Learning Graphical Models by Extending Optimal Spanning Trees

Christian Borgelt and Rudolf Kruse
Dept. of Knowledge Processing and Language Engineering
Otto-von-Guericke-University of Magdeburg
Universitätsplatz 2, D-39106 Magdeburg, Germany
e-mail: {borgelt,kruse}@iws.cs.uni-magdeburg.de

Abstract: In learning graphical models we often face the problem that a good fit to the data may call for a complex model, while real time requirements for later inferences force us to strive for a simpler one. In this paper we suggest a learning algorithm that tries to achieve a compromise between the goodness of fit of the learned graphical model and the complexity of inferences in it. It is based on the idea to extend an optimal spanning tree in order to improve the fit to the data, while restricting the extension in such a way that the resulting graph has hypertree structure with maximal cliques of at most size 3.

Keywords: Graphical Model, Learning from Data, Optimal Spanning Tree

1 Introduction

In recent years graphical models [18, 12]—especially Bayesian networks [14, 8] and Markov networks [11], but also the more general valuation-based networks [17] and, though to a lesser degree, the newer possibilistic networks [6, 1]—gained considerable popularity as powerful tools to model dependences in complex domains and thus to make inferences under uncertainty in these domains feasible. Graphical models are based on the idea that under certain conditions a multidimensional (probability or possibility) distribution can be decomposed into (conditional or marginal) distributions on lower dimensional subspaces. This decomposition is represented by a graph, in which each node stands for an attribute and each edge for a direct dependence between two attributes. The graph representation also supports drawing inferences, because the edges indicate the paths along which evidence has to be transmitted [8, 2]. However, in order to derive correct and efficient evidence propagation methods, the graphs have to satisfy certain conditions. In general, cycles pose problems, making it possible that the same information can travel on different routes to other attributes. In order to avoid erroneous results in this case, the graphs are often transformed into singly connected structures, namely so-called *join* or *junction trees* [11, 8, 2].

Since constructing graphical models manually can be tedious and time consuming, a large part of recent research has been devoted to learning them from a dataset of sample cases [4, 7, 5, 6, 1]. However, many known learning algorithms do not take into account that the learned graphical model may later be used to draw time-critical inferences and that in this case the time complexity of evidence propagation may have to be restricted, even if this can only be achieved by accepting approximations. The main problem is that during join tree construction edges may have to be added, which can make the graph more complex than is acceptable. In such situations it is desirable that the complexity of the join tree can be controlled at learning time, even at the cost of a less exact representation of the domain under consideration.

To achieve this we suggest an algorithm that constructs a graphical model by extending an optimal spanning tree in such a way that the resulting graph has hypertree structure with maximal cliques of at most size 3.

2 Optimal Spanning Trees

Constructing an optimum weight spanning tree is a special case of methods that learn a graphical model by measuring the strength of marginal dependences between attributes. The idea underlying these heuristic, but often highly successful approaches is the frequently valid assumption that in a graphical model correctly representing the probability or possibility distribution on the domain of interest an attribute is more strongly dependent on adjacent attributes than on attributes that are not directly connected to it. Consequently, it should be possible to find a proper graphical model by selecting edges that connect strongly dependent attributes. Among the methods based on this idea constructing an optimum weight spanning tree is the simplest and best known learning algorithm. It is at the same time the oldest approach, as it was suggested as early as 1968 in [3].

In general, the algorithm consists of two components: an evaluation measure, which is used to assess the strength of dependence of two attributes, and a method to construct an optimum weight spanning tree from given edge weights (which are, of course, provided by the evaluation measure). The latter component may be, for example, the well-known Kruskal algorithm [10]. For the former component, i.e., the evaluation measure, there is a variety of measures to choose from. In [3], in which learning probabilistic graphical models was considered, *mutual information* (also called *information gain* or *cross entropy*) was used. It is defined as (A and B are attributes):

$$I_{\mathrm{mut}}(A, B) \;=\; H(A) + H(B) - H(AB),$$

where $H(A)$ is the *Shannon entropy* of the probability distribution on A, i.e.,

$$H(A) \;=\; -\sum_{a\in\mathrm{dom}(A)} P(a)\log_2 P(a).$$

(Here $P(a)$ is an abbreviation of $P(A = a)$ and denotes the probability that A assumes—as a random variable—the value a.) $H(B)$ and $H(AB)$ are defined analogously. Alternatively, one may use the χ^2 *measure*

$$\chi^2(A, B) = N \sum_{\substack{a\in\mathrm{dom}(A)\\ b\in\mathrm{dom}(B)}} \frac{(P(a)P(b) - P(a, b))^2}{P(a)P(b)},$$

where N is the number of cases in the dataset to learn from (which is often dropped in applications), or the *symmetric Gini index* (see, for example, [1] for a definition), etc.

While the above measures are designed for learning probabilistic networks, it is clear that the same approach may also be used to learn possibilistic networks: We only have to choose a measure for the possibilistic dependence of two attributes. Best known among such measures is the *specificity gain*

$$S_{\mathrm{gain}}(A, B) = \mathrm{nsp}(A) + \mathrm{nsp}(B) - \mathrm{nsp}(AB),$$

where $\mathrm{nsp}(A)$ denotes the U-uncertainty measure of nonspecificity [9] of the (marginal) possibility distribution π_A on attribute A:

$$\mathrm{nsp}(A) = \int_0^{\sup(\pi_A)} \log_2 |[\pi_A]_\alpha| \mathrm{d}\alpha.$$

($[\pi_A]_\alpha$ denotes the α-cut of the possibility distribution.) $\mathrm{nsp}(B)$ and $\mathrm{nsp}(AB)$ are defined analogously. It should be noted that the formula of specificity gain is very similar to the formula of information gain/mutual information due to the fact that in possibility theory the measure of nonspecificity plays roughly the same role Shannon entropy plays in probability theory.

Alternatively, one may use *possibilistic mutual information* [1]:

$$d_{\mathrm{mi}}(A, B) \\ = -\sum_{\substack{a\in\mathrm{dom}(A)\\ b\in\mathrm{dom}(B)}} \pi_{AB}(a, b)\log_2 \frac{\pi_{AB}(a, b)}{\min\{\pi_A(a), \pi_B(b)\}},$$

which is based on a translation of a different way of writing mutual information to the possibilistic setting (see [1] for details) or a possibilistic version of the χ^2 measure [1]:

$$d_{\chi^2}(A, B) \\ = \sum_{\substack{a\in\mathrm{dom}(A)\\ b\in\mathrm{dom}(B)}} \frac{(\min\{\pi_A(a), \pi_B(b)\} - \pi_{AB}(a, b))^2}{\min\{\pi_A(a), \pi_B(b)\}}.$$

It is worth noting that the optimum weight spanning tree approach has an interesting property in the probabilistic setting: Provided that there is a perfect tree-structured graphical model of the domain of interest and the evaluation measure used has a certain property (at least mutual information and the χ^2 measure have this property), then the perfect model can be found by constructing an optimum weight spanning tree (see [1] for details). For mutual information even more can be shown: Constructing an optimum weight spanning tree with this measure yields the best tree-structured approximation of the probability distribution on the domain of interest w.r.t. *Kullback-Leibler information divergence* [3, 14].

Unfortunately, these properties do not carry over to the possibilistic setting. Even if there is a perfect graphical model with tree structure, constructing an optimum weight spanning tree with any of the possibilistic measures mentioned above is not guaranteed to find this tree (see [1] for a counterexample). As a consequence there is no analog of the stronger approximation statement either.

3 Extending Spanning Trees

Even if there is no perfect tree-structured graphical model of the domain of interest, constructing an optimum weight spanning tree can be a good starting point for learning a graphical model. The algorithm suggested in [16], for example, starts by constructing a(n undirected) spanning tree and then turns it into a (directed) polytree by directing the edges based on the outcomes of conditional independence tests. The advantage of this approach is that it keeps the single-connectedness of the graph and thus allows for a simple derivation of evidence propagation methods. However, by doing so, it does not really restrict the complexity of later inferences, as this complexity depends on the number of parents an attribute has in the polytree. This can be seen by considering the construction of a join tree for the polytree [2]. The first step consists in forming a so-called *moral graph* by "marrying" the parents of an attribute (i.e., connecting them with an edge). In this way the set of parents of an attribute together with the attribute itself become a clique in the resulting graph and thus a node in the final join tree. As the size of the nodes in the join tree is a decisive factor of the complexity of inferences, the number of parents directly determines this complexity. Unfortunately, there is no way to restrict the number of parents in this algorithm. On the other hand, the restriction to singly connected graphs may be too strong for some learning tasks, as such graphs cannot capture certain rather simple dependence structures.

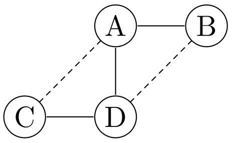

Figure 1: The dotted edges cannot both be the result of "marrying" parents in a directed graph, but may be generated in our algorithm.

To amend these drawbacks, we suggest a simple learning algorithm, which also starts from an initial optimum weight spanning tree, but may yield more complex structures than polytrees, while at the same time restricting the size of the nodes in the join tree. The basic idea of this algorithm is as follows: First an (undirected) optimum weight spanning tree is constructed. Then this tree is enhanced by edges where a conditional independence statement implied by the tree does not hold. However, we do not check arbitrary conditional independence statements, but only those that refer to edges, which connect nodes having a common neighbor in the optimum weight spanning tree. It should be noted that this restriction is similar to directing the edges of the spanning tree, because adding an edge between two nodes having a common neighbor is similar to directing the edges of the spanning tree towards the common neighbor (because the construction of a moral graph would add exactly this edge). However, our approach is more general, since it allows for structures like those shown in figure 1, which cannot result from directing edges alone.

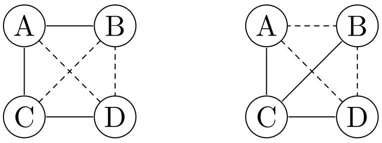

Figure 2: Maximal cliques with four or more nodes cannot be created without breaking the rules for adding edges.

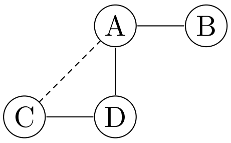

Figure 3: The node A can be bypassed only by an edge connecting the node D to a neighbor of A (which may or may not be B).

A further restriction of the additional edges is achieved by the following requirement: If all edges of the optimum weight spanning tree are removed, the remaining graph must be acyclic. This condition is interesting, because it guarantees that the resulting graph has hypertree structure (a precondition for the construction of a join tree, see [11, 2] for details) and that its maximal cliques comprise at most three nodes. Consequently, with this condition we can restrict the size of the join tree nodes and thus the complexity of inferences.

Theorem: If an undirected tree is extended by adding edges only between nodes with a common neighbor in the tree and if the added edges do not form a cycle, then the resulting graph has hypertree structure and its maximal cliques contain at most three nodes.

Proof: Consider first the size of the maximal cliques. Figure 2 shows, with solid edges, the two possible structurally different spanning trees with four nodes. In order to turn these into cliques the dotted edges have to be added. However, in the graph on the left the edge (B, D) connects two nodes not having a common neighbor in the original tree and in the graph on the right the additional edges form a cycle. Therefore it is impossible to get a clique with a size greater than three without breaking the rules for adding edges.

In order to show that the resulting graph has hypertree structure, it is sufficient to show that all cycles with a length greater than three have a chord (i.e., an edge connecting two nodes of the cycle that are not adjacent in the considered cycle). This is easily verified with the following argument. Neither the original tree nor the graph without the edges of this tree contain a cycle. Therefore in all cycles there must be a node A at which an edge from the original tree meets an added edge. Let the former edge connect the nodes B and A and the latter connect the nodes C and A. Since edges may only be added between nodes that have a common neighbor in the tree, there must be a node D that is adjacent to A as well as to C in the original tree. This node may or may not be identical to B. If it is identical to B and the cycle has a length greater than three, then the edge (B, C) clearly is a chord. Otherwise the edge (A, D) is a chord, because D must also be in the cycle. To see this, consider Figure 3, which depicts the situation referred to. To close the cycle we are studying there must be a path connecting B and C that does not contain A. However, from the figure it is immediately clear that any such path must contain D, because A can only be bypassed via an edge that has been added between D and a neighbor of A (note that this neighbor may or may not be B). □

In order to test for conditional (in)dependence, we simply use the conditional forms of the marginal dependence measures mentioned above. That is, in the probabilistic case we compute for a measure m

$$m_{\text{ci}}(A, B \mid C) = \sum_{c \in \text{dom}(C)} P(c) \cdot m(A, B \mid c),$$

where $m(A, B \mid C = c)$ is defined as $m(A, B)$ with all marginal probabilities $P(a)$ and $P(b)$ replaced by their conditional counterparts $P(a \mid c)$ and $P(b \mid c)$. The possibilistic case is analogous. We only have to take into account that the possibility degrees may not add up be 1, so that normalization is necessary, i.e.,

$$m_{\text{ci}}(A, B \mid C) = \sum_{c \in \text{dom}(C)} \frac{\pi_C(c)}{s} \cdot m(A, B \mid c),$$

net	eds.	pars.	train	test
indep.	0	59	−19921	−20087
orig.	22	219	−11391	−11506
I_{gain}	20	286	−12123	−12340
χ^2	20	283	−12123	−12336
I_{gain}	35	1484	−11454	−12029
χ^2	35	1732	−11441	−12034
I_{gain}	35	1342	−11229	−11818
χ^2	35	1301	−11235	−11805
K2	23	230	−11385	−11511

Table 1: Probabilistic network learning.

where $s = \sum_{c \in \text{dom}(C)} \pi_C(c)$. Based on these measures we select the additional edges greedily (similar to the Kruskal algorithm).

As a final remark we would like to point out that this approach is not guaranteed to find the best possible graph with the stated properties, neither in the probabilistic nor in the possibilistic setting. That is, if there is a perfect graphical model of the domain under consideration, which has hypertree structure and the maximal cliques of which have at most size 3, then this approach may not find it. An example of such a case can be found in [1].

4 Experimental Results

We implemented our algorithm in a prototypical fashion as part of the INES program (Induction of NEtwork Structures) [1] and tested it on the well-known Danish Jersey cattle blood group determination problem [15].

For our probabilistic tests, we used databases randomly generated from a human expert designed Bayesian network for the Danish Jersey cattle domain. Details of the experimental setup can be found in [1]. Table 1 shows the results. The first section contains the results for a network without any edges and the original network, followed by results obtained with a pure optimal spanning tree approach. The third section lists the results of the algorithm suggested in this paper and the final section shows the result of greedy parent selection w.r.t. a topological order. All networks were evaluated by computing the log-likelihood of the training and a test dataset.

net	eds.	pars.	min.	avg.	max.
indep.	0	80	10.06	10.16	11.39
orig.	22	308	9.89	9.92	11.32
S_{gain}	20	415	8.88	8.99	10.71
d_{χ^2}	20	449	8.66	8.82	10.33
d_{mi}	20	372	8.47	8.60	10.39
S_{gain}	29	2110	8.14	8.30	10.13
d_{χ^2}	35	1672	8.10	8.28	10.18
d_{mi}	31	1353	7.97	8.14	10.25
S_{gain}	31	1630	8.52	8.62	10.29
d_{χ^2}	35	1486	8.15	8.33	10.20
d_{mi}	33	774	8.21	8.34	10.42

Table 2: Possibilistic network learning.

For our possibilistic tests we used a database of 500 real world sample cases, which contains a large number of missing values and is thus well suited for a possibilistic approach. The results are shown in table 2. The meaning of the sections is the same as for table 1, although the evaluation is done differently (details about how we assess the quality of a possibilistic network can be found in [1]).

As was to be expected, in both cases, probabilistic as well as possibilistic, the results are in between those of the pure optimum weight spanning tree algorithm and the greedy parent selection algorithm. However, in comparisons with the latter it should be noted that the greedy parent selection needs a topological order to work on and is thus provided with important additional information, while our algorithm relies on the data alone.

5 Conclusions and Future Work

In this paper we suggested a learning algorithm for graphical models, which extends an optimal spanning tree by adding edges. Due to specific restrictions, which edges may be added, the result is guaranteed to have hypertree structure and maximal cliques of limited size, thus providing for efficient inferences. The experimental results are promising, especially for possibilistic networks.

A drawback of the suggested algorithm is that the size of the maximal cliques is restricted to a fixed value, namely 3. Obviously, it would be more desirable if the size restriction were a parameter. Therefore in our future research we plan to search for conditions that enable us to extend optimal spanning trees in more complex ways, while restricting the model to hypertrees with maximal cliques of at most size 4, 5 etc. Unfortunately, such conditions seem to be much more complex and thus difficult to find.

References

[1] C. Borgelt. *Data Mining with Graphical Models.* Ph.D. thesis, University of Magdeburg, Germany 2000

[2] E. Castillo, J.M. Gutierrez, and A.S. Hadi. *Expert Systems and Probabilistic Network Models.* Springer, New York, NY, USA 1997

[3] C.K. Chow and C.N. Liu. Approximating Discrete Probability Distributions with Dependence Trees. *IEEE Trans. on Information Theory* 14(3):462–467. IEEE Press, Piscataway, NJ, USA 1968

[4] G.F. Cooper and E. Herskovits. A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning* 9:309–347. Kluwer, Dordrecht, Netherlands 1992

[5] J. Gebhardt and R. Kruse. Learning Possibilistic Networks from Data. *Proc. 5th Int. Workshop on Artificial Intelligence and Statistics (Fort Lauderdale, FL, USA)*, 233–244. Springer, New York, NY, USA 1995

[6] J. Gebhardt. *Learning from Data: Possibilistic Graphical Models.* Habil. thesis, University of Braunschweig, Germany 1997

[7] D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning* 20:197–243. Kluwer, Dordrecht, Netherlands 1995

[8] F.V. Jensen. *An Introduction to Bayesian Networks.* UCL Press Ltd. / Springer, London, United Kingdom 1996

[9] G.J. Klir and M. Mariano. On the Uniqueness of a Possibility Measure of Uncertainty and Information. *Fuzzy Sets and Systems* 24:141–160. North Holland, Amsterdam, Netherlands 1987

[10] J.B. Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proc. American Mathematical Society* 7(1):48–50. American Mathematical Society, Providence, RI, USA 1956

[11] S.L. Lauritzen and D.J. Spiegelhalter. Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. *Journal of the Royal Statistical Society, Series B*, 2(50):157–224. Blackwell, Oxford, United Kingdom 1988

[12] S.L. Lauritzen. *Graphical Models.* Oxford University Press, Oxford, United Kingdom 1996

[13] R. Lopez de Mantaras. A Distance-based Attribute Selection Measure for Decision Tree Induction. *Machine Learning* 6:81–92. Kluwer, Dordrecht, Netherlands 1991

[14] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA, USA 1988 (2nd edition 1992)

[15] L.K. Rasmussen. *Blood Group Determination of Danish Jersey Cattle in the F-blood Group System (Dina Research Report 8).* Dina Foulum, Tjele, Denmark 1992

[16] G. Rebane and J. Pearl. The Recovery of Causal Polytrees from Statistical Data. *Proc. 3rd Workshop on Uncertainty in Artificial Intelligence (Seattle, WA, USA)*, 222–228. USA 1987.

[17] G. Shafer and P.P. Shenoy. *Local Computations in Hypertrees (Working Paper 201).* School of Business, University of Kansas, Lawrence, KS, USA 1988

[18] J. Whittaker. *Graphical Models in Applied Multivariate Statistics.* J. Wiley & Sons, Chichester, United Kingdom 1990