226** | 4.2 | *6.6* | 9.0 | *19.8* | *0.06653* |
| M99 | 3.8 | **12.2** | 13.4 | **29.4** | **0.45608** | 6.8 | **8.8** | 11.8 | *27.4* | 0.14513 | 4.6 | **6.4** | 14.0 | 25.0 | 0.06952 |
| BIC | 4.0 | 23.0 | 12.0 | 39.0 | 0.97628 | *4.4* | 14.8 | 15.8 | 35.0 | 0.25910 | 5.2 | 11.0 | 12.4 | 28.6 | 0.13403 |
| K2 | 9.2 | 17.0 | 19.4 | 45.6 | 0.52187 | 10.6 | 12.8 | 23.2 | 46.6 | 0.16905 | 10.4 | 11.8 | 21.4 | 43.6 | 0.10118 |
| BD1 | 6.2 | 17.2 | 13.8 | 37.2 | 0.57087 | 6.2 | 12.0 | 14.8 | 33.0 | 0.18197 | 7.2 | 10.6 | 19.0 | 36.8 | 0.12997 |
| BD2 | 5.6 | 14.8 | 14.2 | 34.6 | *0.48989* | 7.2 | 12.6 | 21.0 | 40.8 | 0.16623 | 8.8 | 11.0 | 18.6 | 38.4 | 0.13644 |
| BD4 | 9.4 | 15.0 | 19.0 | 43.4 | 0.50435 | 8.6 | 10.8 | 14.4 | 33.8 | 0.15113 | 6.0 | 8.4 | 16.4 | 30.8 | 0.08331 |
| BD8 | 16.2 | 16.4 | 17.8 | 50.4 | 0.53299 | 14.6 | 11.6 | 21.6 | 47.8 | 0.15281 | 10.2 | 9.2 | 13.2 | 32.6 | 0.09064 |
| BD16 | 22.2 | 14.6 | 19.6 | 56.4 | 0.58103 | 20.4 | 10.0 | 24.4 | 54.8 | *0.14247* | 18.8 | 7.6 | 19.8 | 46.2 | 0.08384 |
| PC90 | 2.0 | 30.6 | **8.8** | 41.4 | 2.31070 | **0.2** | 22.2 | **8.4** | 30.8 | 0.96871 | *0.2* | 19.4 | **4.8** | 24.4 | 0.58962 |
| PC95 | *1.8* | 30.6 | *9.0* | 41.4 | 2.31837 | **0.2** | 22.4 | *9.6* | 32.2 | 1.03911 | *0.2* | 19.6 | *5.0* | 24.8 | 0.57544 |
| PC99 | **1.4** | 31.2 | **8.8** | 41.4 | 2.42852 | **0.2** | 23.2 | 10.8 | 34.2 | 1.05543 | **0.0** | 20.0 | 5.4 | 25.4 | 0.62231 |
| true | | | | | 0.55527 | | | | | 0.12023 | | | | | 0.06205 |
| empty | | | | | 8.46596 | | | | | 8.44041 | | | | | 8.43720 |

Table 4: Results for the Insurance network

We believe that these results support the conclusion that the MIT score can compete favorably with state-of-the-art scoring functions and constraint-based algorithms for the task of learning general purpose Bayesian networks. Moreover, in the case that we wish to select a non-Bayesian scoring function based on information theory, we would recommend BIC/MDL be discarded and MIT used instead.

It is also interesting to remark that the two scoring functions that behave best (MIT and K2) are not score equivalent, whereas the two that obtain comparatively poor results (BIC and BDeu), are. Therefore, score equivalence does not seem to be an important property for learning Bayesian networks by searching in the DAG space. This confirms the previous results stated by Yang and Chang (2002).

While it is clear from the previous experiments that the new score, in combination with the particular search procedure being used, has an excellent performance, we would also like to test whether the different scores differentiate structures that are more accurate or generalize better, inde-

| | times best/times best or second best | | | | |
|---|---|---|---|---|---|
| Score | A | D | I | H | KL |
| M9999 | *3* / *5* | 0 / 5 | **5** / **7** | **6** / *8* | **6** / **7** |
| M999 | 0 / 3 | *2* / *8* | *4* / *6* | *4* / **11** | 1 / 5 |
| M99 | 0 / 1 | **7** / **9** | 3 / 4 | 2 / 5 | *3* / 4 |
| BIC | 1 / 2 | 0 / 0 | 0 / 2 | 0 / 0 | 0 / 0 |
| K2 | 0 / 1 | 0 / 0 | 0 / 0 | 0 / 0 | 2 / *6* |
| BD1 | 0 / 0 | 0 / 2 | 0 / 1 | 0 / 0 | 0 / 1 |
| BD2 | 0 / 0 | 1 / 2 | 0 / 0 | 0 / 0 | 0 / 1 |
| BD4 | 0 / 0 | *2* / 3 | 0 / 0 | 0 / 0 | 0 / 0 |
| BD8 | 0 / 0 | *2* / 2 | 0 / 0 | 0 / 0 | 0 / 0 |
| BD16 | 0 / 0 | 1 / 2 | 0 / 0 | 0 / 0 | 0 / 1 |
| PC90 | 2 / 4 | 0 / 0 | **5** / 5 | 1 / 1 | 0 / 0 |
| PC95 | *3* / **9** | 0 / 0 | 1 / 5 | 0 / 1 | 0 / 0 |
| PC99 | **8** / **9** | 0 / 0 | 1 / 2 | 0 / 0 | 0 / 0 |

Table 5: Number of times that each method obtained the best/the best or second best result in terms of each performance measure

| | Kullback-Leibler | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M9999 | M999 | M99 | K2 | BIC | BD1 | BD2 | BD4 | BD8 | BD16 | PC90 | PC95 | PC99 |
| M9999 | – | 7 | 7 | 10 | 12 | 10 | 11 | 11 | 12 | 11 | 12 | 12 | 12 |
| M999 | 4 | – | 8 | 6 | 12 | 11 | 10 | 11 | 11 | 12 | 12 | 12 | 12 |
| M99 | 5 | 4 | – | 6 | 9 | 9 | 8 | 8 | 8 | 7 | 12 | 12 | 12 |
| K2 | 2 | 6 | 6 | – | 12 | 10 | 9 | 8 | 10 | 10 | 12 | 12 | 12 |
| BIC | 0 | 0 | 3 | 0 | – | 3 | 2 | 2 | 3 | 3 | 12 | 12 | 12 |
| BD1 | 2 | 1 | 3 | 2 | 9 | – | 6 | 3 | 4 | 6 | 12 | 12 | 12 |
| BD2 | 1 | 2 | 4 | 3 | 10 | 6 | – | 6 | 6 | 7 | 12 | 12 | 12 |
| BD4 | 1 | 1 | 4 | 4 | 10 | 9 | 6 | – | 8 | 8 | 12 | 12 | 12 |
| BD8 | 0 | 1 | 4 | 2 | 9 | 8 | 6 | 4 | – | 9 | 12 | 12 | 12 |
| BD16 | 1 | 0 | 5 | 2 | 9 | 6 | 5 | 4 | 3 | – | 12 | 12 | 12 |
| PC90 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | – | 7 | 7 |
| PC95 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | – | 9 |
| PC99 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | – |

Table 6: Number of times that the methods in rows are better than the ones in columns in terms of the Kullback-Leibler divergence

pendently of the search issues. One way to do this is to generate an ensemble of networks that were found by the search procedures using the different scores and see how each of the scores rank the networks in this ensemble. So, for each of the sixty databases used in the previous experiments we have considered the ten networks obtained by the different scoring functions, computing the ranking of these networks according to each score. We have also computed the ranking of these networks according to each of the two main performance measures, the KL divergence and the Hamming distance.

| Hamming | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M9999 | M999 | M99 | K2 | BIC | BD1 | BD2 | BD4 | BD8 | BD16 | PC90 | PC95 | PC99 |
| M9999 | – | 5 | 8 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 11 | 11 | 11 |
| M999 | 6 | – | 9 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 11 | 11 | 11 |
| M99 | 3 | 3 | – | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 9 | 9 | 11 |
| K2 | 0 | 0 | 0 | – | 4 | 6 | 8 | 9 | 11 | 12 | 4 | 4 | 4 |
| BIC | 0 | 0 | 0 | 8 | – | 8 | 10 | 11 | 11 | 12 | 7 | 7 | 7 |
| BD1 | 0 | 0 | 0 | 6 | 4 | – | 10 | 8 | 9 | 10 | 5 | 4 | 5 |
| BD2 | 0 | 0 | 0 | 4 | 2 | 2 | – | 6 | 10 | 10 | 4 | 4 | 4 |
| BD4 | 0 | 0 | 0 | 3 | 1 | 4 | 5 | – | 11 | 11 | 3 | 3 | 3 |
| BD8 | 0 | 0 | 0 | 1 | 1 | 3 | 2 | 1 | – | 10 | 3 | 3 | 2 |
| BD16 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 1 | 2 | – | 3 | 3 | 3 |
| PC90 | 1 | 1 | 3 | 8 | 5 | 7 | 8 | 9 | 9 | 9 | – | 5 | 7 |
| PC95 | 1 | 1 | 3 | 8 | 5 | 7 | 8 | 9 | 9 | 9 | 4 | – | 8 |
| PC99 | 1 | 1 | 1 | 8 | 5 | 7 | 8 | 9 | 10 | 9 | 4 | 2 | – |

Table 7: Number of times that the methods in rows are better than the ones in columns in terms of the Hamming distance

To measure the degree of association between the rankings generated by each scoring function and each measure of performance, we have used the nonparametric Spearman correlation coefficient[14] for ordinal data (Hogg and Craig, 1994), which varies between $-1$ (perfect negative correlation) and $+1$ (perfect positive correlation).

Tables 8 and 9 display the average values of the Spearman coefficient with respect to Hamming distance and KL divergence, respectively, grouped by problem and database size.

| Average Spearman correlation w.r.t. Hamming distance | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Problem | | | | Database size | | | All |
| | Alarm | Boblo | Hailfinder | Insurance | 1000 | 5000 | 10000 | |
| M9999 | **0.69** | *0.97* | **0.74** | **0.69** | **0.83** | **0.72** | **0.77** | **0.77** |
| M999 | 0.62 | **0.98** | *0.71* | *0.68* | *0.81* | *0.70* | *0.73* | *0.75* |
| M99 | 0.53 | 0.96 | 0.66 | 0.65 | 0.77 | 0.65 | 0.68 | 0.70 |
| K2 | 0.55 | 0.63 | -0.02 | 0.21 | 0.32 | 0.27 | 0.44 | 0.34 |
| BIC | *0.67* | 0.93 | 0.60 | 0.61 | 0.75 | 0.64 | 0.72 | 0.70 |
| BD1 | 0.44 | 0.50 | -0.40 | 0.40 | 0.12 | 0.18 | 0.40 | 0.23 |
| BD2 | 0.41 | 0.29 | -0.39 | 0.42 | 0.06 | 0.12 | 0.35 | 0.18 |
| BD4 | 0.32 | -0.12 | -0.42 | 0.38 | -0.13 | -0.03 | 0.28 | 0.04 |
| BD8 | 0.20 | -0.59 | -0.48 | 0.35 | -0.27 | -0.17 | 0.06 | -0.13 |
| BD16 | -0.02 | -0.77 | -0.53 | 0.21 | -0.50 | -0.28 | -0.05 | -0.28 |

Table 8: Average values of the Spearman correlation coefficient between the rankings generated by each scoring function and the Hamming distance

These results confirm that, in terms of the KL divergence, MIT and K2 are the best scores (with K2 being in this case slightly better than MIT), whereas MIT and BIC are the best scores in terms of

[14] $\rho = 1 - \frac{6\sum_{i=1}^{N} d_i^2}{N(N^2-1)}$, where $\{d_i\}$ are the differences between the ranks of each observation on the two variables.

| Average Spearman correlation w.r.t. KL divergence | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Problem | | | | Database size | | | All |
| | Alarm | Boblo | Hailfinder | Insurance | 1000 | 5000 | 10000 | |
| M9999 | 0.80 | 0.72 | *0.51* | 0.77 | 0.66 | 0.68 | *0.76* | 0.70 |
| M999 | 0.83 | *0.74* | 0.47 | 0.80 | *0.71* | *0.69* | 0.74 | *0.71* |
| M99 | *0.85* | *0.74* | 0.34 | 0.82 | 0.70 | 0.66 | 0.71 | 0.69 |
| K2 | **0.92** | **0.81** | **0.55** | 0.70 | **0.76** | **0.70** | **0.77** | **0.74** |
| BIC | 0.48 | 0.65 | 0.33 | 0.30 | 0.34 | 0.44 | 0.55 | 0.44 |
| BD1 | 0.84 | 0.51 | -0.23 | 0.73 | 0.29 | 0.51 | 0.59 | 0.46 |
| BD2 | 0.84 | 0.38 | -0.17 | 0.79 | 0.28 | 0.52 | 0.58 | 0.46 |
| BD4 | 0.84 | 0.05 | -0.08 | *0.83* | 0.20 | 0.47 | 0.55 | 0.41 |
| BD8 | 0.79 | -0.37 | -0.01 | **0.85** | 0.17 | 0.39 | 0.38 | 0.31 |
| BD16 | 0.61 | -0.51 | -0.01 | *0.83* | 0.01 | 0.34 | 0.35 | 0.23 |

Table 9: Average values of the Spearman correlation coefficient between the rankings generated by each scoring function and the KL divergence

the Hamming distance (with MIT being better than BIC). In our opinion, the fact that MIT behaves very good in terms of both structural and distributional quality support the conclusion that it is a very competitive scoring function.

### 5.3 Experiments in Automatic Classification

As we commented previously, another approach to evaluating the quality of a scoring function is to use it to learn a Bayesian network classifier, and then to measure the performance of the classifier, for example in terms of predictive accuracy. In this section, we shall apply this method in order to compare MIT with the other scores.

Since the objective of a classifier is not to obtain a good representation of a joint probability distribution for the class and the attributes but rather one for the posterior probability distribution of the class given the attributes, several specialized algorithms that carry out the search into different types of restricted DAG topologies have been developed (Acid et al., 2005; Cheng and Greiner, 1999; Ezawa et al., 1996; Friedman, Geiger and Goldszmidt, 1997; Sahami, 1996), most of these being extensions (using augmenting arcs) or modifications of the well-known Naive Bayes basic topology. This approach generally obtains more satisfactory results than the algorithms for learning unrestricted types of Bayesian networks in terms of classification accuracy.

The BN learning algorithm that we shall use carries out a local search in a space of PDAGs called class-focused RPDAGs (C-RPDAGs), which are RPDAGs representing sets of DAGs which are equivalent in terms of classification (in the sense that they produce the same posterior probabilities for the class variable). Using the BDeu score, this algorithm has proved more effective than other Bayesian network classifiers (Acid et al., 2005).

As in the previous section, we shall first give details of the experimental design before going on to present the obtained results.

### 5.3.1 EXPERIMENTAL DESIGN

We have selected 29 data sets which were all obtained from the *UCI repository of machine learning databases* (Blake and Merz, 1998), with the exception of 'mofn-3-7-10' and 'corral', which were designed by Kohavi and John (1997). All these data sets have been widely used in specialist literature for comparative purposes in classification.

Table 10 briefly describes the characteristics of each database, including the number of instances, attributes and states for the class variable. Some of these data sets have been preprocessed in the following way: the continuous variables have been discretized using the procedure proposed by Fayyay and Irani (1993), and the instances with undefined/missing values were eliminated. For this preprocessing stage, we have used the MLC++ System (Kohavi et al., 1994).

| # | Database | N. cases | N. attributes | N. classes |
|---|---|---|---|---|
| 1 | adult | 45222 | 14 | 2 |
| 2 | australian | 690 | 14 | 2 |
| 3 | breast | 682 | 10 | 2 |
| 4 | car | 1728 | 6 | 4 |
| 5 | chess | 3196 | 36 | 2 |
| 6 | cleve | 296 | 13 | 2 |
| 7 | corral | 128 | 6 | 2 |
| 8 | crx | 653 | 15 | 2 |
| 9 | diabetes | 768 | 8 | 2 |
| 10 | flare | 1066 | 10 | 2 |
| 11 | german | 1000 | 20 | 2 |
| 12 | glass | 214 | 9 | 7 |
| 13 | glass2 | 163 | 9 | 2 |
| 14 | heart | 270 | 13 | 2 |
| 15 | hepatitis | 80 | 19 | 2 |
| 16 | iris | 150 | 4 | 3 |
| 17 | letter | 20000 | 16 | 26 |
| 18 | lymphography | 148 | 18 | 4 |
| 19 | mofn-3-7-10 | 1324 | 10 | 2 |
| 20 | mushroom | 8124 | 22 | 2 |
| 21 | nursery | 12960 | 8 | 5 |
| 22 | pima | 768 | 8 | 2 |
| 23 | satimage | 6435 | 36 | 6 |
| 24 | segment | 2310 | 19 | 7 |
| 25 | shuttle-small | 5800 | 9 | 7 |
| 26 | soybean-large | 562 | 35 | 19 |
| 27 | vehicle | 846 | 18 | 4 |
| 28 | vote | 435 | 16 | 2 |
| 29 | waveform-21 | 5000 | 21 | 3 |

Table 10: Description of the data sets used in the classification experiments

For each database and each scoring function, we have built a classifier using the algorithm based on C-RPDAGs. As in our previous experiments, the probability distributions associated with the obtained network structures have been computed from the data sets using the Laplace estimation.

The selected performance measure is predictive accuracy, that is, the percentage of successful predictions on a test set which is different from the training set. This accuracy has been measured as the average of three runs, the accuracy of each run being estimated using 10-fold cross-validation. Within each run, the cross-validation folds were the same for all the classifiers on each data set.[15] We used repeated runs and 10-fold cross-validation according to the recommendations by Kohavi (1995) in order to obtain a good balance between bias and variance of the estimation.

As these experiments are much more computationally expensive than those in the previous section, instead of using all the different versions of MIT and BDeu, we have selected only one. From the results in Tables 6 and 7, we believe that the best candidate scores are M9999 and BD4. We therefore have a $29 \times 4$ design (29 problems and 4 scoring functions), and for each of these 116 configurations, we carry out 3 iterations of 10-fold cross-validation, with a total of 3480 runs of the C-RPDAG learning algorithm.

### 5.3.2 CLASSIFICATION RESULTS

Table 11 displays the results of these experiments. The best results obtained for each problem are highlighted in bold. We can observe that there are no great differences between the different scoring functions (with the exception perhaps of BIC which seems to behave worst).

In order to determine whether the observed differences are statistically significant, we have also used a non-parametric statistical test: the Wilcoxon paired signed rank test, with a significance level equal to 0.01. We have used this test on each of the three cross-validation iterations. We shall then say that there is a significant difference if the Wilcoxon test detects a difference in at least one of the three iterations, and that there is a *very* significant difference if the test detects differences in all the three iterations. Table 11 also indicates whether the results obtained for K2, BIC and BDeu are significantly worse (–), very significantly worse (– –), significantly better (+) or very significantly better (++) than those of MIT for each data set.

In Table 12, we compare each classifier with the others according to these criteria. The entry in row $i$ column $j$ represents the number of times that classifier $i$ is significantly better or very significantly better than classifier $j$. These results confirm that K2, BDeu and MIT behave in a similar way, with MIT being slightly better, and that BIC is clearly the worst score.

## 6. Concluding Remarks

In this paper, we have defined a new scoring function for learning Bayesian networks through score+search algorithms. This is based on the well-known properties of the mutual information measure and which are used in a novel way. We begin with the idea of minimizing the Kullback-Leibler divergence between the joint probability distribution associated with a data set and the one associated with a Bayesian network, which is equivalent to maximizing the sum of the mutual information measures between each variable and its set of parents in the network. We then use a decomposition property of mutual information in order to express each of these measures as a sum of the conditional mutual information measures between the variable and each of its parents, given the subset of the remaining parent variables which antecede the current parent in a given order.

Using another mutual information property that allows us to build an independence test relying on the chi-square distribution, it is possible to interpret mutual information between a variable and

[15] The cross-validation folds are in fact the same as those considered by Acid et al. (2005).

| # | Database | K2 | BIC | BD4 | M9999 |
|---|---|---|---|---|---|
| 1 | adult | **85.71** | 85.42 (–) | 85.50 | 85.66 |
| 2 | australian | 85.65 | **86.28** | 85.27 | 85.22 |
| 3 | breast | **97.56** | **97.56** | 97.41 | 97.36 |
| 4 | car | 93.73 | 85.63 (– –) | 93.83 | **94.17** |
| 5 | chess | 96.50 | 95.81 | **96.71** (+) | 96.17 |
| 6 | cleve | 80.54 | **82.46** | 81.56 | 82.13 |
| 7 | corral | **100.00** | **100.00** | **100.00** | **100.00** |
| 8 | crx | 85.13 | **86.61** | 86.00 | 86.00 |
| 9 | diabetes | **78.65** | 78.56 | 78.60 | 78.60 |
| 10 | flare | 83.18 | 82.77 | **83.37** | 83.21 |
| 11 | german | 74.63 | 74.40 | **74.87** | 74.23 |
| 12 | glass | 71.57 | 70.12 | 71.56 | **71.85** |
| 13 | glass2 | **85.45** | 84.83 | 85.22 | 85.44 |
| 14 | heart | 82.47 | 82.59 | **83.21** | 82.59 |
| 15 | hepatitis | 90.83 | 87.50 | **92.50** | 90.00 |
| 16 | iris | 93.33 | 94.22 | **94.44** | 94.22 |
| 17 | letter | **85.99** (+) | 76.73 (– –) | 85.55 | 85.45 |
| 18 | lymphography | 82.83 | 81.78 | **83.49** | 81.25 |
| 19 | mofn-3-7-10 | 97.36 (–) | 93.56 (– –) | 99.09 | **100.00** |
| 20 | mushroom | **100.00** | **100.00** | **100.00** | **100.00** |
| 21 | nursery | 94.71 (– –) | 91.30 (– –) | 93.38 (– –) | **95.45** |
| 22 | pima | **78.86** | 78.51 | 78.21 | 78.43 |
| 23 | satimage | 87.84 (–) | 84.57 (– –) | 88.32 | **88.51** |
| 24 | segment | 94.92 | 92.16 (– –) | 94.55 | **95.11** |
| 25 | shuttle-small | 99.67 | **99.79** | 99.60 | 99.65 |
| 26 | soybean-large | **93.30** | 88.85 (–) | 92.64 | 91.81 |
| 27 | vehicle | **72.46** | 71.75 | 72.10 | 72.26 |
| 28 | vote | **94.79** | 92.95 | 93.72 | 94.03 |
| 29 | waveform-21 | 82.47 | 82.47 | **83.06** | 82.21 |
|  | Average | 87.94 | 86.52 | **88.06** | 87.97 |

Table 11: Predictive accuracy of the different scoring functions

|  | K2 | BIC | BD4 | M9999 |
|---|---|---|---|---|
| K2 | — | 9 / **5** | 2 / **1** | 1 / **0** |
| BIC | 0 / **0** | — | 1 / **0** | 0 / **0** |
| BD4 | 3 / **1** | 8 / **6** | — | 1 / **0** |
| M9999 | 3 / **1** | 8 / **6** | 1 / **1** | — |

Table 12: Number of times that the classifiers in rows are significantly better / **very significantly better** than the ones in columns

its parents as a sum of the statistics associated with a set of simultaneous conditional independence tests. Each of these tests indicates whether it is worth adding a new parent, taking into account those parents which have already been included. The value of each statistic is compared with a reference value, and the sum of the differences between statistics and reference values is used to quantify the global quality of the parent set. The result is a scoring function (called MIT) which is similar to those based on maximizing a penalized version of the log-likelihood, such as BIC/MDL. In our case, however, the penalization component is specific rather than global for each variable and its parents, and takes into account not only the complexity of the structure but also its reliability. Although MIT is a scoring function, the result of using it within an algorithm based on score and search has many similarities with learning algorithms based on independence tests. However, in our case, the tests are not only used to decide whether the variables are independent or not, but they also quantify the extent to which they are.

We have also carried out a complete experimental evaluation of the proposed score, comparing it with state-of-the-art scoring functions (such as K2, BDeu and BIC/MDL) and with a constraint-based algorithm using different evaluation criteria: structural differences between the original and the learned networks, distance between the probability distributions associated with these networks, and predictive accuracy of the classifiers constructed using the different scores. The results of these experiments show that MIT can compete with the Bayesian scores and that it should be the score of reference within those based on information theory.

The MIT scoring function is decomposable and is not score equivalent, although it satisfies a restricted form of score equivalence which allows us to use it to search not only in the DAG space but also in the RPDAG space. Nevertheless, for future research we would like to develop a scoring function which is based on the same MIT principles but which satisfies the score equivalence property, to be used by learning algorithms that search in the space of essential graphs. Furthermore, the expression of the MIT score depends on a free parameter: the confidence level $\alpha$ associated with the chi-square independence tests. Although experimental results confirm our previous analysis which states that this parameter should be set to a high value (much higher than is usual for a single statistical test), it would also be interesting to find some guidelines in order to automatically select an appropriate value of $\alpha$ depending on the characteristics of the problem domain being considered.

## Acknowledgments

I would like to acknowledge support for this work from the Spanish ‘Consejería de Innovación Ciencia y Empresa de la Junta de Andalucía’, under Project TIC-276. I am grateful to the entire Elvira system development team, especially to my colleagues Silvia Acid, Javier G. Castellano, Serafín Moral and José M. Puerta. Their collaboration in previous work and their contributions to the Elvira system have made the experimental part of this paper possible. I am also grateful to the anonymous reviewers for useful comments and suggestions.

## Appendix A

**Proof of Theorem 2**. We should first explain what a Shur-concave function is. Let us consider two n-dimensional vectors $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$, and let $\mathbf{x}^{\downarrow} = (x_1^{\downarrow}, \ldots, x_n^{\downarrow})$ and $\mathbf{y}^{\downarrow} = (y_1^{\downarrow}, \ldots, y_n^{\downarrow})$ be the vectors whose entries are the entries of $\mathbf{x}$ and $\mathbf{y}$, arranged in decreasing order, that is, $x_1^{\downarrow} \geq x_2^{\downarrow} \geq \ldots \geq x_n^{\downarrow}$ and $y_1^{\downarrow} \geq y_2^{\downarrow} \geq \ldots \geq y_n^{\downarrow}$. If $\sum_{j=1}^{m} x_j^{\downarrow} \leq \sum_{j=1}^{m} y_j^{\downarrow}$ $\forall m \leq n$, then it is said that $\mathbf{x}$ is majorized by $\mathbf{y}$, written $\mathbf{x} \prec \mathbf{y}$. A function $f : \mathcal{N}^n \longrightarrow \mathcal{R}$ is Shur-concave if for every vector $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ such that $\mathbf{x} \prec \mathbf{y}$, then $f(x_1, \ldots, x_n) \geq f(y_1, \ldots, y_n)$. This is one of the essential properties of entropy and establishes that the more uniform a distribution is, the greater the entropy.

Let us assume that the function $f_{i,\alpha}(l_1,\dots,l_{s_i}) = \sum_{j=1}^{s_i} \chi_{\alpha,l_j}$ is Shur-concave, and we shall prove the result stated in the theorem. For any permutation $\sigma_i$, let us consider the vector $\mathbf{l_{i\sigma_i}} = (l_{i\sigma_i(1)},\dots, l_{i\sigma_i(s_i)})$. As $r_{ik} \geq 2 \forall k$, then $l_{i\sigma_i(j)} = (r_i-1)(r_{i\sigma_i(j)}-1)\prod_{k=1}^{j-1} r_{i\sigma_i(k)} \leq (r_i-1)r_{i\sigma_i(j)}\prod_{k=1}^{j-1} r_{i\sigma_i(k)} \leq (r_i-1)\ (r_{i\sigma_i(j+1)}-1)r_{i\sigma_i(j)}\prod_{k=1}^{j-1} r_{i\sigma_i(k)} = (r_i-1)(r_{i\sigma_i(j+1)}-1)\prod_{k=1}^{j} r_{i\sigma_i(k)} = l_{i\sigma_i(j+1)}$. Therefore $l_{i\sigma_i(s_i)} \geq \dots \geq l_{i\sigma_i(2)} \geq l_{i\sigma_i(1)}$, that is, $l^{\downarrow}_{i\sigma_i(1)} = l_{i\sigma_i(s_i)}$,.., $l^{\downarrow}_{i\sigma_i(s_i)} = l_{i\sigma_i(1)}$.

Then, the values of $\sum_{j=1}^{m} l^{\downarrow}_{i\sigma_i(j)}$ can be expressed as follows:

$$\sum_{j=1}^{m} l^{\downarrow}_{i\sigma_i(j)} = \sum_{j=s_i-m+1}^{s_i} l_{i\sigma_i(j)} = \sum_{j=s_i-m+1}^{s_i}\left((r_i-1)(r_{i\sigma_i(j)}-1)\prod_{k=1}^{j-1} r_{i\sigma_i(k)}\right)$$

$$= (r_i-1)\sum_{j=s_i-m+1}^{s_i}\left(r_{i\sigma_i(j)}\prod_{k=1}^{j-1} r_{i\sigma_i(k)} - \prod_{k=1}^{j-1} r_{i\sigma_i(k)}\right) = (r_i-1)\sum_{j=s_i-m+1}^{s_i}\left(\prod_{k=1}^{j} r_{i\sigma_i(k)} - \prod_{k=1}^{j-1} r_{i\sigma_i(k)}\right)$$

$$= (r_i-1)\left(\prod_{k=1}^{s_i} r_{ik} - \prod_{k=1}^{s_i-m} r_{i\sigma_i(k)}\right).$$

As the permutation $\sigma_i^*$ ranks the variables in decreasing order of the number of states, $\prod_{k=1}^{s_i-m} r_{i\sigma_i(k)} \leq \prod_{k=1}^{s_i-m} r_{i\sigma_i^*(k)}$ and therefore $\sum_{j=1}^{m} l^{\downarrow}_{i\sigma_i^*(j)} \leq \sum_{j=1}^{m} l^{\downarrow}_{i\sigma_i(j)}$, that is, $\mathbf{l_{i\sigma_i^*}} \prec \mathbf{l_{i\sigma_i}}$. By applying the Shur-concavity of $f_{i,\alpha}$, we then obtain $\sum_{j=1}^{s_i} \chi_{\alpha,l_{i\sigma_i(j)}} \leq \sum_{j=1}^{s_i} \chi_{\alpha,l_{i\sigma_i^*(j)}} \forall \sigma_i$, hence $\sum_{j=1}^{s_i} \chi_{\alpha,l_{i\sigma_i^*(j)}} = \max_{\sigma_i} \sum_{j=1}^{s_i} \chi_{\alpha,l_{i\sigma_i(j)}}$. ■

**Argument supporting Conjecture 3**. We try to prove that the functions $f_{i,\alpha}$ are Shur-concave. We shall use the well-known result (Marshall and Olkin, 1979) which states that $\mathbf{x} \prec \mathbf{y}$ if and only if $F(\mathbf{x}) \geq F(\mathbf{y})$, where $F(\mathbf{x}) = \sum_{i=1}^{n} g(x_i)$, for all concave functions $f$. In our case $F(\mathbf{l}) = f_{i,\alpha}(l_1,\dots,l_{s_i}) = \sum_{j=1}^{s_i} \chi_{\alpha,l_j}$, so that we must only prove that the function $f_\alpha(l) = \chi_{\alpha,l}$ is concave in order to obtain the result. A function $f(l)$ is concave if and only if $\forall l_1 \leq l_2 \leq l_3$, $\frac{f(l_2)-f(l_1)}{l_2-l_1} \geq \frac{f(l_3)-f(l_1)}{l_3-l_1}$, which is equivalent to

$$\forall h,k \geq 0, \forall l, (h+k)f(l) \geq kf(l+h)+hf(l-k).$$

We could prove the concavity of $f$ by using induction on the 'distances' $h$ and $k$. The base case is $h=k=1$, that is,

$$2f(l) \geq f(l+1)+f(l-1), \forall l. \tag{23}$$

Let us assume that $\forall h \leq h_0, \forall k \leq k_0$, with $k_0 \leq h_0$, $(h+k)f(l) \geq kf(l+h)+hf(l-k) \forall l$. For the values [$l$, $h=h_0$, $k=k_0$], we then obtain

$$(h_0+k_0)f(l) \geq k_0 f(l+h_0)+h_0 f(l-k_0). \tag{24}$$

Using the values [$l-k_0$, $h=k_0$, $k=1$], we now obtain

$$(k_0+1)f(l-k_0) \geq f(l)+k_0 f(l-k_0-1).$$

Simple algebraic manipulations of these two inequalities lead to $(h_0+k_0+1)f(l) \geq (k_0+1)f(l+h_0)+h_0 f(l-k_0-1)$.

Similarly, using the values $[l+h_0,\ h=1,\ k=h_0]$ instead of $[l-k_0,\ h=k_0,\ k=1]$, we obtain

$$(h_0+1)f(l+h_0) \geq h_0 f(l+h_0+1)+f(l). \tag{25}$$

Once again, after algebraic manipulations of the inequalities (24) and (25), we obtain $(h_0+k_0+1)f(l) \geq k_0 f(l+h_0+1)+(h_0+1)f(l-k_0)$. The induction step is therefore complete.

We must still prove the base case. Unfortunately, we have not been able to analytically prove the inequality in Equation 23 when $f(l)=f_\alpha(l)=\chi_{\alpha,l}$. Therefore, in order to prove it empirically, we have built a computer program that computes the values $\chi_{\alpha,l}$ and tests the truth of the inequality. It is obvious that while we cannot compute $\chi_{\alpha,l}$ for all the values of $l$ and $\alpha$, we can for all the values of practical interest. More specifically, we have tested all the values of $l$ from 2 to 1000 and all the values of $\alpha$ from 0.1000 to 0.9999 with a stepsize of 0.0001. The results of these experiments are as follows: the inequality in Equation 23 is always true from $\alpha=0.5827$ to 0.9999; from $\alpha=0.5429$ to 0.5826, it is always true except for the case $l=2$; from $\alpha=0.4922$ to 0.5428, the inequality is false for many values of $l$ (the lower $\alpha$ is, the more frequent the number of failures), and from $\alpha=0.1000$ to 0.4921 it is always false. It can be seen that since the behavior of the function $f_\alpha(l)$ is quite homogeneous, we do not expect it to behave differently for the intermediate values of $\alpha$ which have not been tested. We may therefore conclude that $f_\alpha(l)$ is concave for all the values of $\alpha$ that may be of interest when computing the MIT score. ■

**Proof of Theorem 4**. We shall use induction on the number of variables in $Pa_G(X_i)$. The base case, where $|Pa_G(X_i)|=1$, is obviously true. Let us suppose that the result is true when the size of the parent set of $X_i$ is equal to $s_i-1$ and consider a case where $|Pa_G(X_i)|=s_i$. Then, if $\sigma_{ij}$ denotes a permutation of the variables in the set $Pa_G(X_i)\setminus\{X_{ij}\}$, we have

$$\begin{aligned}
g_r(X_i,Pa_G(X_i):D) &= \min_{X_{ij}\in Pa_G(X_i)} \Big\{ g_r(X_i,Pa_G(X_i)\setminus\{X_{ij}\}:D)+ \\
&\qquad\qquad 2N\,MI_D(X_i,X_{ij}|Pa_G(X_i)\setminus\{X_{ij}\})-\chi_{\alpha,l^r_{ij}} \Big\} \\
&= \min_{X_{ij}\in Pa_G(X_i)} \Big\{ 2N\,MI_D(X_i,Pa_G(X_i)\setminus\{X_{ij}\}) - \max_{\sigma_{ij}} \sum_{k=1}^{s_i-1} \chi_{\alpha,l_{i\sigma_{ij}(k)}} + \\
&\qquad\qquad 2N\,MI_D(X_i,X_{ij}|Pa_G(X_i)\setminus\{X_{ij}\})-\chi_{\alpha,l^r_{ij}} \Big\} \\
&= \min_{X_{ij}\in Pa_G(X_i)} \Big\{ 2N\,MI_D(X_i,Pa_G(X_i)) - \max_{\sigma_{ij}} \sum_{k=1}^{s_i-1} \chi_{\alpha,l_{i\sigma_{ij}(k)}} - \chi_{\alpha,l^r_{ij}} \Big\} \\
&= 2N\,MI_D(X_i,Pa_G(X_i)) - \max_{X_{ij}\in Pa_G(X_i)} \Big\{ \max_{\sigma_{ij}} \sum_{k=1}^{s_i-1} \chi_{\alpha,l_{i\sigma_{ij}(k)}} + \chi_{\alpha,l^r_{ij}} \Big\} \\
&= 2N\,MI_D(X_i,Pa_G(X_i)) - \max_{X_{ij}\in Pa_G(X_i)} \Big\{ \max_{\sigma_{ij}} \Big\{ \sum_{k=1}^{s_i-1} \chi_{\alpha,l_{i\sigma_{ij}(k)}} + \chi_{\alpha,l^r_{ij}} \Big\} \Big\}.
\end{aligned}$$

The value $\sum_{k=1}^{s_i-1}\chi_{\alpha,l_{i\sigma_{ij}(k)}}+\chi_{\alpha,l^r_{ij}}$ in the last expression can be seen as the value associated with a permutation of the variables in $Pa_G(X_i)$ where the last element is restricted to be $X_{ij}$, that is, if we define a permutation $\sigma_{i\setminus j}$ as $\sigma_{i\setminus j}(k)=\sigma_{ij}(k),\ \forall k=1,\ldots,s_i-1$ and $\sigma_{i\setminus j}(s_i)=j$, then $\sum_{k=1}^{s_i-1}\chi_{\alpha,l_{i\sigma_{ij}(k)}}+\chi_{\alpha,l^r_{ij}}=\sum_{k=1}^{s_i}\chi_{\alpha,l_{i\sigma_{i\setminus j}(k)}}$.

The union of the sets of permutations of $Pa_G(X_i)$ where the last element is fixed to $X_{ij}$, for all $X_{ij}$, is the set of all the permutations of $Pa_G(X_i)$, hence

$$\max_{X_{ij}\in Pa_G(X_i)} \max_{\sigma_{ij}} \left\{ \sum_{k=1}^{s_i-1} \chi_{\alpha,l_{i\sigma_{ij}(k)}} + \chi_{\alpha,l_{ij}^r} \right\} = \max_{X_{ij}\in Pa_G(X_i)} \max_{\sigma_{i\setminus j}} \sum_{k=1}^{s_i} \chi_{\alpha,l_{i\sigma_{i\setminus j}(k)}} = \max_{\sigma_i} \sum_{k=1}^{s_i} \chi_{\alpha,l_{i\sigma_i(k)}}.$$

Therefore, we have $g_r(X_i, Pa_G(X_i) : D) = 2N\, MI_D(X_i, Pa_G(X_i)) - \max_{\sigma_i} \sum_{k=1}^{s_i} \chi_{\alpha,l_{i\sigma_i(k)}}$ and the result is also true for parent sets of $X_i$ with size equal to $s_i$. This completes the induction step. ■

**Proof of Theorem 5**. This result is evident as the scoring function is, by definition, a sum of local scores. ■

**Proof of Theorem 6**. As all DAGs that are represented by the same RPDAG have the same skeleton and the same head-to-head patterns (either coupled or uncoupled), then the differences between these DAGs can only be due to the different direction of certain arcs linking two nodes $X_i$ and $X_j$ that have at most a single parent. In such cases, the chi-square value associated with the local score of the corresponding node (either $X_i$ or $X_j$) is always the same, $\chi_{\alpha,l}$, with $l = (r_i - 1)(r_j - 1)$. ■

## References

B. Abramson, J. Brown, A. Murphy, and R. L. Winkler. Hailfinder: A Bayesian system for forecasting severe weather. *International Journal of Forecasting*, 12:57–71, 1996.

S. Acid and L. M. de Campos. Learning right sized belief networks by means of a hybrid methodology. *Lecture Notes in Artificial Intelligence*, 1910:309–315, 2000.

S. Acid and L. M. de Campos. A hybrid methodology for learning belief networks: Benedict. *International Journal of Approximate Reasoning*, 27:235–262, 2001.

S. Acid and L. M. de Campos. Searching for Bayesian network structures in the space of restricted acyclic partially directed graphs. *Journal of Artificial Intelligence Research*, 18:445–490, 2003.

S. Acid, L. M. de Campos, and J. G. Castellano. Learning Bayesian network classifiers: searching in a space of partially directed acyclic graphs. *Machine Learning*, 59:213–235, 2005.

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974.

S. Andersson, D. Madigan, and M. Perlman. A Characterization of Markov equivalence classes for acyclic digraphs. *Annals of Statistics*, 25:505–541, 1997.

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 *Proceedings of the European Conference on Artificial Intelligence in Medicine*, pages 247–256, 1989.

J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29:213–244, 1997.

C. L. Blake and C. J. Merz. UCI Repository of machine learning databases. *http://www.ics.uci.edu/∼mlearn/MLRepository.html*, University of California, Irvine, Dept. of Information and Computer Sciences, 1998.

R. Blanco, I. Inza, and P. Larrañaga. Learning Bayesian networks in the space of structures by estimation of distribution algorithms. *International Journal of Intelligent Systems*, 18:205–220, 2003.

R. R. Bouckaert. Belief networks construction using the minimum description length principle. *Lecture Notes in Computer Science*, 747:41–48, 1993.

R. R. Bouckaert. *Bayesian Belief Networks: from Construction to Inference*. PhD thesis, University of Utrecht, 1995.

W. Buntine. Theory refinement of Bayesian networks. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 52–60, 1991.

J. Cheng, R. Greiner, J. Kelly, D. A. Bell, and W. Liu. Learning Bayesian networks from data: an information-theory based approach. *Artificial Intelligence*, 137:43–90, 2002.

J. Cheng and R. Greiner. Comparing Bayesian network classifiers. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 101–108, 1999.

D. M. Chickering. A transformational characterization of equivalent Bayesian network structures. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 87–98, 1995.

D. M. Chickering. Learning equivalence classes of Bayesian network structures. *Journal of Machine Learning Research*, 2:445–498, 2002.

D. M. Chickering, D. Geiger, and D. Heckerman. Learning Bayesian networks: Search methods and experimental results. In *Preliminary Papers of the Fifth International Workshop on Artificial Intelligence and Statistics*, pages 112–128, 1995.

C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14:462–467, 1968.

G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–348, 1992.

D. Dash and M. Druzdzel. A hybrid anytime algorithm for the construction of causal models from sparse data. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 142–149, 1999.

L. M. de Campos. Independency relationships and learning algorithms for singly connected networks. *Journal of Experimental and Theoretical Artificial Intelligence*, 10:511–549, 1998.

L. M. de Campos, J. M. Fernández-Luna, J. A. Gámez, and J. M. Puerta. Ant colony optimization for learning Bayesian networks. *International Journal of Approximate Reasoning*, 31:291–311, 2002.

L. M. de Campos, J. M. Fernández-Luna, and J. M. Puerta. Local search methods for learning Bayesian networks using a modified neighborhood in the space of dags. *Lecture Notes in Computer Science*, 2527:182–192, 2002.

L. M. de Campos, J. M. Fernández-Luna, and J. M. Puerta. An iterated local search algorithm for learning Bayesian networks with restarts based on conditional independence tests. *International Journal of Intelligent Systems*, 18:221–235, 2003.

L. M. de Campos, J. A. Gámez, and J. M. Puerta. Learning Bayesian networks by ant colony optimization: Searching in two different spaces. *Mathware and Soft Computing*, IX:251–268, 2002.

L. M. de Campos and J. F. Huete. A new approach for learning belief networks using independence criteria. *International Journal of Approximate Reasoning*, 24:11–37, 2000.

L. M. de Campos and J. F. Huete. Stochastic algorithms for searching causal orderings in Bayesian networks. In *Technologies for Constructing Intelligent Systems 2 - Tools*, B. Bouchon-Menieur, J. Gutiérrez-Rios, L. Magdalena, R.R. Yager (Eds.), Physica-Verlag, pages 327–340, 2002.

L. M. de Campos and J. M. Puerta. Stochastic local and distributed search algorithms for learning belief networks. In *Proceedings of the III International Symposium on Adaptive Systems: Evolutionary Computation and Probabilistic Graphical Model*, pages 109–115, 2001.

L. M. de Campos and J. M. Puerta. Stochastic local search algorithms for learning belief networks: Searching in the space of orderings. *Lecture Notes in Artificial Intelligence*, 2143:228–239, 2001.

Elvira Consortium. Elvira: An environment for probabilistic graphical models. In *Proceedings of the First European Workshop on Probabilistic Graphical Models*, pages 222–230, 2002. Available at http://www.leo.ugr.es/∼elvira.

M. Evans, N. Hastings, and B. Peacock. *Statistical Distributions*, Second edition. Wiley, 1993.

K. Ezawa, M. Singh, and S. Norton. Learning goal oriented Bayesian networks for telecommunications risk management. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 139–147, 1996.

U. M. Fayyad and K. B. Irani. Multi-valued interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1022–1027, 1993.

N. Friedman and M. Goldszmidt. Learning Bayesian networks with local structure. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 252–262, 1996.

N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

N. Friedman and D. Koller. Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50:95–126, 2003.

I. J. Good. *The Estimation of Probabilities*. MIT Press, 1965.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

E. Herskovits and G. F. Cooper. Kutató: An entropy-driven system for the construction of probabilistic expert systems from databases. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 54–62, 1990.

I. D. Hill and M. C. Pike. Algorithm 299: Chi-squared integral. *Communications of the ACM*, 10:243–244, 1965.

I. D. Hill and M. C. Pike. Remark on algorithm 299: Chi-squared integral. *ACM Transactions on Mathematical Software*, 11:185–185, 1985.

R. V. Hogg and A. T. Craig. *Introduction to Mathematical Statistics*, 5th Edition. Prentice Hall, New York, 1994.

F. V. Jensen. *An Introduction to Bayesian Networks*. UCL Press, 1996.

M. Kayaalp and G. F. Cooper. A Bayesian network scoring metric that is based on globally uniform parameter priors. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pages 251–258, 2002.

T. Kocka and R. Castelo. Improved learning of Bayesian networks. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 269–276, 2001.

R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1137–1143, 1995.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

R. Kohavi, G. John, R. Long, D. Manley, and K. Pfleger. MLC++: A machine learning library in C++. In *Proceedings of the Sixth International Conference on Tools with Artificial Intelligence*, pages 740–743, 1994.

S. Kullback. *Information Theory and Statistics*. Dover Publication, 1968.

W. Lam and F. Bacchus. Learning Bayesian belief networks. An approach based on the MDL principle. *Computational Intelligence*, 10:269–293, 1994.

P. Larrañaga, M. Poza, Y. Yurramendi, R. Murga, and C. Kuijpers. Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:912–926, 1996.

P. Larrañaga, C. Kuijpers, and R. Murga. Learning Bayesian network structures by searching for the best ordering with genetic algorithms. *IEEE Transactions on System, Man and Cybernetics*, 26:487–493, 1996.

D. Madigan, S. A. Andersson, M. D. Perlman, and C. T. Volinsky. Bayesian model averaging and model selection for Markov equivalence classes of acyclic digraphs. *Communications in Statistics – Theory and Methods*, 25:2493–2520, 1996.

A. W. Marshall and I. Olkin. *Inequalities: Theory of Majorization and Its Applications.* Academic Press, New York, 1979.

C. Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 403–410, 1995.

J. W. Myers, K. B. Laskey, and T. Levitt. Learning Bayesian networks from incomplete data with stochastic search algorithms. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 476–485, 1999.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, 1988.

J. Pearl and T. S. Verma. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 220–227, 1990.

J. Pearl and T. S. Verma. A theory of inferred causation. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, pages 441–452, 1991.

L. K. Rasmussen. Bayesian network for blood typing and parentage verification of cattle. PhD thesis, Research Centre Foulum, Denmark, 1995.

J. Rissanen. Stochastic complexity and modeling. *Annals of Statistics*, 14:1080–1100, 1986.

M. Sahami. Learning limited dependence Bayesian classifiers. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 335–338, 1996.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

M. Singh and M. Valtorta. Construction of Bayesian network structures from data: A brief survey and an efficient algorithm. *International Journal of Approximate Reasoning*, 12:111–131, 1995.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction and Search*. Lecture Notes in Statistics 81, Springer Verlag, New York, 1993.

P. Spirtes and C. Meek. Learning Bayesian networks with discrete variables from data. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 294–299, 1995.

J. Suzuki. A construction of Bayesian networks from databases based on the MDL principle. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, pages 266–273, 1993.

J. Tian. A branch-and-bound algorithm for MDL learning Bayesian networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 580–587, 2000.

T. Verma and J. Pearl. Causal networks: Semantics and expressiveness. In *Uncertainty in Artificial Intelligence, 4*, R.D. Shachter, T.S. Lewitt, L.N. Kanal, J.F. Lemmer (Eds.), North-Holland, Amsterdam, pages 69–76, 1990.

N. Wermuth and S. Lauritzen. Graphical and recursive models for contingence tables. *Biometrika*, 72:537–552, 1983.

M. L. Wong, W. Lam, and K. S. Leung. Using evolutionary computation and minimum description length principle for data mining of probabilistic knowledge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:174–178, 1999.

S. Yang and K. Chang. Comparison of score metrics for Bayesian network learning. *IEEE Transactions on System, Man and Cybernetics–Part A: Systems and Humans*, 32:419–428, 2002.