Probabilistic Network Construction Using the Minimum Description Length Principle

Remco R. Bouckaert

RUU-CS-94-27
July 1994

Utrecht University
Department of Computer Science
Padualaan 14, P.O. Box 80.089,
3508 TB Utrecht, The Netherlands,
Tel. : + 31 - 30 - 531454

Probabilistic Network Construction Using the Minimum Description Length Principle

Remco R. Bouckaert

Technical Report RUU-CS-94-27
July 1994

Department of Computer Science
Utrecht University
P.O.Box 80.089
3508 TB Utrecht
The Netherlands

ISSN: 0924–3275

Probabilistic Network Construction Using the Minimum Description Length Principle

Remco R. Bouckaert

Department of Computer Science, Utrecht University,
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands,
e-mail: remco@cs.ruu.nl

Abstract

Probabilistic networks can be constructed from a database of cases by selecting a network that has highest quality with respect to this database according to a given measure. A new measure is presented for this purpose based on a minimum description length (MDL) approach. This measure is compared with a commonly used measure based on a Bayesian approach both from a theoretical and an experimental point of view. We show that the two measures have the same properties for infinite large databases. For smaller databases, however, the MDL measure assigns equal quality to networks that represent the same set of independencies while the Bayesian measure does not. Preliminary test results suggest that an algorithm for learning probabilistic networks using the minimum description length approach performs comparably to a learning algorithm using the Bayesian approach. However, the former is slightly faster.

1 Introduction

The framework of probabilistic networks, also known as causal networks and Bayesian belief networks, offers a mathematically sound formalism for representing probabilistic information. Efficient algorithms have been designed for making inferences with information represented in a probabilistic network [9, 12, 15]. In various domains the framework has been applied successfully [1, 2, 7] indicating its practical use.

Constructing a probabilistic network for a given domain by hand is a time consuming task: the domain knowledge of one or more experts must be modelled in the formalism of probabilistic networks and once an initial network is constructed it needs to be tested and examinated for improval. This process is repeated until no more improvements can be made. Automated learning of probabilistic networks from a database of cases can help shorten this build and test cycle by suggesting an initial network.

Learning algorithms for probabilistic networks developed so far can be divided into algorithms based on non-Bayesian approaches [4, 17, 22, 23, 24] and algorithms based on a Bayesian approach [5, 10, 14, 21]. The non-Bayesian approaches employ statistical tests on databases for deciding on the existence of arcs in the probabilistic network under construction. The Bayesian approach assumes a prior probability distribution over all possible networks and updates this distribution after observing the database; it then chooses the network with highest updated probability. The Bayesian approach has several advantages over the non-Bayesian approaches. No statistical tests for conditional independence are used thus avoiding the need of huge databases. Also a natural stopping criterion for selection algorithms is provided instead of a more or less arbitrary threshold value. In addition, a collection of most likely networks can be obtained and prior knowledge of the domain at hand can be easily incorporated.

At the heart of the Bayesian approach lies a quality measure, essentially being the probability of the network after observing the database. In this paper, we will present a quality measure based on the minimum description length (MDL) principle; the MDL principle stems from coding theory where the aim is to find an as short as possible description of a database with as few parameters as possible. The MDL measure can be regarded as an approximation of the Bayesian measure, and thus has a Bayesian interpretation. However, it offers several advantages over the Bayesian measure.

In Section 2, we introduce notational conventions, definitions, and assumptions used in the remainder of the paper. In Section 3, we describe the Bayesian approach. In Section 4, we introduce the minimum description length principle and discuss some of its properties; a learning algorithm based on the principle is presented. In Section 5 some preliminary results obtained from experiments are described. The paper is rounded off with conclusions in Section 6.

2 Preliminaries

In this section, we present notational conventions and definitions used in the remainder of this paper. In addition, we list the conditions that we assume to hold for the theory to be presented.

Let U be a set of discrete variables $\{x_1, \ldots, x_n\}$, $n \geq 1$. Each variable $x_i \in U$ can take a value from the set $\{x_{i1}, \ldots, x_{ir_i}\}$, $r_i > 1$, $i = 1, \ldots, n$. We will assume that every variable is an element of U and every set of variables is a subset of U unless stated otherwise. In the sequel, we will use capital letters to denote sets of variables and lower case letters to denote single variables. To prevent an abundant usage of braces, we sometimes write x to denote $\{x\}$, XY to denote $X \cup Y$, and xy to denote $\{x, y\}$.

A *probabilistic network* B over U is a pair $B = (B_S, B_P)$ where the *network structure* B_S is a directed acyclic graph (DAG) with a node for every variable in U; B_P is a set of conditional probability tables associated with B_S. For every variable $x_i \in U$, the set B_P contains a conditional probability table $P(x_i|\pi_i)$ that enumerates the probabilities of all values of x_i given all combinations of values of the variables in its *parent set* π_i in the network structure B_S; in the sequel, such a combination of values will be called an *instantiation*. The network B represents the joint probability distribution represented $P(U)$ defined by $P(U) = \prod_{i=1}^{n} P(x_i|\pi_i)$ [15].

Let P be a joint probability distribution over U. Let X, Y, and Z be sets of variables of U. We say that X and Y are *conditionally independent* given Z in P, written $I(X, Z, Y)$, if $P(XY|Z) = P(X|Z)P(Y|Z)$ for all value assignments of XYZ. A statement $I(X, Z, Y)$ is called an *independency statement*. An *independency model* M is a set of independency statements.

Let B_S be a network structure. From a network structure independency statements can be read using the so-called d-separation criterion. To this end, we introduce the notion of blocked trail. A *trail* in B_S is a path in the underlying (undirected) graph; a *head-to-head node* in a trail in B_S is a node z such that the sequence $x \rightarrow z \leftarrow y$ is part of the trail. We say that a trail in B_S between two nodes x and y is *blocked* by a set of nodes Z if at least one of the following two conditions holds:

- the trail contains a head-to-head node z such that z nor any descendant of z is in Z;
- the trail contains a node x in the trail such that $x \in Z$ and x is not a head-to-head node in the trail.

Let B_S be a network structure. We say that X is *d-separated* from Y by Z if every trail between any node $x \in X$ and any node $y \in Y$ is blocked by Z. With a network structure B_S, we associate an independency model M_S by taking $I(X, Z, Y) \in M_S$ if and only if X and Y are d-separated by Z in B_S.

Now, let M be the set of independency statements that hold in a joint probability distribution P over U. A network structure B_S is an *independency map* or *I-map* of M if X and Y are d-separated by Z in B_S implies $I(X, Z, Y) \in M$; B_S is a *minimal I-map* of M if B_S is an I-map of M and no proper subgraph of B_S is an I-map of M.

For a network structure B_S, two nodes x and y are *adjacent*, written $x \Leftrightarrow y$, if no set $S \subseteq U \backslash xy$ d-separates x and y; otherwise the nodes are *non-adjacent*, written $x \not\Leftrightarrow y$. The nodes x and y are *conditionally adjacent* given a node z, written $x \Leftrightarrow y|z$, if no set $S \subseteq U \backslash xy$ containing z d-separates x and y; otherwise x and y are *conditionally non-adjacent* given z, written $x \not\Leftrightarrow y|z$. A *v-node* in B_S is a triple of nodes x, y, z such that $x \rightarrow y$ and $z \rightarrow y$ are arcs in B_S and $x \not\Leftrightarrow z$ in B_S.

Let $\theta : U \rightarrow \{1, \ldots, n\}$ be a total ordering on U. For two variables x and y, we write $x <_\theta y$ if $\theta(x) < \theta(y)$; as long as ambiguity cannot occur, we write $<$ instead of $<_\theta$. We say that a network structure B_S *obeys* an ordering θ on U if for each arc $x \rightarrow y$ in B_S, the property $x < y$ holds.

On a network structure B_S we define the *arc reversal* operation. This operation applies to two nodes x_j and x_k such that there is no path from x_k to x_j nor from x_j to x_k with the possible exception of the path $x_j \rightarrow x_k$. The network structure B_S is transformed into a new network structure $B_{S'}$ by taking $B_{S'}$ equal to B_S if $x_j \notin \pi_k$. However, if $x_j \in \pi_k$ then $\pi'_i = \pi_i$ for all $i \neq j$ and $i \neq k$ and $\pi'_j = \pi_j \cup \pi_k \cup x_k$ and $\pi'_k = \pi_j \cup \pi_k \backslash x_j$. We write $B_{S'} = arcr(B_S, x_j, x_k)$ to denote that $B_{S'}$ is the network structure obtained by applying an arc-reversal on x_j and x_k in B_S.

A *case* over U is a value assignment to all variables $x_i \in U$. A *database D of cases* over U is a list of cases over. In this paper, we assume that the cases in the database are independent of each other so that the order of the cases in the database is of no importance. Further, we assume that there are no cases with missing values in the database.

We assume that no probability table is preferred for a given structure before the database has been seen, that is, we assume the density function $f(B_P|B_S)$ is uniform.

3 Learning Probabilistic Networks: A Bayesian Approach

Learning a probabilistic network from a database D of cases comprises two tasks: learning the network structure B_S, and, after a proper network structure is identified, learning the set of conditional probability tables B_P. In this paper, we focus on learning the network structure B_S. Once B_S is known, B_P can be estimated directly from the database, [5].

3.1 The Bayesian Measure

The basic idea of the Bayesian approach is to maximize the probability of the network structure given the data, that is, to maximize $P(B_S|D)$ over all possible network structures B_S given the cases of the database D. To this end, the probability given the database is calculated for various network structures and the one with the highest probability is selected. In order to compare the probabilities of two network structures B_{S_1} and B_{S_2} we can calculate

$$\frac{P(B_{S_1}|D)}{P(B_{S_2}|D)} = \frac{\frac{P(B_{S_1},D)}{P(D)}}{\frac{P(B_{S_2},D)}{P(D)}} = \frac{P(B_{S_1},D)}{P(B_{S_2},D)}.$$

Note that as for all network structures $P(D)$ is the same, it suffices to calculate $P(B_S, D)$ for all B_S. To this end, Cooper and Herskovits provide the following formula, [5].

Theorem 3.1 *Let U be the set of variables $\{x_1, \ldots, x_n\}$, $n \geq 1$, where each x_i can take a value from $\{x_{i1}, ..., x_{ir_i}\}$, $r_i \geq 1$, $i = 1, \ldots, n$. Let D be the database of cases over U and let N be the number of cases in D. Let B_S denote a network structure over U, and for each variable x_i, let π_i be the set of parents of x_i in B_S. Furthermore, for each π_i, let w_{ij}*

a	b
0	0
0	0
0	0
0	1
1	0
1	0
1	1
1	1

Table 1: A database of cases over two binary variables.

denote the jth instantiation of π_i relative to D, $j = 1, \ldots, q_i$, $q_i \geq 0$. Now, let N_{ijk} be the number of cases in D in which variable x_i has the value x_{ik} and π_i is instantiated as w_{ij}. let $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. Then,

$$P(B_S, D) = P(B_S) \cdot \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i \Leftrightarrow 1)!}{(N_{ij} + r_i \Leftrightarrow 1)!} \cdot \prod_{k=1}^{r_i} N_{ijk}!. \tag{1}$$

In the right hand side, the term $P(B_S)$ denotes the prior probability of the network structure B_S. In this term, information prior to observation of the database can be incorporated. For example, if an expert suggests the existence of a specific arc or the direction of an arc, network structures that model this suggestion can be given a higher prior probability. If no prior information is available, $P(B_S)$ can be chosen to be uniformly distributed; note that in that case the term can be neglected when two network structures are compared. The other terms in the right hand side of Formula(1) represent how well the network structure fits the database; however, these terms are not very intuitive. Formula (1) can be regarded as a quality measure $Q(B_S, D)$ of a network structure B_S given a database D, and we will refer to it as the *Bayesian measure.*

It would be nice of network structures that represent the same set of independency statements would have the same quality given a database if no prior information is available. Unfortunately, this is not the case for the Bayesian measure. Consider the database in Table 1 and the structure S_1 which is $a \rightarrow b$ and structure S_2 which is $b \rightarrow a$. Both structures represent the same set of independencies (namely none). Yet,

$$P(S_1, D) = P(S_1) \frac{(2 \Leftrightarrow 1)!}{(8 + 2 \Leftrightarrow 1)!} 4!4! \frac{(2 \Leftrightarrow 1)!}{(4 + 2 \Leftrightarrow 1)!} \frac{(2 \Leftrightarrow 1)!}{(4 + 2 \Leftrightarrow 1)!} 3!1!2!2! = P(S_1) \frac{24}{25} \frac{1}{9!}$$

and

$$P(S_2, D) = P(S_2) \frac{(2 \Leftrightarrow 1)!}{(8 + 2 \Leftrightarrow 1)!} 5!3! \frac{(2 \Leftrightarrow 1)!}{(5 + 2 \Leftrightarrow 1)!} \frac{(2 \Leftrightarrow 1)!}{(3 + 2 \Leftrightarrow 1)!} 3!1!2!2! = P(S_2) \frac{1}{9!}.$$

So, $P(B_{S_1}, D) = \frac{24}{25}P(B_{S_2}, D)$ if we assume both structures equiprobable ($P(S_1) = P(S_2)$).

3.2 Heuristic Search Procedure

Based on the Bayesian measure, Cooper and Herskovits have developed an algorithm for learning network structures from data. We observe that the number of different network structures over n nodes is given by the recursive formula $G(0) = 1$, $G(n) = \sum_{i=1}^{n}(\Leftrightarrow 1)^{i+1}\binom{n}{i}2^{i(n-1)}G(n \Leftrightarrow i)$, [20]. For example, for $n = 10$ there are approximately 4.2×10^{18} different network structures. As this number is exponential in the number of nodes, it is not feasible from a computational point of view to regard all network structures. To alleviate the computational burden involved, Cooper and Herskovits assume an ordering on the nodes is given. This leaves only $O(2^{n^2})$ structures to be considered. As, this number is still exponential in the number of nodes, they have developed a greedy heuristic algorithm called K2 (see below) for selecting a network structure that considers at most $O(n^3)$ different structures. In K2, all nodes are considered independent of each other. For each node, a parent set is calculated by starting with the empty parent set and successively adding to the parent set the node that maximally improves $P(B_S, D)$ until no more node can be added such that $P(B_S, D)$ increases.

```
Algorithm K2

Let the variables of $U$ be ordered $x_1, \ldots, x_n$
for $i = 1, \ldots, n$ do $\pi_{i,new} \leftarrow \pi_{i,old} \leftarrow \emptyset$
for $i = 2, \ldots, n$ do
    repeat
        $\pi_{i,old} \leftarrow \pi_{i,new}$
        Let $B_S$ be defined by $\pi_{1,old} \ldots \pi_{n,old}$
        $z \leftarrow \mathrm{argmax}_y\{P(B_{S_y}, D)/P(B_S, D) \mid y \in \{x_1, \ldots, x_{i-1}\} \backslash \pi_{i,old}$, where
                                    $B_{S_y}$ is $B_S$ but with $\pi_i = \pi_{i,old} \cup \{y\}\}$
        if $P(B_{S_z}, D)/P(B_S, D) > 1$ then $\pi_{i,new} \leftarrow \pi_{i,old} \cup \{z\}$
    until $\pi_{i,new} = \pi_{i,old}$ or $|\pi_{i,new}| = i \Leftrightarrow 1$
output $B_S$ defined by $\pi_{1,new} \ldots \pi_{n,new}$
```

A major drawback of K2 is that the ordering that is chosen on the nodes influences the resulting network structure and the quality of this structure to a large extent. So, it is essential to choose a ‘good’ ordering before K2 is started in order to guarantee a good performance. Such an ordering may be provided by an expert, but automated learning is often applied to avoid participation of expensive experts. An alternative is to start with a random ordering, apply K2 with this ordering, and to optimize this ordering. In [3] an algorithm has been presented for optimizing an ordering for this purpose of removing arcs from a given network structure.

4 A Minimum Description Length Approach

Another way to judge the quality of a network structure is by the minimum description length principle [18, 19] which stems from coding theory where the aim is to create a network structure that describes the database as accurately as possible with as few symbols as possible.

4.1 The MDL Measure

The MDL principle results in the following measure.

Definition 4.1 *Let U, B_S, D, N, n, r_i, N_{ijk}, and N_{ij} be as in Theorem 3.1. Let q_i the number of all possible instantiations of the parent set of x_i. Then, the description length $L(B_S, D)$ of the network structure B_S given the database D is defined by*

$$L(B_S, D) = \log P(B_S) \Leftrightarrow N \cdot H(B_S, D) \Leftrightarrow \frac{1}{2} K \cdot \log N, \qquad (2)$$

where $K = \sum_{i=1}^{n} q_i \cdot (r_i \Leftrightarrow 1)$ and $H(B_S, D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \Leftrightarrow \frac{N_{ijk}}{N} \log \frac{N_{ijk}}{N_{ij}}$.

Note that here q_i is defined as the number of all possible instantiations of π_i while for the Bayesian measure q_i is the number of observed instantiations of π_i in the database. Like the Bayesian measure, Formula (2) is a metric for network structures and databases and it will be referred to as the *MDL measure.*

The first term of Formula (2), $\log P(B_S)$, models the prior distribution on network structures just like for the Bayesian measure. Note that this term is no part of the original MDL principle.

The second term of the formula, $N \cdot H(B_S, D)$, represents the conditional entropy of the network structure B_S. Entropy is a non-negative measure of uncertainty which is maximal when uncertainty is maximal and zero when there is complete knowledge; the more information is given the lower the entropy. It will be evident that adding nodes to a parent set will decrease the entropy term in the formula since a probability distribution can be more accurately described.

In the third term, $\frac{1}{2}K \cdot \log N$, the factor K is the number of (independent) probabilities that have to be estimated from the database D for obtaining the probability tables B_P for the network structure B_S. With every probability that is estimated, a small error is introduced. The term $\frac{1}{2}K \cdot \log N$ now represents the error introduced by estimating all required probabilities. This term automatically induces the principle of Occams razor: a

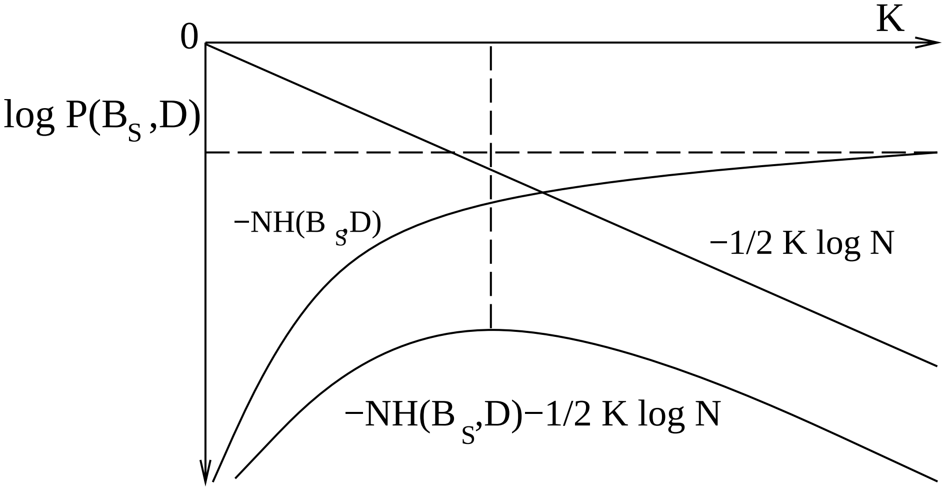

Figure 1: Relation between the various terms of the MDL measure.

network structure with fewer arcs is preferred over a network structure with more arcs unless the conditional entropy of the more complex model is much lower than that of the simpler one.

Figure 1 gives for a given database an impression of the interaction between the entropy term and the third term of formula (2); $P(B_S)$ is assumed uniform for all networks structures. The x-axis represents K which is more or less proportional to the number of arcs in a network structure. The y-axis models the MDL measure and its separate terms. Since we have assumed the prior distribution on network structures to be uniform, $\log P(B_S)$ is constant. With an increasing number of arcs, a network structure will more accurately described the distribution from which the database is obtained; so, the entropy term $\Leftrightarrow N \cdot H(B_S, S)$ increases. On the other hand, the cost term $\frac{1}{2}K \cdot \log N$ decreases when more arcs are added. In sum, the MDL measure will first increase when arcs are added to the network structure and eventually decrease. The network structure with the highest quality will have a balanced contribution of both these terms. Note that due to this property, the MDL principle gives a natural stopping criterion for heuristics that search for network structures.

Approaches based on information criteria as proposed in for example [11, 13], apply a quality measure that is closely related to the MDL measure: the $\log N$ term is replaced by another function and the prior distribution on probabilistic network structures is assumed uniform.

4.2 Comparing the Bayesian and MDL measures

In this section, we compare the MDL measure with the logarithm of the Bayesian measure. The following theorem tells that the MDL measure is approximately equal to the logarithm of the Bayesian measure. So, if the Bayesian measure prefers a network structure B_S over $B_{S'}$, then the MDL measure will prefer B_S over $B_{S'}$ most of the time.

Theorem 4.1 *Let U be a set of variables. Let B_S be a network structure and let D be a database over U with N cases such that all instantiations of the parent sets of B_S occur in the database. Let $P(B_S, D)$ be the Bayesian measure of B_S given D and let $L(B_S, D)$ be the MDL measure of B_S given D. Then,*

$$L(B_S, D) = \log P(B_S, D) + C \tag{3}$$

where C is a constant that does not depend on N.

Proof: Consider once more Formula (1) from Theorem 3.1. By taking the logarithm of this formula, we find,

$$\log P(B_S, D) = \log P(B_S) + \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left\{ \log(r_i \Leftrightarrow 1)! \Leftrightarrow \log(N_{ij} + r_i \Leftrightarrow 1)! + \sum_{k=1}^{r_i} \log N_{ijk}! \right\}. \tag{4}$$

Now, consider the contribution of one variable x_i and one instantiation of its parent set π_i to the expression in the right hand side of this equality:

$$\log(r_i \Leftrightarrow 1)! \Leftrightarrow \log(N_{ij} + r_i \Leftrightarrow 1)! + \sum_{k=1}^{r_i} \log N_{ijk}!.$$

This expression can be written as,

$$\log(r_i \Leftrightarrow 1)! \Leftrightarrow \log(N_{ij} + 1) \dots (N_{ij} + r_i \Leftrightarrow 1) \Leftrightarrow \log N_{ij}! + \sum_{k=1}^{r_i} \log N_{ijk}!. \tag{5}$$

We now apply Stirling's formula $x! \approx \sqrt{2\pi x}(\frac{x}{e})^x$ to the last two terms of expression (5), giving,

$$\Leftrightarrow \log \sqrt{2\pi N_{ij}} \left(\frac{N_{ij}}{e}\right)^{N_{ij}} + \sum_{k=1}^{r_i} \log \sqrt{2\pi N_{ijk}} \left(\frac{N_{ijk}}{e}\right)^{N_{ijk}}. \tag{6}$$

Note that since for larger x, $\sqrt{2\pi x}(\frac{x}{e})^x$ has a relative error of about $\frac{1}{12x}$ ([6] p.112) we introduce an $O(1)$ error. Expression (6) equals,

$$\Leftrightarrow \frac{1}{2} \log 2\pi \Leftrightarrow \left(N_{ij} + \frac{1}{2}\right) \log N_{ij} + N_{ij} \log e + \sum_{k=1}^{r_i} \left\{ \frac{1}{2} \log 2\pi + \left(N_{ijk} + \frac{1}{2}\right) \log N_{ijk} \Leftrightarrow N_{ijk} \log e \right\}.$$

Now note that $\sum_{k=1}^{r_i} N_{ijk} = N_{ij}$ by definition. By exploiting this observation and grouping terms, the $\log e$ terms cancel out, and we find,

$$\sum_{k=1}^{r_i} \frac{1}{2}\log N_{ijk} \Leftrightarrow \frac{1}{2}\log N_{ij} + \sum_{k=1}^{r_i} N_{ijk}\log\frac{N_{ijk}}{N_{ij}} + \frac{r_i \Leftrightarrow 1}{2}\log 2\pi .$$

For N large enough the last term of this expression is negligible; we therefore omit this term, once more introducing an $O(1)$ error. Substitution of the result for the last two terms of expression (5) gives,

$$\log(r_i \Leftrightarrow 1)! \Leftrightarrow \log(N_{ij}+1)\dots(N_{ij}+r_i \Leftrightarrow 1) + \sum_{k=1}^{r_i}\frac{1}{2}\log N_{ijk} \Leftrightarrow \frac{1}{2}\log N_{ij} + \sum_{k=1}^{r_i} N_{ijk}\log\frac{N_{ijk}}{N_{ij}}. \tag{7}$$

The $\log(r_i \Leftrightarrow 1)!$ term is negligible for N large enough and is deleted, again introducing an $O(1)$ error.

Now consider the second term of expression (7). This term $\Leftrightarrow \log(N_{ij}+1)\dots(N_{ij}+r_i \Leftrightarrow 1)$ can be approximated by $\Leftrightarrow \log N_{ij}^{r_i-1}$. By this approximation, an error $\sum_{p=1}^{r_i-1}\log\frac{N_{ij}+p}{N_{ij}}$ is introduced. Using $\log\frac{N_{ij}+p}{N_{ij}} < \log p \leq \log(r_i \Leftrightarrow 1)$, we find $\sum_{p=1}^{r_i-1}\log\frac{N_{ij}+p}{N_{ij}} < (r_i \Leftrightarrow 1)\log(r_i \Leftrightarrow 1)$. As $(r_i \Leftrightarrow 1)log(r_i \Leftrightarrow 1)$ is a constant with respect to N, the approximation of $\Leftrightarrow \log(N_{ij}+1)\dots(N_{ij}+r_i \Leftrightarrow 1)$ by $\Leftrightarrow \log N_{ij}^{r_i-1}$ introduces an $O(1)$ error.

Expression (7), therefore, can be approximated by,

$$\begin{aligned}
& \Leftrightarrow \log N_{ij}^{r_i-1} + \sum_{k=1}^{r_i}\frac{1}{2}\log N_{ijk} \Leftrightarrow \frac{1}{2}\log N_{ij} + \sum_{k=1}^{r_i} N_{ijk}\log\frac{N_{ijk}}{N_{ij}} \\
= \;& \log\frac{\sqrt{\prod_{k=1}^{r_i} N_{ijk}}}{N_{ij}^{r_i-1}\sqrt{N_{ij}}} + \sum_{k=1}^{r_i} N_{ijk}\log\frac{N_{ijk}}{N_{ij}} \\
= \;& \log\frac{\sqrt{\prod_{k=1}^{r_i}\frac{N_{ijk}}{N}}}{\left(\frac{N_{ij}}{N}\right)^{r_i-\frac{1}{2}}}\frac{N^{\frac{1}{2}r_i}}{N^{r_i-\frac{1}{2}}} + \sum_{k=1}^{r_i} N_{ijk}\log\frac{N_{ijk}}{N_{ij}} \\
= \;& \Leftrightarrow \frac{r_i \Leftrightarrow 1}{2}\log N + \sum_{k=1}^{r_i}\frac{1}{2}\log\frac{N_{ijk}}{N} \Leftrightarrow \left(r_i \Leftrightarrow \frac{1}{2}\right)\log\frac{N_{ij}}{N} + \sum_{k=1}^{r_i} N_{ijk}\log\frac{N_{ijk}}{N_{ij}} \\
\approx \;& \Leftrightarrow \frac{r_i \Leftrightarrow 1}{2}\log N + \sum_{k=1}^{r_i} N_{ijk}\log\frac{N_{ijk}}{N_{ij}}.
\end{aligned}$$

The last approximation is allowed since by the strong law of large numbers $\frac{N_{ijk}}{N}$ approximates $P(x_i = x_{ik}, \pi_i = w_{ij})$ and $\frac{N_{ij}}{N}$ approximates $P(\pi_i = w_{ij})$ which both are not functions of N; by this approximation again an $O(1)$ error is introduced.

Summation over j of the above expression gives,

$$\Leftrightarrow \frac{q_i(r_i \Leftrightarrow 1)}{2}\log N + N\sum_{j=1}^{q_i}\sum_{k=1}^{r_i}\frac{N_{ijk}}{N}\log\frac{N_{ijk}}{N_{ij}}.$$

Further, summation over i gives,

$$N\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i}\frac{N_{ijk}}{N}\log\frac{N_{ijk}}{N_{ij}} \Leftrightarrow \sum_{i=1}^{n}\frac{q_i(r_i \Leftrightarrow 1)}{2}\log N. \tag{8}$$

Now recall from the conditions of our theorem that all possible instantiations of the parent sets of B_S occur in the database and therefore, $q_i = \prod_{x_j \in \pi_i} r_j$. Expression (8) therefore equals

$$\Leftrightarrow N \cdot H(B_S, D) \Leftrightarrow \frac{1}{2}K \log N,$$

from which the desired result is obtained. It is easily verified that every approximation made in the course of the derivation has introduced an error of $O(1)$ with respect to N only. □

Note that it will not always behave exactly the same because of the small error C in (3) which make the Bayesian and MDL measure slightly different. Also for databases in which not all possible instantiations of the parent sets in a network occur will result in a different behavior.

Further, from the approximation of $\log N_{ijk}$ and $\log N_{ij}$ by $\log N$, it is easily seen that the MDL measure assigns a larger weight to the cost of estimating parameters (the $\frac{1}{2}K \log N$-term) than the Bayesian measure. As a result, using the MDL measure may yield a network with fewer arcs than using the Bayesian measure.

4.3 Properties of the MDL measure

In this section, we will show that the MDL measure assigns equal quality to network structures that have the same independency model. Before proving this in detail, we state some related properties of the MDL measure.

Consider a network structure B_S. A single arc-reversal operation on a pair of nodes x and y that are not adjacent does not influence the quality of the model. However if x and y are adjacent, then the quality does not change if the parent set of x equals the parent set of y excluding x itself.

Lemma 4.1 *Let U be a set of variables. Let D be a database over U. Let B_S be a network structure over U. Let the prior probability distribution over network structures be uniform. Let x_a and x_b be two nodes in B_S such that $x_a \notin \pi_b$ or $x_a \in \pi_b$ and $\pi_a = \pi_b \backslash x_a$. Let $B_{S'} = arcr(B_S, x_a, x_b)$. Then,*

$$L(B_S, D) = L(B_{S'}, D).$$

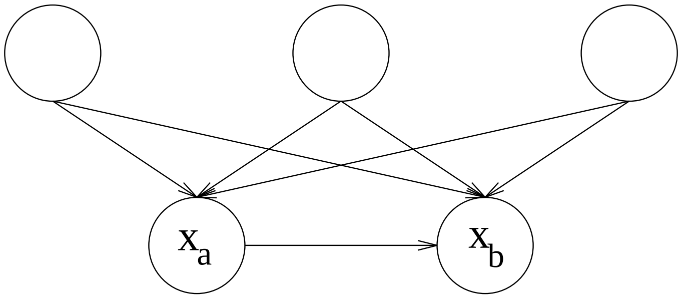

Figure 2: Situation of $x_a \in \pi_b$ and $\pi_a = \pi_b \backslash x_a$.

Proof: In the case that $x_a \notin \pi_b$ we have by definition of arc-reversal that $B_S = B_{S'}$ and the lemma is trivially true. In the remainder of the proof we assume that $x_a \in \pi_b$ and $\pi_a = \pi_b \backslash x_a$ as depicted in Figure 2.

We have to show that $\log P(B_S) \Leftrightarrow N \cdot H(B_S, D) \Leftrightarrow \frac{1}{2}K \cdot \log N = \log P(B_{S'}) \Leftrightarrow N \cdot H(B_{S'}, D) \Leftrightarrow \frac{1}{2}K' \cdot \log N$, where N is the number of cases in D and K and K' are the numbers of independent probabilities to be assessed for B_S and $B_{S'}$ respectively. We will prove the equality by showing that $\log P(B_S) = \log P(B_{S'})$, $N \cdot H(B_S, D) = N \cdot H(B_{S'}, D)$, and $\frac{1}{2}K \cdot \log N = \frac{1}{2}K' \cdot \log N$.

Since we assumed a uniform distribution over all network structures, $\log P(B_S)$ equals $\log P(B_{S'})$. Now consider the entropy terms $N \cdot H(B_S, D)$ and $N \cdot H(B_{S'}, D)$. For B_S, let r_i, q_i, N_{ijk}, and N_{ij} be as defined in Theorem 3.1, and for $B_{S'}$ let r'_i, q'_i, N'_{ijk}, and N'_{ij} be likewise. Note that $r_i = r'_i$ for all $i = 1, \ldots, n$. By definition, we have,

$$N \cdot H(B_S, D) = \Leftrightarrow \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}}{N_{ij}}.$$

Since $\pi_i = \pi'_i$ in case i is not a or b, we have by definition of arc-reversal that the entropy term equals

$$\Leftrightarrow \sum_{i=1, i \neq a, i \neq b}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N'_{ijk} \log \frac{N'_{ijk}}{N'_{ij}} \Leftrightarrow \sum_{j=1}^{q_a} \sum_{k=1}^{r_a} N_{ajk} \log \frac{N_{ajk}}{N_{aj}} \Leftrightarrow \sum_{j=1}^{q_b} \sum_{k=1}^{r_b} N_{bjk} \log \frac{N_{bjk}}{N_{bj}}. \tag{9}$$

Consider the last two terms of Expression (9). These terms equal,

$$\sum_{j=1}^{q_a} \sum_{k=1}^{r_a} N_{ajk} \log N_{aj} \Leftrightarrow \sum_{j=1}^{q_a} \sum_{k=1}^{r_a} N_{ajk} \log N_{ajk} + \sum_{j=1}^{q_b} \sum_{k=1}^{r_b} N_{bjk} \log N_{bj} \Leftrightarrow \sum_{j=1}^{q_b} \sum_{k=1}^{r_b} N_{bjk} \log N_{bjk}. \tag{10}$$

Now consider the third term of expression (10). Using $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$, we find that $\sum_{j=1}^{q_b} \sum_{k=1}^{r_b} N_{bjk} \log N_{bj} = \sum_{j=1}^{q_b} N_{bj} \log N_{bj}$. From $\pi_b = \pi_a \cup x_a$, we have that for every N_{bj} there is an index j' such that $N_{aj'k} = N_{bj}$. So, $\sum_{j=1}^{q_b} N_{bj} \log N_{bj}$ can be written as $\sum_{j=1}^{q_a} \sum_{k=1}^{r_a} N_{ajk} \log N_{ajk}$. Substitution in (10) gives,

$$\sum_{j=1}^{q_a} \sum_{k=1}^{r_a} N_{ajk} \log N_{aj} \Leftrightarrow \sum_{j=1}^{q_a} \sum_{k=1}^{r_a} N_{ajk} \log N_{ajk} + \sum_{j=1}^{q_a} \sum_{k=1}^{r_a} N_{ajk} \log N_{ajk} \Leftrightarrow \sum_{j=1}^{q_b} \sum_{k=1}^{r_b} N_{bjk} \log N_{bjk}.$$

Note that the middle two terms cancel out, yielding

$$\sum_{j=1}^{q_a}\sum_{k=1}^{r_a} N_{ajk}\log N_{aj} \Leftrightarrow \sum_{j=1}^{q_b}\sum_{k=1}^{r_b} N_{bjk}\log N_{bjk}. \tag{11}$$

For the first term of expression (11) we once more have $\sum_{j=1}^{q_a}\sum_{k=1}^{r_a} N_{ajk}\log N_{aj}$ equals $\sum_{j=1}^{q_a} N_{aj}\log N_{aj}$. Now observe that from $\pi'_b = \pi_a$ we have that $q'_b = q_a$ and $N'_{bj} = N_{aj}$. So, we can write the sum $\sum_{j=1}^{q_a} N_{aj}\log N_{aj}$ as $\sum_{j=1}^{q'_b} N'_{bj}\log N'_{bj}$, which is equal to $\sum_{j=1}^{q'_b}\sum_{k=1}^{r'_b} N'_{bjk}\log N'_{bj}$.

Now, we consider the second term $\sum_{j=1}^{q_b}\sum_{k=1}^{r_b} N_{bjk}\log N_{bjk}$. N_{bjk} is the number of cases in which π_b takes value w_{bj} and x_b takes value x_{bk}. Likewise, $N'_{aj'k'}$ is the number of cases in which π'_a takes value $w'_{aj'}$ and x_a takes value $x'_{bk'}$. Since N_{bjk} and $N'_{aj'k'}$ involve the same set of variables, for each N_{bjk} there are indexes j' and k' such that N_{bjk} is equal to $N'_{aj'k'}$. So, $\sum_{j=1}^{q_b}\sum_{k=1}^{r_b} N_{bjk}\log N_{bjk}$ can be written as $\sum_{j=1}^{q'_a}\sum_{k=1}^{r'_a} N'_{ajk}\log N'_{ajk}$ thus obtaining that (11) equals,

$$\sum_{j=1}^{q'_b}\sum_{k=1}^{r'_b} N'_{bjk}\log N'_{bj} \Leftrightarrow \sum_{j=1}^{q'_a}\sum_{k=1}^{r'_a} N'_{ajk}\log N'_{ajk}. \tag{12}$$

We have,

$$0 = \sum_{j=1}^{q'_a}\sum_{k=1}^{r'_a} N'_{ajk}\log N'_{aj} \Leftrightarrow \sum_{j=1}^{q'_a}\sum_{k=1}^{r'_a} N'_{ajk}\log N'_{aj} = \sum_{j=1}^{q'_a}\sum_{k=1}^{r'_a} N'_{ajk}\log N'_{aj} \Leftrightarrow \sum_{j=1}^{q'_b}\sum_{k=1}^{r'_b} N'_{bjk}\log N'_{bjk},$$

in which the last equation follows from a same line of reasoning as above but now realizing that $\pi'_a = \pi'_b \cup x_b$. Adding this term to (12) gives,

$$\sum_{j=1}^{q'_b}\sum_{k=1}^{r'_b} N'_{bjk}\log N'_{bj} \Leftrightarrow \sum_{j=1}^{q'_b}\sum_{k=1}^{r'_b} N'_{bjk}\log N'_{bjk} + \sum_{j=1}^{q'_a}\sum_{k=1}^{r'_a} N'_{ajk}\log N'_{aj} \Leftrightarrow \sum_{j=1}^{q'_a}\sum_{k=1}^{r'_a} N'_{ajk}\log N'_{ajk}.$$

Using $\log x \Leftrightarrow \log y = \log\frac{x}{y}$ we get,

$$\Leftrightarrow \sum_{j=1}^{q'_b}\sum_{k=1}^{r'_b} N'_{bjk}\log\frac{N'_{bjk}}{N'_{bj}} \Leftrightarrow \sum_{j=1}^{q'_a}\sum_{k=1}^{r'_a} N'_{ajk}\log\frac{N'_{ajk}}{N'_{aj}}.$$

Substituting this result in (9) gives,

$$\Leftrightarrow \sum_{i=1,i\neq a,j\neq b}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N'_{ijk}\log\frac{N'_{ijk}}{N'_{ij}} \Leftrightarrow \sum_{j=1}^{q'_a}\sum_{k=1}^{r_a} N'_{ajk}\log\frac{N'_{ajk}}{N'_{aj}} \Leftrightarrow \sum_{j=1}^{q'_b}\sum_{k=1}^{r_b} N'_{bjk}\log\frac{N'_{bjk}}{N'_{bj}},$$

and by reordering terms, this is,

$$\Leftrightarrow \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N'_{ijk}\log\frac{N'_{ijk}}{N'_{ij}},$$

which by definition is $N \cdot H(B_{S'}, D)$.

To complete the prove, it remains to be shown that $\frac{1}{2}K \cdot \log N = \frac{1}{2}K' \cdot \log N$. Note that it is sufficient to show that K equals K'. By definition, we have,

$$K \Leftrightarrow K' = \sum_{i=1}^{n} \{q_i(r_i \Leftrightarrow 1) \Leftrightarrow q'_i(r'_i \Leftrightarrow 1)\}.$$

We recall that $r'_i = r_i$ for all $i = 1, \ldots, n$. In addition, we have for $i \in \{1, \ldots, n\} \backslash \{a, b\}$, that $q'_i = q_i$. From these observations, we have,

$$K \Leftrightarrow K' = q_a(r_a \Leftrightarrow 1) + q_b(r_b \Leftrightarrow 1) \Leftrightarrow q'_a(r_a \Leftrightarrow 1) \Leftrightarrow q'_b(r_b \Leftrightarrow 1).$$

By definition, we have, $q_a = \prod_{x \in \pi_a} r_x$ and $q_b = \prod_{x \in \pi_b} r_x$; a similar observation holds for q'_a and q'_b. Substitution gives,

$$K \Leftrightarrow K' = \prod_{x \in \pi_a} r_x(r_a \Leftrightarrow 1) + \prod_{x \in \pi_b} r_x(r_b \Leftrightarrow 1) \Leftrightarrow \prod_{x \in \pi'_a} r_x(r_a \Leftrightarrow 1) \Leftrightarrow \prod_{x \in \pi'_b} r_x(r_b \Leftrightarrow 1).$$

$$= (\prod_{x \in \pi_a} r_x \Leftrightarrow \prod_{x \in \pi'_a} r_x)(r_a \Leftrightarrow 1) + (\prod_{x \in \pi_b} r_x \Leftrightarrow \prod_{x \in \pi'_b} r_x)(r_b \Leftrightarrow 1).$$

Since $\pi'_a = \pi_a \cup \{b\}$ and $\pi'_b = \pi_b \backslash \{a\} = \pi_a$, we find

$$K \Leftrightarrow K' = \prod_{x \in \pi_a} r_x(1 \Leftrightarrow r_b)(r_a \Leftrightarrow 1) + \prod_{x \in \pi_a} r_x(r_a \Leftrightarrow 1)(r_b \Leftrightarrow 1) = 0.$$

So, $K = K'$. We have shown now that all three terms of the MDL measure remain the same after applying a single arc-reversal operation under the conditions stated in the lemma, from which the propery to be proved follows immedeately. □

From the previous lemma we have that in some cases arc reversal does not influence the quality of the network structure given the data. The condition under which this property holds implies that in these cases arc reversal does not change the independency model represented by the structure. The following lemma now states that any two network structures that represent the same independency model can be transformed into each other by applying successively reversal operations under the conditions mentioned above.

Lemma 4.2 *Let U be a set of variables. Let B_S and $B_{S'}$ be network structures over U such that for their independency models M_S and $M_{S'}$ we have $M_S = M_{S'}$. Then, a finite sequence $B_1, \ldots, B_k$, $k \geq 1$, of network structures over U exists such that $B_S = B_1$, $B_{S'} = B_k$, and for $1 \leq i < k$, $B_{i+1} = arcr(B_i, x_{a_i}, x_{b_i})$ for nodes x_{a_i} and x_{b_i} in B_i with $x_{a_i} \notin \pi_{b_i}$ or $x_{a_i} \in \pi_{b_i}$ and $\pi_{a_i} = \pi_{b_i} \backslash x_{a_i}$.*

Proof: For two network structures B_S and $B_{S'}$, we have $M_S = M_{S'}$ if and only if $x_a \Leftrightarrow\!\!\!\!\!\Leftrightarrow x_b \in B_S \Leftrightarrow x_a \Leftrightarrow\!\!\!\!\!\Leftrightarrow x_b \in B_{S'}$, and x, y, z vorms a v-node in B_S if and only if x, y, z vorms a v-node in $B_{S'}$, [16].

Note that the condition $x_a \Leftrightarrow\!\!\!\!\!\Leftrightarrow x_b \in B_S \Leftrightarrow x_a \Leftrightarrow\!\!\!\!\!\Leftrightarrow x_b \in B_{S'}$ implies that both B_S and $B_{S'}$ have the same underlying undirected graph, however, the direction of the arcs may not be the same in both graphs.

We proof the lemma with induction to the number of reversed arcs in B_S with respect to $B_{S'}$, that is, the number of pairs of nodes x_i, x_j such that $x_i \to x_j$ is an arc in B_S and $x_j \to x_i$ is an arc in $B_{S'}$.

If there are zero reversed arcs then B_S is equal to $B_{S'}$ and the lemma is trivially true for $k = 1$.

Assume that the lemma holds if there are $d \geq 0$ reversed arcs. Let $B_{S'}$ be a network stucture that contains $d+1$ reversed arcs with respect to B_S. We show that an arc reversal on $B_1 = B_S$ can be performed such that the $B_{S'}$ contains d reversed arcs with respect to the obtained network stucture B_2.

By inspection of the definition of arc reversal we find that performing and arc-reversal on two nodes x_a and x_b for which the conditions of the lemma apply (the conditions for arc reversal hold, $x_a \notin \pi_b$ or $x_a \in \pi_b$ and $\pi_a = \pi_b \backslash x_a$) will not introduce new adjacencies nor remove old ones. In addition no new v-nodes will appear. We conclude that the independency model of B_2 is the same as the one of B_1 if we can find two such nodes x_a and x_b.

Let θ be an ordering obeyed by B_1. Let x_a and x_b be two nodes in B_i on which an arc reversal can be performed and let $x_b \to x_a$ be an arc in $B_{S'}$. Furthermore, let x_b be the lowest ordered node according to θ for which this condition hold.

Suppose that $x_a \in \pi_b$ and not $\pi_a = \pi_b \backslash x_a$. We distinguish two cases for π_a and π_b:

- $\pi_b \backslash (\pi_a \cup x_a) \neq \emptyset$. Let x be a node in $\pi_b \backslash (\pi_a \cup x_a)$. Then in B_1, we have $x_a \not\Leftrightarrow\!\!\!\!\!\Leftrightarrow x$ on the one hand and on the other hand. However, if $x_a \to x_b$ would be reversed in $B_{S'}$ x_a, x_b, x cannot be vorming a v-node. So, $\pi_b \backslash (\pi_a \cup x_a)$ must be empty.

- $(\pi_a \cup x_a) \backslash \pi_b \neq \emptyset$. Let x be a node in $(\pi_a \cup x_a) \backslash \pi_b$. Then in B_1, we have $x_a \Leftrightarrow\!\!\!\!\!\Leftrightarrow x$, $x_a \Leftrightarrow\!\!\!\!\!\Leftrightarrow x_b$ and $x \not\Leftrightarrow\!\!\!\!\!\Leftrightarrow x_b$. Furthermore, $<x_b, \pi_b, x>$ holds in B_1, thus also in B_S and $B_{S'}$. In B_2 we must have these properties also. However, if the arrow $x_a \to x_b$ is just flipped in direction, we would obtain $x \to x_a \leftarrow x_b$ and $<x_b, \pi_b, x>$ would not hold. For the d-separation statement to hold in $B_{S'}$, $x \to x_a$ will have to be flipped in direction in $B_{S'}$ also. But, than a pair x'_a, x'_b would have to exist with $x'_b <_\theta x_b$ which must be false by our choise of x_b.

So, our assumption that $x_a \in \pi_b$ and not $\pi_a = \pi_b \backslash x_a$ has to be false and thus $x_a \notin \pi_b$ or $\pi_a = \pi_b \backslash x_a$. Application of an arc reversal under these conditions does not introduce new adjacencies nor new head-to-head nodes.

So, in B_1, $1 \leq i < k$, always two nodes x_a and x_b can be found such that an arc-reversal can be performed that does not change the represented independendcy model. Furthermore, in B_2 there are only d reversed arcs with respect to $B_{S'}$ so by the indution hypothesis there is a finite sequence $B_2, \ldots, B_{S'}$ such that $B_{i+1} = arcr(B_i, x_{a_i}, x_{b_i})$ under the conditions stated in the Lemma. Therefore, there is a finite sequence $B_S, \ldots, B_{S'}$ such that $B_{i+1} = arcr(B_i, x_{a_i}, x_{b_i})$ under the conditions stated in the Lemma. □

The lemma says that when we have two network stucture s that represent the same independency model then always a sequence of arcreversals exist that leaves the represented independency model unaltered and that transform the first network stucture into the other.

Theorem 4.2 *Let U be a set of variables. Let D be a database over U and B_S be a network structure over U. Let the prior probability distribution on network structures be uniform. Then, for every network structure $B_{S'}$ that represents the same independency model as B_S we have,*

$$L(B_S, D) = L(B_{S'}, D).$$

Proof: From Lemma 4.2, we have that a finite sequence of network structures $B_1, \ldots, B_k$, $k \geq 1$, exists such that $B_S = B_1$, $B_{S'} = B_k$ and $B_{i+1} = arcr(B_i, x_{a_i}, x_{b_i})$ with $x_{a_i} \notin \pi_{b_i}$ or $x_{a_i} \in \pi_{b_i}$ and $\pi_{a_i} = \pi_{b_i} \backslash x_{a_i}$ in B_i. From Lemma 4.1, we have that such arc-reversals do not change the MDL measure for the resulting network structures. So, $L(B_{i+1}, D) = L(B_i, D)$ for $i = 1, \ldots, k$ hence $L(B_S, D) = L(B_{S'}, D)$. □

From this theorem we have that all network structures that represent the same set of independencies have the same quality according to the MDL measure. Recall from Section 3.1 that a similar property does not hold in general for the Bayesian measure. Yet, the MDL principle for probabilistic networks inherits all advantages of the Bayesian approach. For example, no statistical tests for conditional independence are used thus avoiding the need of huge databases. Also a natural stopping criterion for selection algorithms is provided instead of a more or less arbitrary threshold value. In addition, a collection of most likely networks can be obtained and prior knowledge of the domain at hand can be easily incorporated.

4.4 Asymptotic Behavior of the MDL Measure

For the Bayesian measure it is known that it prefers minimal I-maps over other network structures for large databases [10]. In this section, we investigate the behavior of the MDL measure for large databases.

Theorem 4.3 *Let U be a set of variables and let θ be a total ordering on U. Let P_D be a distribution over U with a unique minimal I-map obeys θ. Let D be a database D generated from P_D. Let B_S be a minimal I-map of P_D and let $B_{S'}$ be a network structure such that both B_S and $B_{S'}$ that obey θ. Let the prior probability distribution over all network structures be uniform. Then,*

$$\lim_{N\to\infty}\left(L(B_{S'},D)\Leftrightarrow L(B_S,D)\right)\ll 0$$

if and only if $B_{S'}\neq B_S$.

Proof: Let N, n, r_i, x_i, x_{ik}, w_{ij}, N_{ijk}, and N_{ij} be as in Theorem 3.1 for network structure B_S and let r'_i, x'_i, x'_{ik}, w'_{ij}, N'_{ijk}, and N'_{ij} be likewise for $B_{S'}$. Let K and K' be the numbers of probabilities to be assessed and q_i and q'_i the number of instantiations that the parents of x_i can be assigned to for B_S and $B_{S'}$, respectively. We consider $\lim_{N\to\infty}\left(L(B_{S'},D)\Leftrightarrow L(B_S,D)\right)$ which by definition is equal to

$$\lim_{N\to\infty}\left(\log P(B_{S'})\Leftrightarrow N\cdot H(B_{S'},D)\Leftrightarrow\frac{1}{2}K'\cdot\log N\Leftrightarrow\log P(B_S)+N\cdot H(B_S,D)+\frac{1}{2}K\cdot\log N\right).$$

Since we assumed a uniform distribution on network structures, the terms $\log P(B_{S'})$ and $\log P(B_S)$ cancel out. So, we consider the expression,

$$\lim_{N\to\infty}\left(\Leftrightarrow N\cdot H(B_{S'},D)+N\cdot H(B_S,D)\Leftrightarrow\frac{1}{2}(K'\Leftrightarrow K)\cdot\log N\right). \qquad (13)$$

Now, consider the behavior in the limit of the entropy term $\Leftrightarrow N\cdot H(B_{S'},D)$ which by definition is $\Leftrightarrow N\cdot\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i}\Leftrightarrow\frac{N_{ijk}}{N}\log\frac{N_{ijk}}{N_{ij}}$. By the strong law of large numbers, we have, $\lim_{N\to\infty}\frac{N_{ijk}}{N}=P(x_i=x_{ik},\pi_i=w_{ij})$ and $\lim_{N\to\infty}\frac{N_{ijk}}{N_{ij}}=P(x_i=x_{ik}|\pi_i=w_{ij})$. Therefore,

$$NH(B_S,D)=N\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i}\Leftrightarrow P(x_i=x_{ik},\pi_i=w_{ij})\log P(x_i=x_{ik}|\pi_i=w_{ij}).$$

A similar property holds for the behavior of the entropy term $N\cdot H(B_{S'},D)$ in the limit of Expression (13). To examine the behavior of $\Leftrightarrow N\cdot H(B_S,D)+N\cdot H(B_{S'},D)$ in the limit, we distinguish between two cases: $B_{S'}$ is not an I-map of P_D, and $B_{S'}$ is an I-map of P_D but not a minimal one. First, suppose that $B_{S'}$ is not an I-map of P_D. Then, in the limit $\Leftrightarrow N\cdot H(B_{S'},D)+N\cdot H(B_S,D)$ is equal to,

$$N\cdot\sum_{i=1}^{n}\left\{\sum_{j=1}^{q'_i}\sum_{k=1}^{r'_i}P(x_i=x_{ik},\pi'_i=w'_{ij})\log P(x_i=x_{ik}|\pi'_i=w'_{ij})\right.$$
$$\left.\Leftrightarrow\sum_{j=1}^{q_i}\sum_{k=1}^{r_i}P(x_i=x_{ik},\pi_i=w_{ij})\log P(x_i=x_{ik}|\pi_i=w_{ij})\right\}.$$

Note that $r'_i = r_i$ for $i = 1, \ldots, n$. Now observe that since $B_{S'}$ is not an I-map of P_D, there is an index i such that $\pi_i \not\subseteq \pi'_i$; if for all i $\pi_i \subseteq \pi'_i$ then $B_{S'}$ would represent less independencies as B_S which is an I-map and thus $B_{S'}$ would be an I-map too. For this index i, let $\pi''_i = \pi_i \cup \pi'_i$, let w''_{ij} be the jth instantiation of π''_i, and let q''_i be the number of all possible instantiations of π''_i. Let $w''_{ij} \cong \pi'_i$ denote the instantiation of the variables in π'_i such that they take the values when $\pi''_i = w''_{ij}$. For $j = 1, \ldots, q''_i$, we have that $P(x_i = x_{ik}|\pi''_i = w''_{ij}) = P(x_i = x_{ik}|\pi_i = w_{ij})$ for each w''_{ij} such that $w_{ij} = (w''_{ij} \cong \pi_i)$, because $I(x_i, \pi_i, \pi''_i)$. So, the above equation can be written as,

$$\begin{aligned} N \cdot \sum_{i=1}^{n} \Bigg\{ & \sum_{j=1}^{q''_i} \sum_{k=1}^{r_i} P(x_i = x_{ik}, \pi''_i = w''_{ij}) \log P(x_i = x_{ik}|\pi'_i = (w''_{ij} \cong \pi'_i)) \\ & \Leftrightarrow \sum_{j=1}^{q''_i} \sum_{k=1}^{r_i} P(x_i = x_{ik}, \pi''_i = w''_{ij}) \log P(x_i = x_{ik}|\pi''_i = w''_{ij}) \Bigg\}. \end{aligned}$$

We now use Shannon's inequality which states $\sum_i \Leftrightarrow a_i \log a_i \leq \sum_i \Leftrightarrow a_i \log b_i$ for all $a_i, b_i \geq 0$ such that $\sum_i a_i = \sum_i b_i = 1$. Using this inequality, the term within brackets must be greater than or equal to 0 because there are instantiations of π''_i such that $P(x_i = x_{ik}|\pi'_i = w''_{ij} \to \pi'_i)$ is not equal to $P(x_i = x_{ik}|\pi''_i = w''_{ij})$. So, the entropy of $B_{S'}$ will be higher than the entropy of B_S. Since, $O(N)$ dominates $O(\log N)$ when $N \to \infty$ the $K \cdot \log N$ and $K' \cdot \log N$ terms vanish in (13) and $\Leftrightarrow N \cdot H(B_{S'}, D) + N \cdot H(B_S, D) \to \Leftrightarrow \infty$.

If $B_{S'}$ is a non minimal I-map then the entropies will be the same. However, at least one extra arc has to be added in $B_{S'}$ and therefore, $K' \Leftrightarrow K > 0$. So, $\Leftrightarrow \frac{1}{2}(K' \Leftrightarrow K) \log N \to \Leftrightarrow \infty$.
□

From the fact that positive distributions have unique minimal I-maps for network structures that obey a given ordering we have the property stated in the following corollary.

Corollary 4.1 *Let U be a set of variables and let θ be a total ordering on U. Let P_D be a positive distribution over U. Let D be a database D generated from P_D. Let B_S be a minimal I-map of P_D and let $B_{S'}$ be a network structure such that both B_S and $B_{S'}$ obey θ. Let the prior probability distribution over all network structures be uniform. Then,*

$$\lim_{N \to \infty} (L(B_{S'}, D) \Leftrightarrow L(B_S, D)) \ll 0$$

if and only if $B_{S'} \neq B_S$.

So, the measure proposed will prefer the original network structure overwhelmingly over other network structures when the number of observations grows to infinity in most cases.

4.5 Heuristic Search Procedure

The MDL measure can be used in algorithms for learning networks structures from data. It will be evident that for the MDL measure the same considerations hold as posed in Section 3 for the Bayesian measure. Therefore, we again assume that an ordering on the variables is given and develop a greedy heuristic algorithm. Our algorithm called K3 is a modification of K2 where the Bayesian measure is replaced by the MDL measure; a uniform prior distribution over network structures is assumed.

```
**Algorithm K3**

Let the variables of $U$ be ordered $x_1, \ldots, x_n$
for $i = 1, \ldots, n$ do $\pi_{i,new} \leftarrow \pi_{i,old} \leftarrow \emptyset$
for $i = 2, \ldots, n$ do
    repeat
        $\pi_{i,old} \leftarrow \pi_{i,new}$
        Let $B_S$ be defined by $\pi_{1,old} \ldots \pi_{n,old}$
        $z \leftarrow \text{argmax}_y\{L(B_{S_y}, D) \Leftrightarrow L(B_S, D) \mid y \in \{x_1, \ldots, x_{i-1}\} \backslash \pi_{i,old}$, where
                                        $B_{S_y}$ is $B_S$ but with $\pi_i = \pi_{i,old} \cup \{y\}\}$
        if $L(B_{S_z}, D) \Leftrightarrow L(B_S, D) > 0$ then $\pi_{i,new} \leftarrow \pi_{i,old} \cup \{z\}$
    until $\pi_{i,new} = \pi_{i,old}$ or $|\pi_{i,new}| = i \Leftrightarrow 1$
output $B_S$ defined by $\pi_{1,new} \ldots \pi_{n,new}$
```

Note that the quality of a network structure B_S compared to a structure B_{S_y} with one more arc can be calculated efficiently, since the terms $L(B_{S_y}, D)$ and $L(B_S, D)$ have many terms in common.

5 Preliminary Test Results

To compare the performance of the heuristic algorithms K2 and K3, we performed some experiments. In these experiments, we proceeded as follows. First, an acyclic network structure B_S with ten binary variables was generated randomly. To this end the variables $x_1, \ldots, x_{10}$ were ordered. Then variables x_i and x_j, $i \neq j$, were chosen randomly, and if $i > j$ an arc $x_i \rightarrow x_j$ was added to the network under construction and $x_j \rightarrow x_i$ otherwise. For the next arc to be added, we randomly selected one node x_i among the nodes that are already incident on an arc and one node x_j among the nodes that are not yet incident on an arc; an arc is placed from the lower ordered node to the higher ordered one. The last step was repeated until nine arcs were generated and a connected graph was yielded. Note that with this method, not every connected network structure has an equal

	K2 vs. Original			K3 vs. Original			K3 vs. K2		
obs.	extra	missing	sum	extra	missing	sum	extra	missing	sum
100	2.2	3.3	5.5	0.8	4.7	5.5	0.0	2.8	2.8
200	1.6	2.3	4.9	0.1	2.6	2.7	0.0	1.8	1.8
300	1.4	1.8	3.2	0.3	3.5	3.8	0.0	2.8	2.8
400	1.4	1.9	3.3	0.2	2.7	2.9	0.0	2.0	2.0
500	1.8	1.5	3.3	0.1	2.1	2.3	0.0	2.3	2.3

Table 2: Test results.

probability of being generated: network structures with long chains and low node degree are less often generated than network structures with a high degree for a couple of nodes and low degree for others. For the generated network structure B_S, conditional probability tables were generated randomly; for each $1 \leq i \leq 10$, $1 \leq j \leq q_i$, $P(x_i = 0|\pi_i = w_{ij})$ was assigned a random number in the unit interval and $P(x_i = 1|\pi_i = w_{ij})$ calculated by $1 \Leftrightarrow P(x_i = 0|\pi_i = w_{ij})$.

With the resulting probabilistic network, a set of cases was generated using logic sampling [8] to constitute a database D. Both K2 and K3 were applied to this database, with the node ordering used for generating the network structure. This procedure was repeated for various database sizes.

The performance of the algorithms was measured in terms of the number of extra arcs and missing arcs in the network structures generated by the algorithms compared with the original network structure. These numbers indicate how closely the structure of the original network is recovered. In Table 2 the average results over ten databases generated from ten different network structures are presented. The first column represents the size of the databases used. The columns labeled 'extra' indicate the number of arcs that can be found in the first network but not in the other; the columns labeled 'missing' indicate the number of arcs found in the second network but not in the first one. The columns labeled 'sum' specify the total number of wrongly placed arcs; this number may be interpreted as the total error made by the learning algorithms.

The first three columns of the table show the results for the performance of K2; the next three columns show the results for K3 and in the last three columns a comparison of K2 and K3 is given. We observe that the number of mismatched arcs tends to decrease as the number of cases increases for both K2 and K3. This tendency is expected, since larger database contain more information than smaller ones.

The last three columns of the table indicate that K3 shows a tendency to stop adding arcs earlier than K2 does; this is seen from the zeros in the second but last column. This tendency confirms our earlier theoretical observations based on Theorem 4.1. Since K3 adds less arcs than K2, K3 performs less computations than K2 and, therefore, K3 has a slightly shorter run-time than K2. Yet, the table suggests that in sum K3 performs comparable to K2. However, it yields structures with more missing arcs than K2 does. As a consequence, it may be possible that K2 outputs an I-map while K3 does not. We would like to emphasize that it highly depends on the purpose for which a learning algorithm is used, which algorithm is to be preferred over the other one; for example, if the learning algorithm is used to start a build-test cycle where an expert is confronted with a network generated by K2 or K3, an abundance of erroneous arcs may be as disturbing as the omission of arcs. It is not quite clear why the sum of mismatched arcs is less on average for K3 compared to K2; further experiments on a wider range of network structures need to be performed to confirm this behavior.

If the network structure in itself is of less interest than the distribution to be learned, another measure analyzing the performance is needed. An example of such a measure is the divergence $\sum_U P(U) \log P(U)/\hat{P}(U)$ where $P(U)$ is the distribution represented in the original network and $\hat{P}(U)$ is the distribution in the learned network. Further experiments are necessary to make final conclusions on this issue.

6 Conclusions

Probabilistic networks can be constructed from a database of cases by selecting a network that has highest quality with respect to a database according to a given measure. In this paper, we have presented a new measure for the quality of a network structure given a database based on the minimum description length (MDL) principle. We have shown that this measure can be regarded an approximation of the logarithm of the Bayesian measure presented by Cooper and Herskovits in many cases. As a consequence, the MDL measure inherits all desirable properties of the Bayesian measure. In addition, it assigns the same quality to all network structures that represent the same set of independencies. Based on the MDL measure, we have presented a new heuristic algorithm, called K3, for learning network structures. We have compared this algorithm with the K2 algorithm based on the Bayesian measure. Preliminary test results suggest that these algorithms perform comparable. The algorithm based on the MDL approach, however, tends to be slightly faster, outputting network structures with fewer arcs than K2. To reach decisive conclusions, however, further experiments are necessary for a wider range of network structures.

A major drawback of both K2 and K3 for learning network structures is that their performance is highly dependent on the ordering on the variables taken as point of departure. To circumvent this drawback, new heuristics need to be developed that do not need this extra information. One approach may be to start with a random ordering, apply K2 or K3, and optimize the ordering outputted network stucture by applying simple operations, such as arc-reversals. The presented MDL measure may be more suitable to this approach since in many cases the quality measure of the network need not be recalculated after the performance of an arc-reversal.

Acknowledgement

I thank Linda van der Gaag for her many helpful remarks that improved the presentation of this paper considerably.

References

[1] S. Andreassen, M. Wolbye, B. Falck, and S.K. Andersen. MUNIN - a causal probabilistic network for interpretation of electromyographic findings. In *Proceedings of the IJCAI*, pages 366–372, 1987.

[2] I. Beinlich, H. Seurmondt, R. Chavez, and G. Cooper. The alarm monitoring system: a case study with two probabilistic inference techniques for belief networks. In *Proceedings Artificial Intelligence in Medical Care*, pages 247–256, 1989.

[3] R.R. Bouckaert. Optimizing causal orderings for generating DAGs from data. In *Proceedings Uncertainty in Artificial Intelligence 8*, pages 9–16, 1992.

[4] C.K. Chow and C.N. Liu. Approximating discrete probability distributions with dependency trees. *IEEE Transactions on Information Theory*, IT-14:462–467, 1986.

[5] G.F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, pages 309–347, 1992.

[6] R.L. Graham, D.E. Knuth, and O. Patashnik. *Concrete mathematics.* Addison-Wesley, 1989.

[7] D. Heckerman, E. Horvitz, and B. Nathwani. Towards normative expert systems: Part I, the pathfinder project. *Methods of Information in Medicine*, 31:90–105, 1992.

[8] M. Henrion. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In *Proceedings Uncertainty in Artificial Intelligence 4*, pages 149–163, 1988.

[9] M. Henrion. An introduction to algorithms for inference in belief nets. In *Proceedings Uncertainty in Artificial Intelligence 6*, pages 129–138, 1990.

[10] E. Herskovits. *Computer-based probabilistic-network construction.* PhD thesis, Section of Medical Informatics, University of Pittsburgh, 1991.

[11] S. Højsgaard and B. Thiesson. Bifrost– block recursive models induced from relevant knowledge, observationsm and statistical techniques. Technical Report R 92-2010, Institute for Electronic Systems, University of Aalborg, Denmark, 1992.

[12] S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their applications to expert systems (with discussion). *Journal of the Royal Statistical Society B*, 50:157–224, 1988.

[13] S.L. Lauritzen, B. Thiesson, and D.J. Spiegelhalter. Diagnostic systems created by model selection methods – a case study. In *Proceedings 4th International Workshop on AI and Statistics*, 1993.

[14] D. Madigan and J. York. Bayesian graphical models for discrete data. Technical Report 259, Department of Statistics, University of Washington, Seattle, 1993.

[15] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufman, inc., San Mateo, CA, 1988.

[16] J. Pearl, D. Geiger, and T. Verma. The logic of influence diagrams. In R.M. Oliver and J.Q. Smith, editors, *Influence Diagrams, Belief Nets and Decision Analysis*, pages 67–87. John Wiley & Sons Ltd., 1990.

[17] G. Rebane and J. Pearl. The recovery of causal polytrees from statistical data. In *Proceedings Uncertainty in Artificial Intelligence 3*, pages 222–228, 1987.

[18] J. Rissanen. Stochastic complexity and modeling. *Annals of Statistics*, 14(3):1080–1100, 1986.

[19] J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society B*, 49(3):223–239, 1987.

[20] R.D. Robinson. Counting unlabeled acyclic digraphs. In *Proceedings of the fifth Australian Conference on Combinatorial Mathematics*, pages 28–43, 1976.

[21] D.J. Spiegelhalter, A.P. Dawid, S.L. Lauritzen, and R.G. Cowell. Bayesian analysis in expert systems. *Statistical Science*, 8:219–283, 1993.

[22] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search.* ??, 1993.

[23] T. Verma and J. Pearl. Causal networks: semantics and expressiveness. In *Proceedings Uncertainty in Artificial Intelligence 4*, pages 352–359, 1988.

[24] N. Wermuth and S.L. Lauritzen. Graphical and recursive models for contingency tables. *Biometrika*, 72:537–552, 1983.