Appears in Proceedings of
2nd Pacific Rim International Conference on Artificial Intelligence 1992
Pages 1237–1243.

Learning Bayesian Belief Networks An approach based on the MDL Principle*

Wai Lam and Fahiem Bacchus
Department of Computer Science
University of Waterloo
Waterloo, Ontario,
Canada, N2L 3G1

Abstract A new approach for learning Bayesian belief networks from raw data is presented. The approach is based on Rissanen's Minimal Description Length (MDL) principle, which is particularly well suited for this task. Our approach does not require any prior assumptions about the distribution being learned. In particular, our method can learn *unrestricted multiply-connected* belief networks. Furthermore, unlike other approaches our method allows us to tradeoff accuracy against complexity in the learned model. This is important since if the learned model is very complex (highly connected), it can be computationally intractable to use. In such a case it would be preferable to use a simpler model even if it is less accurate. MDL offers a principled method for making this tradeoff. We also show that our method generalizes previous approaches based on Kullback cross-entropy. Experiments have been conducted to demonstrate the feasibility of the approach.

1 Introduction Bayesian belief networks, advanced by Pearl [9], have become an important paradigm for representing and reasoning with uncertainty. Systems based on Bayesian networks have been constructed in a number of different application areas, ranging from medical diagnosis, e.g., [2], to reasoning about the oil market, e.g., [1]. Despite these successes, a major obstacle to using Bayesian networks lies in the difficulty of constructing them in complex domains. It can be a very time-consuming and error-prone task to specify a network that can serve as an accurate probabilistic model of the problem domain; there is a knowledge engineering bottleneck. Clearly, any mechanism that can help automate this task would be beneficial. A promising approach to this problem is to try to construct, or learn, such network representations ¿from raw data. In many areas raw data can be obtained from databases of records. If techniques can be developed for automat-

lem, but it will also facilitate the automatic refinem of the representation as new data is accumulated.

In this paper we present a new approach to learn Bayesian networks. Our method can discover arbitr network structures from raw data without relying *any* assumptions about the underlying probability tribution that generated the data. In particular, method can learn unrestricted *multiply-connected works*. Multiply-connected networks are more exp sive than tree or polytree networks, and that extra pressiveness is sometimes essential if the network is be a sufficiently accurate model of the underlying dis bution. Our approach is theoretically founded on R sanen's Minimum Description Length (MDL) Princi [13].

It is well known that multiply-connected Bayes networks are in the worst case computationally tractable to reason with; to be precise the reason algorithms are NP-Hard [4]. The complexity of reas ing with a particular network is a function of its c nectivity; the more connected it is the more diffic is reasoning. Hence, there is limited utility in lea ing a multiply-connected network that is too comp to support efficient reasoning. We feel that the m advantage of our approach is that it offers a princip method, the MDL principle, of trading off the compl ity and accuracy of the learned model. It will learn a l complex network if that network is sufficiently accura and at the same time, unlike some previous methods is still capable of learning complex networks if no sim network is sufficiently accurate.

This is particularly important when learning from r data as we do not have direct access to the underly distribution. Instead we can only approximate that tribution through the data that it has generated. Si our information is only approximate it seems inapp priate to try to recover the "true" structure. Rather

more complex and more accurate one.[1]

The MDL principle says that the best model of a set of data is that model which minimizes the sum of the encoding lengths of the data and the model itself. That is, with the aid of the model we can represent, or encode, the data more compactly, by exploiting probabilistic regularities described by the model. However, the model itself will require some representation. The MDL principle specifies that both these components should be taken into consideration. More accurate models minimize the encoding length of the data, but the more complex a model is, the longer will be its encoding. Hence, by minimizing the sum of these two factors the MDL principle offers a tradeoff between complexity and accuracy.

Finding the network (model) that minimizes the sum of these two components is a computationally intractable task however: there are simply too many networks to search. Hence, our realization of the MDL principle is based on a heuristic search algorithm that tries to find a network that has low, but not necessarily minimum, description length. We have conducted a number of experiments that successfully demonstrate the feasibility of our method.

In the sequel we will first discuss related work on learning Bayesian Networks. Then we will discuss in more detail the MDL principle and the manner in which it can be applied to the task at hand. A discussion of our heuristic algorithm follows along with a presentation of our empirical results. We conclude with some discussion of future work.

2 Related Work The earliest work that can be viewed as learning network models was that of Chow and Liu [3]. Their approach was able to recover simple tree-structured belief networks from a database of records. If the database was generated by a distribution that had a tree-structure, it could be exactly recovered. Otherwise their method guaranteed that the probability distribution of the learned tree network was the closest of all tree networks to the underlying distribution of the raw data. The criterion of "closeness" they used was based on the well-known Kullback-Leibler cross-entropy measure [7]. The main restriction of this work was that it could only learn tree structures. Hence, if the raw data was the result of a non-tree structured distribution, the learned structure could be very inaccurate. Rebane and Pearl [12] extended Chow and Liu's methods to the recovery of networks of singly connected trees (polytrees). If the underlying distribution had a polytree structure, its topological structure could be exactly recovered (modulo the orientation of some of the arcs). But again if the raw data came from a non-polytree distribution, the learned structure could be very inaccurate. Recently, Geiger and Pearl [4] have developed a method that can discover a minimal-edge I-map[10]. However, their approach is again limited to polytrees; it is only guaranteed to work in the case where the underlying distribution has an exact polytree structure.

All of the above approaches fail to recover the richer and more realistic class of multiply-connected networks which topologically are directed acyclic graphs (dags). Recently, Spirtes et al. [16] have developed an algorithm that can construct multiply-connected networks. And Verma and Pearl [17, 11] have developed what they call an IC-Algorithm that can also recover these kinds of structures. However, both approaches require that the underlying distribution being learned be *dag-isomorphic*.[2] But, not all distributions are. As a result both of these methods have the common drawback that they are not guaranteed to work when the underlying distribution fails to be dag-isomorphic. In such cases no conclusions can be drawn about the closeness of fit between the learned structure and the underlying distribution.

All of these methods share the common disadvantage that they make assumptions about the underlying distribution. Unfortunately, we are hardly ever in a position to know the underlying distribution. This is why we are trying to learn! Hence, we have no assurance that these methods will work well in practice. These methods might produce very inaccurate models if the underlying distribution fails to fall into the category of distributions they can deal with. Nevertheless, these works have provided a great deal of information pertinent to learning Bayesian networks.

An interesting alternate approach which can deal with multiply-connected networks is that of Cooper and Herskovits [5]. Their approach tries to find the most probable network using a Bayesian approach. As with other Bayesian approaches, they must assume a prior distribution over the space of all possible network structures. They have taken this prior to be uniform.[3] Unfortunately, it seems to us that this is the wrong choice. By choosing this prior their method will always prefer a more accurate network, even if that network is *much* more complex and only slightly more accurate. Given that we must perform learning with only a limited amount of data, this insistence on accuracy is questionable.

One way of viewing the MDL principle is as a Bayesian approach in which the prior distribution over the models is inversely related to their encoding length, i.e., their complexity. Hence, the MDL principle has a bias towards learning models that are as simple as possible. This seems to us to be a far more reasonable approach given that the data is only approximately representative of the underlying distribution. Another advantage

ponents of the model, including, e.g., the conditional probabilities that parameterize the network; although we have not done this yet. In Cooper and Herskovits's approach they must also place a prior distribution on these parameters, and again it is not clear that their choice of a uniform distribution is the appropriate one.

Cooper and Herskovits face the same problem as we do: the space of possible network structures is simply too large to explore. Hence, they also develop a heuristic method that searches a constrained set of structures looking, in their case, for the one with highest posterior probability, and in our case for the one with minimal description length. The heuristic method they choose depends on an inputted ordering of the variables, and the network that they learn respects this ordering (i.e., parents of a node are always lower in the ordering). The heuristic method we develop, however, does not require such an ordering, which is an advantage in situations where there is insufficient causal information to generate a total ordering.

3 The MDL Principle The MDL principle is based on the idea that the best model of a collection of data items is the model that minimizes the sum of (1) the length of the encoding of the model, and (2) the length of the encoding of the data *given the model*, both of which are measured in bits.

To apply the MDL principle to Bayesian networks we need to specify how we can perform the two encodings, the network itself (1) and the raw data given a network (2).

3.1 Encoding the Network To represent a particular Bayesian network, the following information is necessary and sufficient: (a) A list of the parents of each node, and (b) the set of conditional probabilities associated with each node that are required to parameterize the network.

Suppose there are n nodes in the problem domain. For a node with k parents, we need $k \log_2(n)$ bits to list its parents. To represent the conditional probabilities, the encoding length will be the product of the number of bits required to store the numerical value of each conditional probability and the total number of conditional probabilities that are required. In a Bayesian network, a conditional probability is needed for every distinct instantiation of the parent nodes and node itself (except that one of these conditional probabilities can be computed ¿from the others due to the fact that they all sum to 1). For example, if a node that can take on 5 distinct values has 4 parents each of which can take on 3 distinct values, we will need $3^4 \times (5-1)$ conditional probabilities. Hence, under this simple scheme the total description length for a particular network will be:

$$\sum_{i=1}^{n}$$

domain, n and d will be constants. This is not only encoding scheme possible, but it is simple and performs well in our experiments.

By looking at this equation, we see that highly connected networks require longer encodings. First, many nodes the list of parents will get larger, and second the list of conditional probabilities we need to store for that node will also increase. In addition, networks in which nodes that have a larger number of values have parents with a large number of values will require longer encodings. Hence, the MDL principle will tend to favor networks in which the nodes have a smaller number of parents (i.e., networks that are less connected) and against networks in which nodes taking on a large number of values are not parents of nodes that also take on a large number of values.

It also happens that for Bayesian networks the degree of connectivity is closely related to the computational complexity of using the network. For example, extremely efficient algorithms exist for trees, and tractable (polynomial) algorithms exist for singly connected networks [10].[4] Hence, our encoding scheme generates a preference for more efficient networks. The encoding length of the model is, however, not the only factor determining the description length; we also have to consider the encoding length of the data.

3.2 Encoding the Data Using the Model Let us first be more precise about the form of the raw data. The task is to learn the joint distribution of a collection of random variables $X = \{X_1, \ldots, X_n\}$. Each variable X_i has an associated collection of values $\{x_i^1, \ldots, x_i^k\}$ that it can take on, where the number of values k may in general depend on i. Every distinct choice of values for the variables in X defines an atomic event in the underlying joint distribution and is assigned a particular probability by that distribution.

For example, we might have three random variables X_1, X_2, and X_3, with X_1 having $\{1, 2\}$, X_2 having $\{1, 2, 3\}$, and X_3 having $\{1, 2\}$ as possible values. There are $2 \times 3 \times 2$ different complete instantiations of the variables. Each of these is an atomic event in the underlying joint distribution, and has a particular probability of occurring. For example, the event in which $\{X_1 = 1, X_2 = 3, X_3 = 1\}$ is one of these atomic events.

We assume that the data points in the raw data are atomic events. That is, each data point specifies a value for every random variable in X. Furthermore, we assume that the data points are the result of independent random trials. Hence, we would expect, via the central limit theorem, that each particular instantiation of the variables would appear in the database with a relative frequency approximately equal to its probability. These assumptions are standard ones in work in this area.

and an unbiased estimator for node X_i taking on the value v when its parents in the network take on values represented by u is $N_{v,u}/N_u$, where $N_{v,u}$ is the number of data points in which X_i and its parents take on the values v and u, and N_u is the number of data points in which X_i's parents take on the values u.

Given our Bayesian network model we can calculate the probability q_i (according to our model) of every atomic event e_i. Given that we are using the model as a best "guess" representation of the underlying probabilities, the optimal encoding of the data using the probabilities q_i will use approximately $-\log_2(q_i)$ bits to encode each occurrence of the event e_i, i.e., each data point representing event e_i will require that many bits in the encoding.

For example, given the set of variables X_1, X_2 and X_3 as above, our model might assign probability $1/2$ to the event $e_1 = \{X_1 = 1, X_2 = 3, X_3 = 1\}$ and probability $1/4$ to the event $e_2 = \{X_1 = 2, X_2 = 2, X_3 = 1\}$. We could then use the binary code 1 to represent e_1 and the code 01 to represent e_2 reserving the longer codes 001, 0001, etc., for the other less probable events. If the database consists of the sequence of events e_1, e_1, e_2, we could encode it as the 4 bit sequence 1101.[5] Here the database has twice as many occurrences of e_1 as e_2; the probabilities predicted by our model are corroborated by the database. However, if the database consisted of the event sequence e_2, e_2, e_1, the encoding dictated by our model would require a 5 bit sequence 01011 to encode the database. In this case a model that reversed the probability assignments to e_1 and e_2 would have yielded a shorter encoding of the database; such a model would represent e_2 with the shorter code rather than e_1.

If the true probability of event e_i was p_i and the database consisted of N data points, we would expect that on average there would be Np_i occurrences of e_i in the database. Hence, given a model that assigns probability q_i to event e_i, it would require

$$-N\sum_i p_i \log_2(q_i) \qquad (2)$$

bits to encode the database. The following theorem, due to Gibbs [13], provides important information about the properties of this encoding.

Theorem 3.1 (Gibbs) *Let p_i and q_i, $i = 1, \ldots, n$, be non-negative real numbers that sum to 1. Then*

$$-\sum_{i=1}^{n} p_i \log_2(p_i) \leq -\sum_{i=1}^{n} p_i \log_2(q_i),$$

with equality holding if and only if $p_i = q_i$, where we take $0\log_2(0)$ to be 0.

This theorem implies that on average the encoding of the data is minimized only by an absolutely accurate model, i.e., a model that assigns probabilities q_i that are equal to the true underlying probabilities p_i.

Definition 3.2 [Kullback-Leibler Cross-Entropy] L
P and Q be distributions defined over the same ev
space. The Kullback-Leibler cross-entropy between
and Q, $C(P,Q)$, is a measure of how close Q is to
and is defined by the equation

$$C(P,Q) = \sum_i p_i(\log_2(p_i) - \log_2(q_i)).$$

It follows from Gibbs's theorem that this quantity
always non-negative and that it is zero if and only
$P \equiv Q$, i.e., $\forall i.q_i = p_i$.

¿From Equation 2 if follows that the minimal possi
encoding length of the data will be $-N\sum_i p_i \log_2($
Hence, when using a model that assigns probabilities
the encoding length will increase by $N(\sum_i p_i(\log_2(p_i$
$\log_2(q_i)))$. That is, we have the following theorem.

Theorem 3.3 *The encoding length of the data is
monotonically increasing function of the cross-entro
between the distribution defined by the model and
true distribution.*

In previous work Chow and Liu [3] developed
method for finding a tree structure that minimized
cross-entropy, and their method was extended by J
bane and Pearl [12] to finding polytrees with minim
cross-entropy. This theorem shows that in a cert
sense the MDL principle can be viewed as a gener
ization of these approaches. If we were to ignore
complexity (encoding length) of the model and were
restrict the class of models being examined, the M
principle would duplicate their results. The advanta
of considering both the data and the model (i.e., the s
of Equations 1 and 2) is that we can learn a more co
plex model if no simpler model is sufficiently accura
i.e., if every simpler model has very high cross-entro

4 Applying the MDL Principle In theory
MDL principle can be applied by simply examining
ery possible Bayesian network that can be construc
over our set of random variables X. For each of th
networks we could evaluate the encoding length of
data and of the network searching for the network t
minimized the sum of these encodings.

However, this approach is impractical as there
an exponential number of networks over n variable
Hence, we must resort to a heuristic search through
space of possible networks trying to find one that yie
a low, albeit not necessarily minimal, sum of Eq
tions 1 and 2.

We accomplish this search by dividing the probl
into two. There can be between 0 and $n(n-1)/2$ a
in a dag. For each possible number of different arcs
search heuristically for a network with that many a
and low cross-entropy. By Theorem 3.3 we know t
this network will yield a relatively low encoding leng
for the data. We then examine these different networ

To perform the first part of the search, i.e., to find a network with low cross-entropy, we develop some additional results that are based on the work of Chow and Liu [3].

4.1 Evaluating Cross-Entropy The underlying distribution P is a joint distribution over the variables $X = \{X_1, \ldots, X_n\}$, and any Bayesian network model will also define a joint distribution Q over these variables. Using this notation the equation for the cross-entropy between P and Q becomes

$$C(P,Q) = \sum_X P(X) \log_2 \frac{P(X)}{Q(X)},$$

where the sum extends over all distinct vectors of values of the variables in X, i.e., all atomic events.

In an arbitrary Bayesian network $Q(X)$ will take the form [10]:

$$\begin{aligned} Q(X) &= Q(X_1 \mid F_{X_1}) Q(X_2 \mid F_{X_2}) \ldots Q(X_n \mid F_{X_n}) \\ &= P(X_1 \mid F_{X_1}) P(X_2 \mid F_{X_2}) \ldots P(X_n \mid F_{X_n}), \end{aligned} \quad (4)$$

where F_{X_i} is the, possibly empty, set of parents of X_i. We can replace the terms $Q(X_i|F_{X_i})$ by $P(X_i|F_{X_i})$ since we are estimating these conditional probability terms, i.e., the parameters of the Bayesian network, through frequency counts taken over the raw data (as described above). This equality assumes that these estimates are approximately equal to the true underlying values $P(X_i|F_{X_i})$. By the central limit theorem they will be close, with high probability, if we have a sufficient number of data points.

We can extend Chow and Liu's work by defining a weight measure for a node, X_i, with respect to its parents as follows:

$$W(X_i, F_{X_i}) = \sum_{X_i, F_{X_i}} P(X_i, F_{X_i}) \log_2 \frac{P(X_i, F_{X_i})}{P(X_i) P(F_{X_i})} \quad (5)$$

where we are summing over all possible values that X_i and its parents F_{X_i} can take. And we can prove the following theorem.

Theorem 4.1 *$C(P,Q)$ is a monotonically decreasing function of $\sum_{i=1, F_{X_i} \neq \emptyset}^{n} W(X_i, F_{X_i})$. Hence, it will be minimized if and only if the sum is maximized.*

The proof of this and the other theorems is given in our full report [8]. The summation term is the total weight of the directed acyclic graph according to the weight measure defined in Equation 5.

In conclusion, given probabilities computed from the raw data, we can calculate the weight of any proposed network structure. Our theorem shows that structures with greater weight are closer to the underlying distribution. If we can find a directed acyclic graph with maximum total weight, then the probability distribution of this structure will be closest to the underlying distribution of the raw data, and thus it will yield the

then $Q \equiv P$. In other words, if we construct multiply-connected network corresponding to the str ture on the right side of the above expression, the pr ability distribution defined by this structure will ab lutely coincide with the underlying distribution of raw data, and hence it will have lowest possible cr entropy and highest possible weight. However, t structure is a complete graph, and worse still, it d not convey any meaning since it can represent any tribution. This indicates that if we allow structures arbitrarily complex topology, we can obtain a triv match with the underlying distribution.

To further understand the problem, consider the lowing theorem.

Theorem 4.2 *Let M_i be the maximum weight of networks that have i arcs, then*

$$i > j \Rightarrow M_i \geq M_j.$$

That is, we can always increase the quality of learned network, i.e., decrease the error in the se of decreasing the cross-entropy, by increasing the to logical complexity, i.e., by learning networks with m arcs. It is by considering in addition the encoding leng of the network that we resolve this difficulty.

4.2 Searching for Low Cross-Entropy Networ Given our ability to evaluate the cross-entropy of a n work through an evaluation of its weight, we have dev oped a heuristic search algorithm that uses local sea to find networks with low cross-entropy. We sea for low cross-entropy networks with varying numb of arcs, and then we choose among the networks fou that one which minimizes the total description leng i.e., that is best by the MDL principle.

A complete description of the heuristic search al rithm is given in our full report [8]. In empirical tests this algorithm we have found that when provided w time polynomial in the number of data points and number of variables (nodes in the net), the search pro dure can successfully find good networks models of raw data. Furthermore, it can find such models wi out being provided with a prior "causality" ordering the variables, as is required by Cooper and Herskovi procedure [5].

5 Experimental Results A common approach evaluating various learning algorithms has been to g erate raw data from a predetermined network and t to compare the network learned from that data w the original, the aim being to recapture the origi For example, this is the technique used by Cooper a Herskovits [5]. An implicit assumption of this appro is that the aim of learning is to reconstruct the true tribution. However, if one takes the aim of learning be the construction of a *useful* model, i.e., one that i

with the original. Our approach involves a measure of the closeness between two networks. This measure is actualized in two different ways, one using Kullback-Leibler cross-entropy and the other using an average of the difference between the distributions specified by the two networks evaluated a various points. The details of our closeness measure are given in the full report [8].

We have performed three sets of experiments to demonstrate the feasibility of our approach. The first set of experiments consisted of a number of Bayesian networks that were composed of small number of variables (5) as shown in Figure 1. Some of these structures are multiply-connected networks.

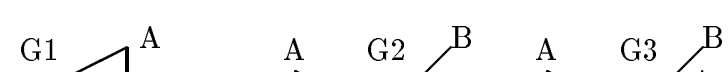
The second experiment consisted of learning a Bayesian network with a fairly large number of variables (37 nodes and 46 arcs). This network was derived from a real-world application in medical diagnosis [2] and is known as the ALARM network (see [8] for a diagram of this network).

The third experiment consisted of learning a small Bayesian network, as shown in in Figure 2. We experimented by varying the conditional probability parameters of this network. Here the aim was to demonstrate that our procedure could often learn a simpler network that was very close to the original.

During the first set of experiments after calculating the description lengths of the networks, the network with the minimum description length was selected. In all these cases we found that the learned network was exactly the same as the one used to generate the raw data.

In the second experiment the Bayesian network recovered by the algorithm was found to be close to the original network structure. Two different arcs and three missing arcs were found, out of 46 arcs. Furthermore, our evaluated closeness between the original network and this learned structure was very small, under both of our measures. One additional feature of our approach, in particular a feature of our heuristic search algorithm, is that we did not require a user supplied ordering of variables (cf. Cooper and Herskovits [5]). We feel that this experiment demonstrates that our approach is feasible for recovering Bayesian networks of practical size.

In the third set of experiments, the original Bayesian network $G4$ consisted of 5 nodes and 5 arcs. We varied the conditional probability parameters during the process of generating the raw data obtaining four different sets of raw data. Exhaustive searching was then carried out and the MDL learning algorithm was applied to each of these sets of raw data. Different learned



structures were obtained, all of which were extrem close to the original network as measured by both our distance formulas. In one case the original netw was recovered.

This experiment demonstrates that our algorit yields a tradeoff between accuracy and complexity the learned structures: in all cases where the or nal network was not recovered a simpler network learned. The type of structure learned depends on parameters, as each set of parameters, in conjunct with the structure, defines a different probability dis bution. Some of these distributions can be accurat modeled with simpler structures. In the first case, distribution defined by the parameters did not hav simpler model of sufficient accuracy, but in the ot cases it did.

6 Conclusions We have argued in this paper t the purpose of learning a Bayesian network from r data is not to recover the underlying distribution, as t distribution might be too complex to use. Rather, should attempt to learn a useful model of the underly phenomena. Hence, there should be some tradeoff tween accuracy and complexity. The MDL principle as its rational this same tradeoff, and it can be natura applied to this particular problem. We have discussed detail how the MDL principle can be applied and h pointed out its relationship to the method of minimiz cross-entropy. Using this relationship we have exten the results of Chow and Liu relating cross-entropy t weighing function on the nodes. This has allowed us develop a heuristic search algorithm for networks t minimize cross-entropy. These networks minimize encoding length of the data, and when we also consi the complexity of the network we can obtain mod that are good under the MDL metric. Our experim tal results demonstrate that our algorithm does in f perform this tradeoff, and further that it can be appl to networks of reasonable size.

There are a number of issues that arise which requ future research. One issue is the search mechanism. are currently dividing the task into first searching fo network that minimizes the encoding length of the d and then searching through the resulting networks one that minimizes the total description length. T method has been successful in practice, but we are a investigating other mechanisms. In particular, it see reasonable to combine both phases into one search. A other important component that has not yet been dressed is the accuracy of the raw data. In general, th will be a limited quantity of raw data, and certain rameters can only be estimated with limited accura We are investigating methods for taking into acco the accuracy of the data in the construction. For exa

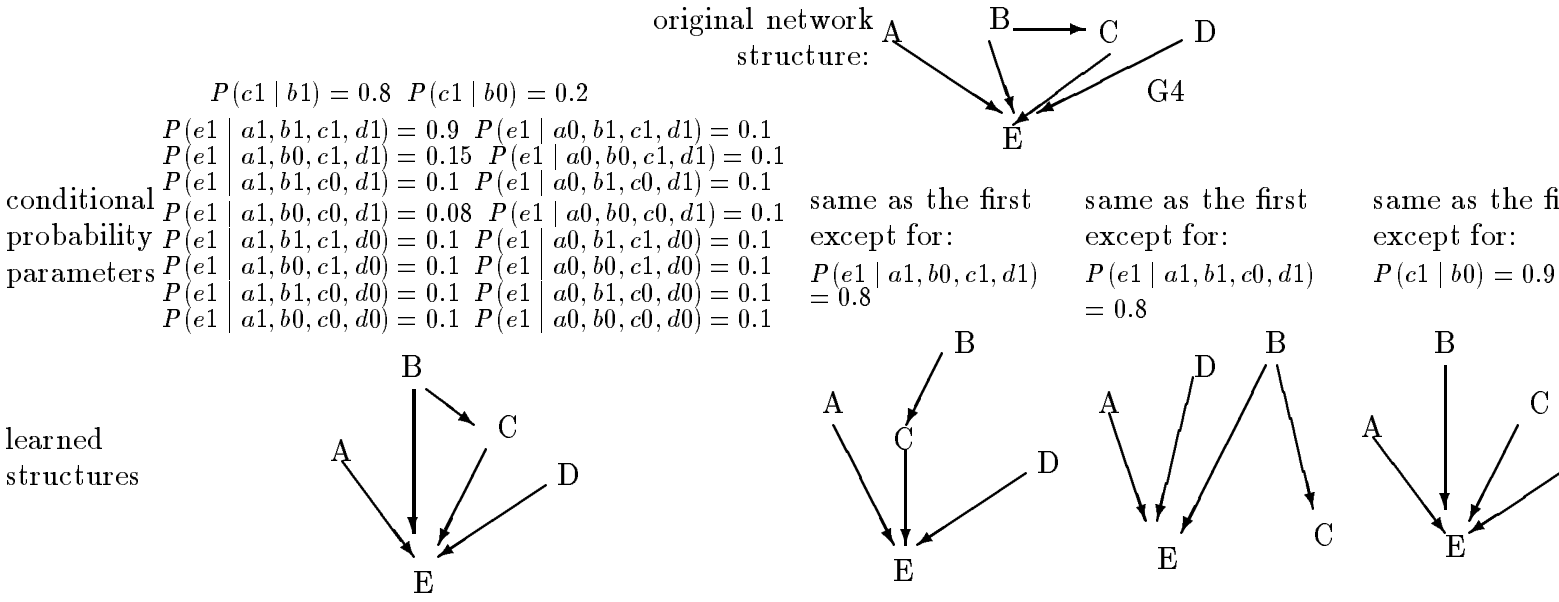


Figure 2: The Quality of Learned Networks

we might know of causal relationships in the domain that bias us towards making certain nodes parents of other nodes. The issue that arises is how can this information be used during learning. We are investigating some approaches to this problem.

References

[1] B. Abramson. ARCO1: An application of belief networks to the oil market. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 1–8, 1991.

[2] 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 2nd European Conference on Artificial Intelligence in Medicine*, pages 247–256, 1989.

[3] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[4] G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

[5] G. F. Cooper and E. Herskovits. A Bayesian method for constructing Bayesian belief networks from databases. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 86–94, 1991.

[6] D. Geiger, A. Paz, and J. Pearl. Learning causal trees from dependence information. In *Proceedings of the AAAI National Conference*, pages 770–776, 1990.

ple. Technical Report CS 92-39, University of W terloo, 1992.

[9] J. Pearl. Fusion, propagation, and structuring belief networks. *Artificial Intelligence*, 29:241–2 1986.

[10] J. Pearl. *Probabilistic Reasoning in Intelligent S tems : Networks of Plausible Inference*. Morg Kaufmann, San Mateo, California, 1988.

[11] J. Pearl and T. S. Verma. A theory of infer causation. In *Proceedings of the 2nd Internatio Conference on Principles of Knowledge Repres tation and Reasoning*, pages 441–452, 1991.

[12] G. Rebane and J. Pearl. The recovery of cau poly-trees from statistical data. In *Proceedings the Conference on Uncertainty in Artificial Inte gence*, pages 222–228, 1987.

[13] J. Rissanen. Modeling by shortest data descripti *Automatica*, 14:465–471, 1978.

[14] J. Rissanen. *Stochastic Complexity in Statist Inquiry*. World Scientific, 1989.

[15] R. W. Robinson. Counting unlabeled acyclic graphs. In *Proceedings of the 5th Australian C ference on Combinatorial Mathematics*, pages 43, 1976.

[16] C. Spirtes, P. Glymour and R. Scheines. Causal from probability. In *Evolving Knowledge in Natu Science and Artificial Intelligence*, pages 181–1 1990.

[17] T. Verma and J. Pearl. Equivalence and synthe of causal models. In *Proceedings of the Confere on Uncertainty in Artificial Intelligence*, pages 22 227, 1990.