Local Structure Learning in Graphical Models

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 A topic in probabilistic network learning is to exploit local network structure, i.e., to capture regularities in the conditional probability distributions, and to learn networks with such local structure from data. In this paper we present a modification of the learning algorithm for Bayesian networks with a local decision graph representation suggested in Chickering *et al.* (1997), which is often more efficient. It rests on the idea to exploit the decision graph structure not only to capture a larger set of regularities than decision trees can, but also to improve the learning process. In addition, we study the influence of the properties of the evaluation measure used on the learning time and identify three classes of evaluation measures.

1 Introduction

Probabilistic inference networks—especially Bayesian networks Pearl (1992), but also Markov networks Lauritzen and Spiegelhalter (1988)—are well-known tools for reasoning under uncertainty in multidimensional spaces. The idea underlying them is to exploit independence relations between variables in order to decompose a multivariate probability distribution into a set of (conditional or marginal) distributions on lower-dimensional subspaces. Efficient implementations include HUGIN Andersen *et al.* (1989) and PATHFINDER Heckerman (1991).

Such independence relations have been studied extensively in the field of graphical modeling Kruse *et al.* (1991) and though using them to facilitate reasoning in multidimensional domains has originated in probabilistic reasoning, this approach has been generalized to be usable with other uncertainty calculi Shafer and Shenoy (1988), e.g. in the so-called valuation-based networks Shenoy (1991), and has been implemented e.g. in PULCINELLA Saffiotti and Umkehrer (1991).

Due to their connection to fuzzy systems and their ability to deal not only with uncertainty but also with imprecision, recently possibilistic networks also gained some attention Gebhardt (1997); Borgelt and Kruse (2002). They have been implemented e.g. in POSSINFER Gebhardt and Kruse (1995a); Kruse *et al.* (1994). In this paper we consider Bayesian networks and a type of possibilistic networks that is based on the context-model interpretation of a degree of possibility Gebhardt and Kruse (1993).

A Bayesian network is a directed acyclic graph in which each node represents a variable that is used to describe some domain of interest, and each edge represents a direct dependence between two variables. The structure of the directed graph encodes a set of conditional independence statements that can be read from the graph using a graph theoretic criterion called *d-separation* Pearl (1992). In addition, it represents a particular joint probability distribution, which is specified by assigning to each node in the network a (conditional) probability distribution for the values of the corresponding variable given the parent variables in the network (if any).

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

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

where $\mathrm{parents}(A_i)$ is the set of parents of variable A_i.

The directed acyclic graph of a Bayesian network captures the *global structure* of the underlying domain, i.e., the structure of (conditional) dependences and independences, but fails to take into account *local structure* that may be present in the conditional probability distributions stored with the nodes. An important issue in Bayesian network research is to capture such local structure and enable learning it from data.

In this paper we present a modification of the approach presented in Chickering *et al.* (1997) to learn Bayesian networks with a local decision graph structure from data. Our approach rests on exploiting the decision graph structure not only to capture a larger set of regularities in conditional probability tables but also to simplify the learning process. Our approach is also more efficient, because it needs fewer visits to the database to learn from.

Furthermore, we apply our local structure learning method to learning possibilistic networks from data. The transfer to this type of networks is straightforward. We also consider a large variety of evaluation measures (or scoring functions) for both probabilistic and possibilistic network learning. Many of these measures originated from decision tree learning, but can also be applied to learning Bayesian networks if the parents of a variable in a Bayesian network are seen as combined into one pseudo-variable. Some of them can easily be transferred to the possibilistic case. We study the influence of the evaluation measure on the running time of the learning algorithm and identify three classes of evaluation measures. Finally, we present experimental results for both learning Bayesian networks and possibilistic networks.

2 Possibilistic Networks

The development of possibilistic networks was triggered by the fact that probabilistic networks are well suited to represent and process *uncertain* information, but cannot that easily be extended to handle *imprecise* information. Since the explicit treatment of imprecise information is more and more claimed to be necessary for industrial practice, it is reasonable to investigate graphical models related to alternative uncertainty calculi, e.g. possibility theory.

Maybe the best way to explain the difference between uncertain and imprecise information is to consider the notion of a degree of possibility. The interpretation we prefer is based on the context model Gebhardt and Kruse (1993); Kruse *et al.* (1994). In this model possibility distributions are seen as information-compressed representations of (not necessarily nested) random sets and a degree of possibility as the one-point coverage of a random set Nguyen (1984).

Let ω_0 be the actual, but unknown state of a domain of interest, which is contained in a set Ω of possible states. Let $(C, 2^C, P)$, $C = \{c_1, c_2, \ldots, c_m\}$, be a finite probability space and $\gamma : C \to 2^\Omega$ a set-valued mapping. C is seen as a set of contexts that have to be distinguished for a set-valued specification of ω_0. The contexts are supposed to describe different physical and observation-related frame conditions. $P(\{c\})$ is the (subjective) probability of the (occurrence or selection of the) context c.

A set $\gamma(c)$ is assumed to be the *most specific correct set-valued specification* of ω_0, which is implied by the frame conditions that characterize the context c. By 'most specific set-valued specification' we mean that $\omega_0 \in \gamma(c)$ is guaranteed to be true for $\gamma(c)$, but is not guaranteed for any proper subset of $\gamma(c)$. The resulting *random set* $\Gamma = (\gamma, P)$ is an imperfect (i.e. imprecise *and* uncertain) specification of ω_0. Let π_Γ denote the *one-point coverage of* Γ (the *possibility distribution induced by* Γ*)*, which is defined as

$$\pi_\Gamma : \Omega \to [0,1], \pi_\Gamma(\omega) = P\left(\{c \in C \mid \omega \in \gamma(c)\}\right).$$

In a complete modeling, the contexts in C must be specified in detail, so that the relationships between all contexts c_j and their corresponding specifications $\gamma(c_j)$ are made explicit. But if the contexts are unknown or ignored, then $\pi_\Gamma(\omega)$ is the total mass of all contexts c that provide a specification $\gamma(c)$ in which ω_0 is contained, and this quantifies the *possibility of truth* of the statement "$\omega = \omega_0$" Gebhardt and Kruse (1993, 1996).

That in this interpretation a possibility distribution represents uncertain *and* imprecise knowledge can be understood best by comparing it to a probability distribution and to a relation. A probability distribution covers *uncertain*, but *precise* knowledge. This becomes obvious, if one notices that a possibility distribution in the interpretation described above reduces to a probability distribution, if $\forall c_j \in C : |\gamma(c_j)| = 1$, i.e. if for all contexts the specification of ω_0 is precise. On the other hand, a relation represents *imprecise*, but *certain* knowledge about dependences between attributes. Thus, not surprisingly, a relation can also be seen as a special case of a possibility distribution, namely if there is only one context. Hence the context-dependent specifications are responsible for the imprecision, the contexts for the uncertainty in the imperfect knowledge expressed by a possibility distribution.

With this interpretation the theory of possibilistic networks can be developed in analogy to the probabilistic case. The only difference is that instead of the product to determine a new joint distribution and the sum to determine a (new) marginal distribution, the operations minimum and maximum have to be used.

As a concept of possibilistic independence we use possibilistic non-interactivity. Let X, Y, and Z be three disjoint subsets of variables. Then X is called *conditionally* *independent* of Y given Z w.r.t. π, if $\forall \omega \in \Omega$:

$$\pi(\omega_{X \cup Y} \mid \omega_Z) = \min\{\pi(\omega_X \mid \omega_Z), \pi(\omega_Y \mid \omega_Z)\}$$

whenever $\pi(\omega_Z) > 0$, where $\pi(\cdot \mid \cdot)$ is a non-normalized conditional possibility distribution

$$\pi(\omega_X \mid \omega_Z) = \max\{\pi(\omega') \mid \omega' \in \Omega \wedge \text{proj}\,_X(\omega) = \omega_X \wedge \text{proj}\,_Z(\omega) = \omega_Z\},$$

with $\text{proj}_X(\omega)$ the projection of a tuple ω to the variables in X.

Learning possibilistic networks from data has been studied in Gebhardt and Kruse (1995b, 1996); Borgelt and Kruse (1997a,b). The idea to exploit local structure can be applied directly to (conditional) possibility distributions, since it is not bound to any specific uncertainty or imprecision calculus.

3 Local Network Structure

Whereas the global structure of a probabilistic or possibilistic network is the directed acyclic graph that encodes the conditional independence statements that hold in a certain domain of interest, the term "local structure" refers to regularities in the conditional probability or possibility tables that are stored with the nodes of the network. Several approaches to exploit such regularities have been studied for Bayesian networks in order to capture additional (i.e. context specific) independences and thus to (potentially) enhance inference. Among these are similarity networks Heckerman (1991) and the related multinets Geiger and Heckerman (1991), the use of asymmetric representations for decision making Smith *et al.* (1993) and probabilistic Horn rules Poole (1993), and finally also decision trees Boutilier *et al.* (1996) and decision graphs Chickering *et al.* (1997). In this paper we focus on the decision tree/decision graph approach, since it appears to be the most convenient one, and review it in the following for discrete Bayesian networks (i.e., in which all variables are discrete).

A very simple way to encode a conditional probability distribution is a table, which for each combination of values of the conditioning variables contains a line stating the corresponding conditional probability distribution for the values of the conditioned variable. As a simple example, let us consider the small section of a Bayesian network shown in figure 1 (and let us assume that in this network the variable C has no other parents than variables A and B). Let $\text{dom}(A) = \{a_1, a_2, a_3\}$, $\text{dom}(B) = \{b_1, b_2\}$, and $\text{dom}(C) = \{c_1, c_2\}$. Then the conditional probabilities $P(C = c_k \mid A = a_i, B = b_j)$ have to be stored with the node for variable C, e.g. as shown in table 1. The second column contains only entries $1 - p_i$, because the probabilities have to sum to 1 and there are only two possible values for variable C.

The same conditional probability distribution can also be stored in a tree, where the leaves hold the conditional probability distributions and each level of inner nodes corresponds to one conditioning variable (see figure 2). The branches in this tree are labeled with the values of the conditioning variables and thus each path from the root to a leaf corresponds to one combination of values of the conditioning variables. Obviously such a tree is equivalent to a decision tree for the variable C (like one learned e.g. by

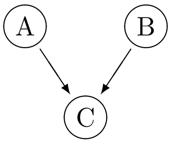

Figure 1. A small section of a Bayesian network.

parents		child	
A	B	$C = c_1$	$C = c_2$
a_1	b_1	p_1	$1 - p_1$
a_1	b_2	p_2	$1 - p_2$
a_2	b_1	p_3	$1 - p_3$
a_2	b_2	p_4	$1 - p_4$
a_3	b_1	p_5	$1 - p_5$
a_3	b_2	p_6	$1 - p_6$

Table 1. A conditional probability table for the network section shown in figure 1.

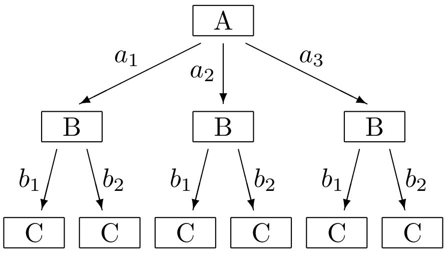

Figure 2. A full decision tree for variable C.

the well-known decision tree induction program C4.5 Quinlan (1993)) with the following restrictions: All leaves have to lie on the same level and in one level of the tree the same variable has to be tested on all paths. If these restrictions hold, we call the tree a *full* decision tree, because all possible combinations of values of the test attributes are explicitly represented in the tree.

Let us assume now that there are some regularities in the conditional probability distribution (see table 2), that is, let certain conditional probabilities be identical. Since the table clearly shows that the value of the variable B is important only if A has the value a_2, the tests of variable B can be removed from the branches for the values a_1 and a_3 (see figure 3). This shows the advantages of a decision tree representation.

Unfortunately, however, a decision tree is not powerful enough to capture all possible regularities that may be present in a conditional probability table. Although we can achieve a lot by accepting a change in the test order of the variables and by accepting binary splits and multiple tests of the same variable (then, for example, the regularities

parents		child	
A	B	$C = c_1$	$C = c_2$
a_1	b_1	p_1	$1 - p_1$
a_1	b_2	p_1	$1 - p_1$
a_2	b_1	p_3	$1 - p_3$
a_2	b_2	p_4	$1 - p_4$
a_3	b_1	p_2	$1 - p_2$
a_3	b_2	p_2	$1 - p_2$

Table 2. A conditional probability table for the section of a Bayesian network shown in figure 1 with some regularities.

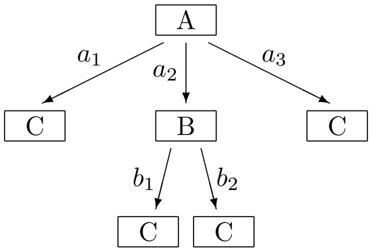

Figure 3. A partial decision tree for variable C.

in table 3 can be represented by the decision tree shown in figure 4), the regularities shown in table 4 cannot be represented by a decision tree.

The problem is that in a decision tree a test of a variable splits the lines of a conditional probability table into disjoint subsets that cannot be brought together again. In table 4 a test of variable B thus separates lines 1 and 2 and a test of variable A separates lines 4 and 5. Hence either test prevents us from exploiting one of the two equivalences of probabilities. This drawback can be overcome by allowing a node of the tree to have more than one parent, thus going from decision trees to decision graphs Chickering *et al.* (1997). With decision graphs the regularities in table 4 can easily be captured, see figure 5.

4 Learning Local Structure

To learn a decision graph three operations are defined in Chickering *et al.* (1997):

- *full split*: Split a leaf node according to the values of some variable.
- *binary split*: Split a leaf node such that one child corresponds to some value a_k of some variable and the other child to all other values of this variable.
- *merge*: merge two distinct leaf nodes.

A greedy algorithm based on these operations can easily be found Chickering *et al.* (1997). It applies all possible operations of the types defined above to a given decision

parents		child	
A	B	$C = c_1$	$C = c_2$
a_1	b_1	p_1	$1 - p_1$
a_1	b_2	p_1	$1 - p_1$
a_2	b_1	p_2	$1 - p_2$
a_2	b_2	p_3	$1 - p_3$
a_3	b_1	p_2	$1 - p_2$
a_3	b_2	p_4	$1 - p_4$

Table 3. A conditional probability table for the section of a Bayesian network shown in figure 1 with a second kind of regularities.

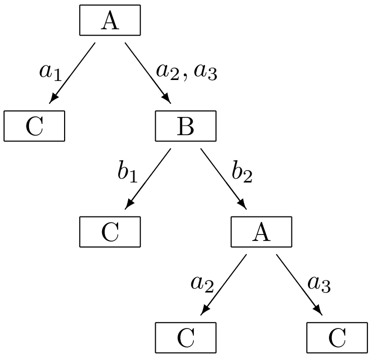

Figure 4. A decision tree with two tests of variable A that captures the regularities in the conditional probability table shown in table 3.

graph and then selects that operation (if any) that leads to the highest improvement of the network score. This search is carried out until no operation can be found that leads to an improvement.

Our approach is only a slight modification of the above. The additional degree of freedom of decision graphs compared to decision trees, namely that a node in a decision graph can have more than one parent, can be exploited not only to capture a larger set of regularities but also to improve the learning process for the local structure of a Bayesian network. Our idea is as follows: With decision graphs, we can always work with the complete set of inner nodes of a full decision tree and let only leaves have more than one parent. Even if we do not care about the order of the conditioning variables in the decision structure and allow only one test per variable on each path, such a structure can capture all regularities in the examples examined in the preceding section. For example, the regularities of table 3 are captured by the decision graph with a full set of inner nodes shown in figure 6.

parents		child	
A	B	$C = c_1$	$C = c_2$
a_1	b_1	p_1	$1 - p_1$
a_1	b_2	p_1	$1 - p_1$
a_2	b_1	p_2	$1 - p_2$
a_2	b_2	p_3	$1 - p_3$
a_3	b_1	p_3	$1 - p_3$
a_3	b_2	p_4	$1 - p_4$

Table 4. A conditional probability table for the section of a Bayesian network shown in figure 1 with a third kind of regularities.

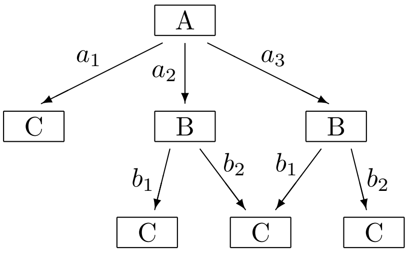

Figure 5. A decision graph for which no equivalent decision tree exists. It captures the regularities in table 4

It is easy to see that such an approach can capture any regularities that may be present in conditional probability tables. Basically, merging the leaves of a full decision tree is the same as merging lines of a conditional probability table. The decision graph structure just makes it much easier to keep track of the different value combinations of the conditioning (i.e. parent) variables, for which the same probability distribution for the values of the conditioned (i.e. child) variable has to be adopted.

In a learning algorithm we use only two operations, namely (1) adding a new level to a decision tree/graph, i.e., splitting all leaves according to the values of a new parent variable, and (2) merging two leaves into one. The first step, which may seem to be costly, does no harm, since it is necessary, even if one only learns a Bayesian network without local structure (provided the conditional distributions are represented as a decision tree). Only this step involves visiting the database to learn from in order to determine the conditional value frequencies. The next step, in which leaves are merged, can be carried out without visiting the database, since all necessary information is already available in the leaf nodes (provided the original leaf nodes are kept during a trial merge and are simply restored afterwards). Thus we need to visit the database only as often as an algorithm for learning a Bayesian network without local structure does. In contrast to this, the algorithm presented in Chickering *et al.* (1997) needs to visit the database each time a split of leaf nodes is considered. This can exceed by far the number of times

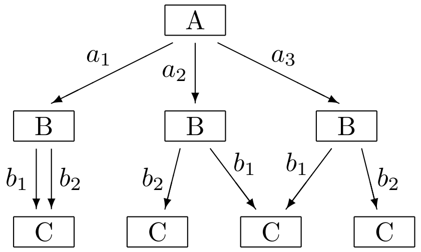

Figure 6. A decision graph with a full set of inner nodes that captures the regularities in table 3. Note that the test of variable B in the leftmost node on the second level is without effect, because both edges lead to the same leaf.

an algorithm for learning a network without local structure needs to visit the database, especially, since multiple tests of the same variable along the same path are permitted.

The leaf merging process is often less costly as it may seem at first sight, since we can exploit the fact that several evaluation measures (or scoring functions) are computed leaf by leaf or from terms that are computed leaf by leaf (see below). Hence, when two leaves are merged, the decision graph need not be reevaluated completely, but the change can often be computed locally from the frequency distributions in the merged leaves and the distribution in the resulting leaf.

To find the best set of mergers of leaf nodes, one can use any of the well-known search heuristics, e.g. greedy search or, if a mechanism for re-splitting leaf nodes is provided (which is easy to program), simulated annealing. Since we chose a greedy parent selection on a topological order (that is, the well-known K2 search method Cooper and Herskovits (1992)) in our experiments, we implemented a simple greedy search. That is, we always merge those two leaf nodes, that lead to the highest improvement of the evaluation measure. The merging process stops, if no leaf merger improves the value of the evaluation measure. However, we implemented the greedy merging in two different ways. In the first approach any merger between two leaf nodes of the current decision graph can be selected. We call this *unrestricted merging.* In the second approach, we first consider merging only such leaves that have the same parent. Only after no merger improving the evaluation measure can be found anymore, we allow mergers of leaves that have the same grandparent, and so on. This approach we call *levelwise merging*, since we climb up in the tree level by level to determine which leaves are considered for a possible merger. The latter approach can be slightly more efficient, since in general a slightly smaller set of mergers is considered. It can also lead to a simpler structure, since mergers of leaves that are “far apart” in the tree are less likely.

Of course, our approach can result in a complicated structure that may hide a simple structure of context-specific independences. But the same is true, though maybe less likely, for the algorithm presented in Chickering *et al.* (1997) and thus some postprocessing to simplify the structure found by the algorithm—for instance, by changing the order of variables and by splitting tests along a path—is always advisable.

5 Evaluation Measures

The process of selecting parent variables when learning a Bayesian network is very similar to selecting a test attribute in decision tree induction. The only difference is that in decision tree learning only single attributes are considered, whereas in Bayesian network learning there can be more than one parent. But this is not really a difference, since we can always view all parents as one pseudo-variable, the domain of which is the Cartesian product of the parents' individual domains.

This view can easily be extended to a decision graph representation, where several paths (and thus several pseudo-values) can lead to the same leaf (the same conditional probability distribution). In this case we only have to combine certain elements of the Cartesian product of the parents' domains into one pseudo-element. For example, for the decision graph shown on figure 5, we can view the parents A and B as one pseudo-variable X with $\mathrm{dom}(X) = \{x_1, x_2, x_3, x_4\}$, where $x_1 \hat{=} (a_1, b_1) \vee (a_1, b_2)$, $x_2 \hat{=} (a_2, b_1)$, $x_3 \hat{=} (a_2, b_2) \vee (a_3, b_1)$ and $x_4 \hat{=} (a_3, b_2)$.

The only thing we have to take care of is that in contrast to the measures commonly used for Bayesian network learning, like Bayesian measures based on the Bayesian Dirichlet metric or measures based on the minimum description length principle, attribute selection measures for decision tree induction usually do not have a built-in property that prevents them from selecting too many parent variables. An example is information gain, which for decision tree induction is known to be biased towards many-valued attributes.[1] Since an additional parent variable obviously increases the number of values of the pseudo-attributes, information gain tends to select too many parents. Fortunately, this drawback can easily be overcome by requiring a candidate parent to improve the value of an evaluation measure by a predefined minimal amount, before this candidate is considered eligible. We made this parameter an optional argument of our program.

Limits of space prevent us from describing in detail the evaluation measures we used in the experiments reported in section 6. Hence we only list them here without much explanation. An interested reader is asked to consult the references or Borgelt and Kruse (2002), which discusses them in some detail.

Probabilistic Measures

- information gain I_{gain} Kullback and Leibler (1951); Chow and Liu (1968) (mutual information/cross entropy)
- information gain ratio I_{gr} Quinlan (1993)
- symmetric information gain ratio I_{sgr} Lopez de Mantaras (1991)
- Gini index Breiman *et al.* (1984); Wehenkel (1996)
- symmetric Gini index Zhou and Dillon (1991)
- modified Gini index Kononenko (1994)
- relief measure Kira and Rendell (1992); Kononenko (1994)
- relevance Baim (1988)

[1] The reason is that a split of a value of a test attribute into two values can lead only to the same or a higher information gain, and in practice almost always leads to a higher information gain, mainly due to a quantization effect.

- χ^2 measure
- K2 metric Cooper and Herskovits (1992); Heckerman *et al.* (1995)
- BDeu metric Buntine (1991); Heckerman *et al.* (1995)
- minimum description length with coding based on relative frequencies l_{rel} Kononenko (1995)
- minimum description length with coding based on absolute frequencies l_{abs} Kononenko (1995) (closely related to the K2-metric)
- stochastic complexity Krichevsky and Trofimov (1983); Rissanen (1987)

Probabilistic Measures

- possibilistic analog of the χ^2-measure Borgelt and Kruse (1997a)
- possibilistic analog of mutual information (mutual specificity) Borgelt and Kruse (1997a)
- specificity gain S_{gain} Gebhardt and Kruse (1996); Borgelt and Kruse (1997a)
- specificity gain ratio S_{gr} Borgelt and Kruse (1997a)
- symmetric specificity gain ratio S_{sgr} Borgelt and Kruse (1997a)

When it comes to learning the local structure of a graphical model, it becomes important whether an evaluation measure can be computed from individual terms for each of the leaves of the decision graph representing the conditional distribution to assess, which makes it possible to compute the new value of the measure after merging two leaves by computing a simple delta, or whether it is not possible to find the new value by such "local" computations, so that the whole conditional distribution has to be reevaluated. This consideration leads to three classes of evaluation measures:

- The improvement resulting from a merger is independent of other mergers.

  Examples:
  - ○ χ^2 measure
  - ○ information gain
  - ○ K2 metric
- The improvement resulting from a merger depends on other mergers, but can be computed locally from the merged leaves and certain cached values.

  Examples:
  - ○ information gain ratio
  - ○ symmetric/modified Gini index
- The improvement resulting from a merger depends on other mergers in such a way that the full tree has to be reevaluated.

  Examples:
  - ○ specificity gain
  - ○ (symmetric) specificity gain ratio

In order to understand this distinction, let us briefly take a closer look at one example from each class. For the first class we consider the *K2 metric* Cooper and Herskovits (1992), which is based on a Bayesian approach. The idea underlying it is to compute the probability of a (directed) graph structure given the data, i.e., to compute

$$P(\vec{G} \mid D) = \frac{1}{P(D)} \int_{\Theta} P(D \mid \vec{G}, \Theta) f(\Theta \mid \vec{G}) P(\vec{G}) \, \mathrm{d}\Theta,$$

where $\vec{G}$ is the directed graph underlying the graphical model, D is the dataset to learn from, and Θ is the set of parameters of the model, i.e., the conditional probabilities. f describes the prior probability (in a Bayesian sense) of a each assignment of parameter values given the structure of the graph. By restricting our considerations to a Bayes factor for comparing networks, which eliminates the need to explicitly compute the probability of the database, and by making certain assumptions about data and parameter independence Cooper and Herskovits (1992), we get

$$P(\vec{G}, D) = \gamma \prod_{k=1}^{r} \prod_{j=1}^{m_k} \underbrace{\int \cdots \int}_{\theta_{ijk}} \left(\prod_{i=1}^{n_k} \theta_{ijk}^{N_{ijk}} \right) f(\theta_{1jk}, \ldots, \theta_{n_k jk}) \, \mathrm{d}\theta_{1jk} \ldots \mathrm{d}\theta_{n_k jk},$$

where γ is a normalization factor, r is the number of variables, m_k the number of distinct instantiations of the parent variables of variable k, n_k the number of values of variable k, θ_{ijk} the conditional probability that variable k has the i-th value given that its parent variables are instantiated with the j-th combination of values, and N_{ijk} is the number of times variable k is instantiated with its i-th value and its parents are instantiated with their j-th value combination in the database D to learn from. To solve this formula, $f(\theta_{1jk}, \ldots, \theta_{n_k jk}) = \text{const.}$ is chosen Cooper and Herskovits (1992) and then the solution can be obtained with Dirichlet's integral:

$$K_2(\vec{G}, D) = \gamma \prod_{k=1}^{r} \prod_{j=1}^{m_k} \frac{(n_k - 1)!}{(N_{.jk} + n_k - 1)!} \prod_{i=1}^{n_k} N_{ijk}!.$$

In implementations the logarithm of this measure is computed, so that the products turn into sums. From this formula it is obvious, that merging two leaves removes two factors from the second product, namely the two that refer to the two merged leaves, and adds a new one that refers to the result of the merger. Therefore the change of this measure as it results from merging leaves can easily be computed as a simple delta. This makes the computations very efficient.

As an example of an evaluation measure from the second class we consider the *information gain ratio* Quinlan (1993). This measure is based on Shannon entropy $H = -\sum_{i=1}^{n} p_i \log_2 p_i$ and can be seen as a normalization of the *information gain*

$$\begin{aligned} I_{\text{gain}}(A, B) &= H_A - H_{A|B} \;=\; H_A + H_B - H_{AB} \\ &= -\sum_{a \in \text{dom}(A)} P(A = a) \log_2 P(A = a) \\ &\quad - \sum_{b \in \text{dom}(B)} P(B = b) \log_2 P(B = b) \\ &\quad + \sum_{a \in \text{dom}(A)} \sum_{b \in \text{dom}(B)} P(A = a, B = b) \log_2 P(A = a, B = b), \end{aligned}$$

namely as

$$I_{\text{gr}}(A, B) = \frac{I_{\text{gain}}(A, B)}{H_A} = \frac{H_A + H_B - H_{AB}}{H_A}$$

The normalization is meant to remove the abovementioned bias towards many-valued attributes. It is easy to see that information gain allows to compute the change that results from merging two leaves as a simple delta, since only terms from the first and the third sum have to be replaced, while it is not possible to compute such a delta for information gain ratio due to the normalization factor. However, if we cache the values of the entropies it is computed from, the recomputation involves almost no additional costs. The entropies can be adapted by computing a delta resulting from the merger and then we only have to recompute the quotient.

As an example of an evaluation measure from the third class we consider the *specificity gain* Gebhardt and Kruse (1996); Borgelt and Kruse (1997a). It can be seen as a generalization of Hartley information gain on the basis of an α-cut view of possibility distributions and is defined as

$$\begin{aligned} S_{\text{gain}}(A,B) &= \int_0^{\sup \Pi} \log_2 \left(\sum_{a \in \text{dom}(A)} [\Pi]_\alpha (A=a) \right) + \log_2 \left(\sum_{b \in \text{dom}(B)} [\Pi]_\alpha (B=b) \right) \\ &\qquad - \log_2 \left(\sum_{a \in \text{dom}(A)} \sum_{b \in \text{dom}(B)} [\Pi]_\alpha (A=a, B=b) \right) \mathrm{d}\alpha. \end{aligned}$$

The formula shows that this measure is analogous to information gain. However, it does not share all of the nice properties of information gain. In particular, its change as it results from merging two leaves cannot be computed as a simple delta. The reason is that the computation of this measures involves sorting the degrees of possibility, and if two leaves are merged, they have to be resorted. This also makes it clear why no values can be cached to make a local computation possible. Fortunately, only the specificity based measures have this disadvantageous property. All other measures, possibilistic as well as probabilistic, belong to one of the other two classes.

6 Experimental Results

All experiments reported here were carried out with a prototype learning program for probabilistic and possibilistic networks called INES (Induction of NEtwork Structures, written by the first author of this paper), into which the described method and all of the listed evaluation measures are incorporated. This program as well as datasets and shell scripts to carry out the experiments can be retrieved free of charge at

`http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html#ines`

As a test case we chose the Danish Jersey cattle blood group determination problem Rasmussen (1992), for which a Bayesian network designed by domain experts and a database of 500 real world sample cases exist. Nevertheless, for learning Bayesian networks, we did not use the real world database, since it contains a lot of missing values. Instead, we used 20 artificially generated databases with 1000 sample cases each, 10 of which we used for learning, 10 for testing the learning result, over which the results were then averaged. The real world dataset was used only for learning possibilistic networks, since with them, missing values can be handled directly.

eval. measure	num. of conds.	add. conds.	miss. conds.	num. of params.	network quality train	 test
indep. vars.	0.0	0.0	22.0	59	−19921	−20087
original	22.0	0.0	0.0	219	−11391	−11506

Table 5. Reference evaluations for Bayesian network learning.

eval. measure	num. of conds.	add. conds.	miss. conds.	num. of params.	network quality train	 test
I_{gain}	35.0	17.1	4.1	1342	−11229	−11818
I_{gr}	24.0	6.7	4.7	209	−11615	−11737
I_{sgr}	32.0	11.3	1.3	317	−11388	−11575
Gini	35.0	17.1	4.1	1342	−11233	−11813
χ^2	35.0	17.3	4.3	1301	−11235	−11805
K2 metric	23.3	1.4	0.1	230	−11385	−11512
BDeu metric	31.2	9.3	0.1	276	−11385	−11521
l_{rel}	22.5	0.6	0.1	220	−11390	−11508

Table 6. Results of Bayesian network learning without local structure.

To evaluate the quality of the learned network, we chose the following approach: Given a Bayesian network, the probability of any (complete) sample case can easily be computed. If we assume the sample cases to be independent, we can compute from these probabilities the probability of the whole database (simply as their product). If we assume all network structures to have the same prior probability, this database probability is a direct measure of the network quality.

For possibilistic networks, we used a similar approach. Given a possibilistic network, the possibility degree of any (complete) tuple can be computed. If a tuple contains missing values, we assign to this tuple the maximal possibility degree over all complete tuples that are compatible with this tuple. The sum of these possibility degrees we used as a quality measure. This is justified, since due to the the way in which a possibilistic network approximates a multivariate possibility distribution, the possibility degree resulting from the network must always be equal or greater than the true possibility degree. Hence, the lower the sum of the possibility degrees for the tuples in the database, the better the network. More details about this evaluation method can be found in Borgelt and Kruse (1997b, 2002).

The results of some of our experiments are shown in tables 5 to 13. In addition to the network evaluation, these tables show the total number of conditions (parents) as a measure of the complexity of the global network structure, the number of additional and missing edges compared to the human expert designed reference network (which is reasonable only for Bayesian network learning), and the number of (probability or possibility) parameters as a measure of the complexity of the local network structure.

eval. measure	num. of conds.	add. conds.	miss. conds.	num. of params.	network quality train	 test
I_{gain}	35.0	17.1	4.1	1260	−11192	−11806
I_{gr}	31.6	11.0	1.4	133	−14979	−15151
I_{sgr}	34.7	13.9	1.2	342	−11424	−11675
Gini	35.0	17.1	4.1	1254	−11195	−11802
χ^2	35.0	17.3	4.3	1216	−11197	−11794
K2 metric	26.4	4.5	0.1	195	−11341	−11507
BDeu metric	36.0	14.3	0.3	306	−11336	−11505
l_{rel}	25.1	3.8	0.7	219	−11350	−11498

Table 7. Results of Bayesian network learning with local structure (unrestricted).

eval. measure	num. of conds.	add. conds.	miss. conds.	num. of params.	network quality train	 test
I_{gain}	35.0	17.1	4.1	1260	−11192	−11806
I_{gr}	32.1	11.7	1.6	132	−15202	−15354
I_{sgr}	34.7	13.9	1.2	342	−11424	−11675
Gini	35.0	17.1	4.1	1254	−11195	−11802
χ^2	35.0	17.3	4.3	1216	−11197	−11794
K2 metric	26.3	4.4	0.1	195	−11341	−11508
BDeu metric	35.9	14.2	0.3	305	−11338	−11504
l_{rel}	25.0	3.7	0.7	219	−11350	−11497

Table 8. Results of Bayesian network learning with local structure (levelwise).

eval. measure	num. of conds.	add. conds.	miss. conds.	num. of params.	network quality train	 test
I_{gain}	35.0	17.1	4.1	1260	−11192	−11806
I_{gr}	24.0	6.7	4.7	121	−14752	−14926
I_{sgr}	32.0	11.3	1.3	217	−11452	−11650
Gini	35.0	17.1	4.1	1253	−11195	−11802
χ^2	35.0	17.3	4.3	1216	−11197	−11794
K2 metric	23.3	1.4	0.1	168	−11352	−11492
BDeu metric	31.2	9.3	0.1	218	−11357	−11484
l_{rel}	22.5	0.6	0.1	162	−11357	−11488

Table 9. Results of Bayesian network learning with local structure preserving the global structure.

Table 5 shows the evaluation results for a graph without edges (independent variables) and the human expert designed reference structure. These evaluation can be used as a baseline for comparisons. From table 6 it can be seen that some measures tend to select too many conditions (parents), thus leading to overfitting. As already said, this disadvantage can be amended to some degree by requiring a certain minimal improvement of the network evaluation when adding a condition.

At first sight it is surprising that allowing local structure to be learned (see tables 7 to 9), although in most cases it leads to a reduction of the number of necessary parameters, makes the global structure more complex, since for several measures the number of selected conditions is larger than for networks without local structure. But a second thought (and a closer inspection of the learned networks) reveals that this could have been foreseen. In a frequency distribution determined from a database of sample cases random fluctuations are to be expected. Usually these do not lead to additional conditions (except for measures like information gain), since the "costs" of an additional level with several (approximately) equivalent leaves prevents the selection of such a condition. But the disadvantage of (approximately) equivalent leaves is removed by the possibility to merge these leaves, and thus those fluctuations that show a higher deviation from the true (independent) probability distribution are filtered out and become significant to the measure. Information gain ratio seems to be an especially pronounced example. The effect occurs for both unrestricted and levelwise merging, which lead to very similar results.

This effect reduces when learning from a larger dataset, but does not vanish completely. We believe this to be a general problem any learning algorithm for local structure has to cope with. Therefore it may be advisable not to combine learning global and local network structure, but to learn the global structure first and to simplify the learned structure afterwards by learning the local structure. To check this assumption, we applied learning the local structure to the outcome of global structure learning, with the sets of parents fixed. The result, which is shown in table 9, is indeed slightly better. However, information gain ratio still yields very bad results compared to the other measures, thus indicating that it is not adequate to select the leaves to merge and to determine when to stop merging.

The results of learning possibilistic networks with local structure, which are shown in tables 10 to 14, are very similar to the results of probabilistic network learning. However, the gains from local structure learning while preserving the learned global structure seem to be much smaller here and thus it seems to be more advisable to combine local and global structure learning.

7 Conclusions

In this paper we presented a method to learn the local structure of a Bayesian network from data, which we believe to be more efficient than the approach presented in Chickering *et al.* (1997). We applied the same idea to possibilistic networks, thus arriving at an algorithm to learn possibilistic networks with local structure. The experimental results show that trying to learn local structure has to be handled with care, since it can lead to the counter-intuitive effect of a more complicated global structure. Maybe it is advisable

eval.	num. of	num. of	network quality		
measure	conds.	params.	avg.	min.	max.
indep. vars.	0	80	10.160	10.064	11.390
original	22	308	9.917	9.888	11.318

Table 10. Results of possibilistic network learning without local structure.

eval.	num. of	num. of	network quality		
measure	conds.	params.	avg.	min.	max.
S_{gain}	31	1630	8.621	8.524	10.292
S_{gr}	18	196	9.553	9.390	11.100
S_{sgr}	28	496	9.057	8.946	10.740
poss. χ^2	35	1486	8.329	8.154	10.200
mut. spec.	33	774	8.344	8.206	10.416

Table 11. Results of possibilistic network learning without local structure.

eval.	num. of	num. of	network quality		
measure	conds.	params.	avg.	min.	max.
S_{gain}	34	768	8.739	8.548	10.620
S_{gr}	21	215	9.637	9.450	11.254
S_{sgr}	28	367	9.225	9.084	10.996
poss. χ^2	35	1348	8.347	8.152	10.222
mut. spec.	33	666	8.332	8.182	10.390

Table 12. Results of possibilistic network learning with local structure (unrestricted).

to base selecting another parent on the score for a full decision tree, and to use local structure learning only to simplify this tree afterwards.

Bibliography

S.K. Andersen, K.G. Olesen, F.V. Jensen, and F. Jensen. HUGIN — A shell for building Bayesian belief universes for expert systems. *Proc. 11th Int. J. Conf. on Artificial Intelligence*, 1080–1085, 1989

P.W. Baim. A Method for Attribute Selection in Inductive Learning Systems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10:888-896, 1988

C. Borgelt and R. Kruse. Evaluation Measures for Learning Probabilistic and Possibilistic Networks. *Proc. 6th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'97)*, Vol. 2:pp. 1034–1038, Barcelona, Spain, 1997

C. Borgelt and R. Kruse. Some Experimental Results on Learning Probabilistic and

eval. measure	num. of conds.	num. of params.	network quality avg.	 min.	 max.
S_{gain}	34	752	8.584	8.349	10.500
S_{gr}	21	215	9.637	9.450	11.254
S_{sgr}	28	361	9.252	9.110	11.008
poss. χ^2	35	1347	8.348	8.152	10.222
mut. spec.	33	674	8.332	8.182	10.390

Table 13. Results of possibilistic network learning with local structure (levelwise).

eval. measure	num. of conds.	num. of params.	network quality avg.	 min.	 max.
S_{gain}	31	1566	8.678	8.566	10.404
S_{gr}	18	182	9.627	9.446	11.202
S_{sgr}	28	455	9.074	8.948	10.812
poss. χ^2	35	1349	8.348	8.162	10.224
mut. spec.	33	621	8.402	8.262	10.502

Table 14. Results of possibilistic network learning with local structure preserving the global structure.

Possibilistic Networks with Different Evaluation Measures. *Proc. 1st Int. Joint Conference on Qualitative and Quantitative Practical Reasoning (ECSQARU/FAPR'97)*, pp. 71–85, Springer, Berlin, Germany, 1997)

C. Borgelt and R. Kruse. *Graphical Models — Methods for Data Analysis and Mining.* J. Wiley & Sons, Chichester, United Kingdom 2002

C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context Specific Independence in Bayesian Networks. *Proc. 12th Conf. on Uncertainty in Artificial Intelligence (UAI'96)*, Portland, OR, 1996

L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*, Wadsworth International Group, Belmont, CA, 1984

W. Buntine. Theory Refinement on Bayesian Networks. *Proc. 7th Conf. on Uncertainty in Artificial Intelligence*, pp. 52–60, Morgan Kaufman, Los Angeles, CA, 1991

D.M. Chickering, D. Heckerman, and C. Meek. A Bayesian Approach to Learning Bayesian Networks with Local Structure. *Proc. 13th Conf. on Uncertainty in Artificial Intelligence (UAI'97)*, pp. 80–89, Morgan Kaufman, San Franscisco, CA, 1997

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

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

J. Gebhardt and R. Kruse. The context model — an integrating view of vagueness and uncertainty *Int. Journal of Approximate Reasoning* 9:283–314, 1993

J. Gebhardt and R. Kruse. POSSINFER — A Software Tool for Possibilistic Inference. In: D. Dubois, H. Prade, and R. Yager, eds. *Fuzzy Set Methods in Information Engineering: A Guided Tour of Applications*, Wiley 1995

J. Gebhardt and R. Kruse. Learning Possibilistic Networks from Data. *Proc. 5th Int. Workshop on Artificial Intelligence and Statistics*, 233–244, Fort Lauderdale, 1995

J. Gebhardt and R. Kruse. Tightest Hypertree Decompositions of Multivariate Possibility Distributions. *Proc. Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems*, 1996

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

D. Geiger and D. Heckerman. Advances in Probabilistic Reasoning. *Proc. 7th Conf. on Uncertainty in Artificial Intelligence (UAI'91)*, pp. 118–126, Morgan Kaufman, San Franscisco, CA, 1997

D. Heckerman. *Probabilistic Similarity Networks.* MIT Press 1991

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

M. Higashi and G.J. Klir. Measures of Uncertainty and Information based on Possibility Distributions. *Int. Journal of General Systems* 9:43–58, 1982

K. Kira and L. Rendell. A Practical Approach to Feature Selection. *Proc. 9th Int. Conf. on Machine Learning (ICML'92)*, pp. 250–256, Morgan Kaufman, San Franscisco, CA, 1992

G.J. Klir and M. Mariano. On the Uniqueness of a Possibility Measure of Uncertainty and Information. *Fuzzy Sets and Systems* 24:141–160, 1987

I. Kononenko. Estimating Attributes: Analysis and Extensions of RELIEF. *Proc. 7th Europ. Conf. on Machine Learning (ECML'94)*, Springer, New York, NY, 1994

I. Kononenko. On Biases in Estimating Multi-Valued Attributes. *Proc. 1st Int. Conf. on Knowledge Discovery and Data Mining*, 1034–1040, Montreal, 1995

R.E. Krichevsky and V.K. Trofimov. The Performance of Universal Coding. *IEEE Trans. on Information Theory*, IT-27(2):199–207, 1983

R. Kruse, E. Schwecke, and J. Heinsohn. *Uncertainty and Vagueness in Knowledge-based Systems: Numerical Methods.* Series: Artificial Intelligence, Springer, Berlin 1991

R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of Fuzzy Systems*, John Wiley & Sons, Chichester, England 1994

S. Kullback and R.A. Leibler. On Information and Sufficiency. *Ann. Math. Statistics* 22:79–86, 1951

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, 1988

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

H.T. Nguyen. Using Random Sets. *Information Science* 34:265–274, 1984

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (2nd edition).* Morgan Kaufman, New York 1992

D. Poole. Probabilistic Horn Abduction and Bayesian Networks. *Artificial Intelligence*, 64(1):81-129, 1993

J.R. Quinlan. *C4.5: Programs for Machine Learning*, Morgan Kaufman, 1993

L.K. Rasmussen. *Blood Group Determination of Danish Jersey Cattle in the F-blood Group System.* Dina Research Report no. 8, 1992

J. Rissanen. Stochastic Complexity. *Journal of the Royal Statistical Society (Series B)*, 49:223-239, 1987

A. Saffiotti and E. Umkehrer. PULCINELLA: A General Tool for Propagating Uncertainty in Valuation Networks. *Proc. 7th Conf. on Uncertainty in AI*, 323–331, San Mateo 1991

G. Shafer and P.P. Shenoy. Local Computations in Hypertrees. Working Paper 201, School of Business, University of Kansas, Lawrence 1988

P.P. Shenoy. Valuation-based Systems: A Framework for Managing Uncertainty in Expert Systems. Working Paper 226, School of Business, University of Kansas, Lawrence, 1991

J.E. Smith, S. Holtzman, and J.E. Matheson. Structuring Conditional Relationships in Influence Diagrams. *Operations Research*, 41(2):280–297, 1993

L. Wehenkel. On Uncertainty Measures Used for Decision Tree Induction. *Proc. IPMU*, 1996

X. Zhou and T.S. Dillon. A statistical-heuristic Feature Selection Criterion for Decision Tree Induction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:834–841, 1991